

Part Number 10111
Copyright © 1996-2000 Compuware Corporation.
All Rights Reserved

Information in this document is subject to change without notice and does not represent a commitment on the part of Compuware Corporation. The software described in this document is furnished under the software license agreement distributed with the product. The software may be used or copied only in accordance with the terms of the license. The purchaser may make one copy of the software for a backup, but no part of this user manual may be reproduced, stored in a retrieval system, or transmitted in any form or by any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use, without prior written permission from Compuware Corporation.

Compuware, the Compuware logo, NuMega, the NuMega logo, DevPartner Studio, TrackRecord and the TrackRecord logo are either trademarks or registered trademarks of Compuware Corporation. Microsoft, Windows and Windows NT are either trademarks or registered trademarks of Microsoft Corporation. Netscape and Netscape Navigator are registered trademarks of Netscape Communications Corporation. Other brand and product names are either trademarks or registered trademarks of their respective holders.

Software License Agreement

This Software License Agreement is not applicable if Licensee has a valid Compuware License Agreement and has licensed this Software under a Compuware Product Schedule.

Please Read This License Carefully

You, either an individual or entity ("Licensee"), are purchasing a license ("Agreement") to use this Compuware Corporation software ("Software") and related user manuals ("Documentation"). The Software is the property of Compuware Corporation ("Compuware") and/or its licensors, is protected by intellectual property laws, and is provided to Licensee only under the license terms set forth below. This Agreement does not transfer title to the intellectual property contained in the Software. Compuware reserves all rights not expressly granted herein. Opening the package and/or using the Software indicates your acceptance of these terms. If you do not agree to all of the terms and conditions, or if after using the Software you are dissatisfied, you may return the Software, Documentation and any copies within thirty (30) days of purchase to the party from whom you received it for a refund, subject in certain cases to a restocking fee.

Title and Proprietary Rights: Licensee acknowledges and agrees that the Software is proprietary to Compuware and/or its licensors, and is protected under the laws of the United States and other countries. Licensee further acknowledges and agrees that all rights, title and interest in and to the Software, including intellectual property rights, are and shall remain with Compuware and/or its licensors. Unauthorized reproduction or distribution is subject to civil and criminal penalties.

Use Of The Software: Compuware grants Licensee the limited right to use the Software and Documentation included in the package with this license, subject to the terms and conditions of this Agreement. Licensee agrees that the Software will be used solely for internal purposes. Only one copy of the Software may be installed on a single computer at any one time unless:

- (i) The Software is designed and intended by Compuware for use in a shared network client server environment, as set forth in the Documentation; and
- (ii) Licensee agrees to provide technical or procedural methods to prevent use of the Software, even at different times, by anyone other than Licensee; and
- (iii) Licensee has purchased a license for each individual user of the Software and/or for each computer that will have access to the Software. Any unauthorized use of this Software may cause termination of this Agreement.

Licensee may make one machine-readable copy of the Software for BACK UP PURPOSES ONLY. This copy shall display all proprietary notices, be labeled externally to show that it is the property of Compuware, and that its use is subject to this Agreement. Documentation may not be copied in whole or part. Licensee agrees to provide technical or procedural methods to prevent use of the Software by anyone other than Licensee, even at different times.

Licensee may not use, transfer, assign, export or in any way permit the Software to be used outside the country of purchase, unless authorized in writing by Compuware. Except as expressly provided in this Agreement, Licensee may not modify, reverse engineer, decompile, disassemble, distribute, sub-license, sell, rent, lease, give or in any way transfer the Software, by any means or in any medium, including telecommunications. Licensee will use its best efforts and take all reasonable steps to protect the Software from unauthorized use, copying or dissemination, and will retain all proprietary notices intact.

Redistribution Rights of Device Driver Development Software: This section shall only apply if the Software is device driver development software, used by Licensee to develop application or device driver programs ("User Software"), as specified in the Documentation. The User Software may include run-time components ("RTCs") that have been extracted by the Software from the library files of the Software, programs to remotely test the User Software, and compiled code examples. These RTCs, examples, and programs are specifically designated as redistributable in the Documentation. Licensee has a non-exclusive, royalty-free, restricted license to:

- (i) modify, compile, and distribute the driver code examples;
- (ii) distribute the remote testing program for testing purposes only;
- (iii) embed the RTCs and driver code examples in its User Software, in object code form only; and
- (iv) reproduce and distribute the RTCs and driver code examples embedded in its User Software, in object code form only, provided that:
 - (a) Licensee distributes the RTCs and driver code examples only in conjunction with and as a part of its User Software;
 - (b) Licensee will be solely responsible to anyone receiving its User Software for any updates, technical and other support obligations, and any other liability which may arise from the distribution of its User Software;

- (c) Licensee does not use Compuware's or its licensors' names, logos, or trademarks to market or distribute its User Software;
- (d) Licensee includes Compuware's and its licensors' copyright and/or proprietary notices and legends within the executable images of its User Software and on Licensee's software media and documentation; and
- (e) Licensee agrees to indemnify, hold harmless and defend Compuware and its licensors from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of its User Software.

Government Users: With respect to any acquisition of the Software by or for any unit or agency of the United States Government, the Software shall be classified as "commercial computer software," as that term is defined in the applicable provisions of the Federal Acquisition Regulation (the "FAR") and supplements thereto, including the Department of Defense (DoD) FAR Supplement (the "DFARS"). If the Software is supplied for use by DoD, the Software is delivered subject to the terms of this Agreement and either (i) in accordance with DFARS 227.7202-1(a) and 227.7202-3(a), or (ii) with restricted rights in accordance with DFARS 252.227-7013(c)(1)(ii) (OCT 1988), as applicable. If the Software is supplied for use by a Federal agency other than DoD, the Software is restricted computer software delivered subject to the terms of this Agreement and (i) FAR 12.212(a); (ii) FAR 52.227-19; or (iii) FAR 52.227-14(ALT III), as applicable. Licensor: Compuware Corporation, 31440 Northwestern Highway, Farmington Hills, Michigan 48334.

Limited Warranty and Remedy: Compuware warrants the Software media to be free of defects in workmanship for a period of ninety (90) days from purchase. During this period, Compuware will replace at no cost any such media returned to Compuware, postage prepaid. This service is Compuware's sole liability under this warranty. COMPUWARE DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO LICENSEE. IN THAT EVENT, ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO THIRTY (30) DAYS FROM THE DELIVERY OF THE SOFTWARE. LICENSEE MAY HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

Infringement of Intellectual Property Rights: In the event of an intellectual property right claim, Compuware agrees to indemnify and hold Licensee harmless, provided Licensee gives Compuware prompt written notice of such claim, permits Compuware to defend or settle the claim, and provides all reasonable assistance to Compuware in defending or settling the claim. In the defense or settlement of such claim, Compuware may obtain for Licensee the right to continue using the Software or replace or modify the Software so that it avoids such claim, or if such remedies are not reasonably available, accept the return of the infringing Software and provide Licensee with a pro-rata refund of the license fees paid for such Software based on a three (3) year use period.

Limitation of Liability: LICENSEE ASSUMES THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE. IN NO EVENT WILL COMPUWARE BE LIABLE TO LICENSEE OR TO ANY THIRD PARTY FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING BUT NOT LIMITED TO, LOSS OF USE, DATA, REVENUES OR PROFITS, ARISING OUT OF OR IN CONNECTION WITH THIS AGREEMENT OR THE USE, OPERATION OR PERFORMANCE OF THE SOFTWARE, WHETHER SUCH LIABILITY ARISES FROM ANY CLAIM BASED UPON CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, AND WHETHER OR NOT COMPUWARE OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO LICENSEE. IN NO EVENT SHALL COMPUWARE BE LIABLE TO LICENSEE FOR AMOUNTS IN EXCESS OF PURCHASE PRICE PAID FOR THE SOFTWARE.

Term and Termination: This License Agreement shall be effective upon Licensee's acceptance of this Agreement and shall continue until terminated by mutual consent, or by election of either Licensee or Compuware in case of the other's unremediated material breach. In case of any termination of this Agreement, Licensee will immediately return to Compuware the Software that Licensee has obtained under this Agreement and will certify in writing that all copies of the Software have been returned or erased from the memory of its computer or made non-readable.

General: This Agreement is the complete and exclusive statement of the parties' agreement. Should any provision of this Agreement be held to be invalid by any court of competent jurisdiction, that provision will be enforced to the maximum extent permissible and the remainder of the Agreement shall nonetheless remain in full force and effect. This Agreement shall be governed by the laws of the State of Michigan and the United States of America.



Table of Contents

Part 1: Using TrackRecord

1 Introducing TrackRecord

What is TrackRecord?	1
The Process at a Glance	1
Step 1: Enter Data	2
Step 2: Retrieve Data	2
Step 3: View Data	3
About the TrackRecord Database	7
Types, Items, and Fields	7
Database Links	7
Data Type Inheritance	8
Change History	9
Client-Server Technology	10
Users and Administrators	10

2 Starting TrackRecord

Before You Begin	11
Running TrackRecord	12
Exploring TrackRecord's Main Window	13
Menu Bar	13
Toolbar	14
Status Bar	15
Changing Your Login Password	16
Using TrackRecord Online Help	16
Displaying the About TrackRecord Window	17

3 Entering Information

Using TrackRecord Forms	19
Item Tab	20

Links Tab	20
Change History Tab	21
TrackRecord Form Controls	22
Using the Item Browser	25
Example: Using a TrackRecord Defect Form	26
Changing the Status of an Item	28
Merging One Item with Another	29

4 Using Queries and Views

Queries and Views	31
About Queries.	32
Query Composition Strategies	36
Using Outline Reports	37
Creating a New Outline Report	38
Adding Multiple Headers	44
Dragging Headers in an Outline Report	44
Opening Items in an Outline Report.	44
Creating New Items from an Outline Report	44
Using the Remainder Header	45
Using the Shortcut (Right-Click) Menu.	46
Using the Header Wizard	47
Using the Graph View	51
Creating Graphs from an Outline Report	52
Formatting Graphs.	53
Using the Milestone Status View	54
Filtering Milestone Status Data	58
Using the Volatility Statistics Windows	59
Interpreting the Volatility Graph	62
Using the Calendar View	62
Setting Up Your Default Calendar	64
Opening Your Calendar View	65
Opening, Modifying, and Deleting Queries and Views.	66
Identifying Favorite Reports and Home Pages.	67
Searching the TrackRecord Database	68
Performing a Simple Search	68
Performing an Advanced Search	69

5 Templates

About Templates.	73
When Are Templates Applied?.	73
When You Create Items Directly.	73
When You Create Items From Within Outline Reports	74

Creating Templates	75
Associating Templates with a Header	77

6 Customizing User Options

Customizing TrackRecord	81
Configuring General Options	81
Identifying Your Favorite Data Types	83
Setting Up Outline Reports	84
Logging In as Someone Else	85

7 TrackRecord and Other Applications

Using TrackRecord with Text Editors	87
Using TrackRecord with Graphics Applications	88
Using TrackRecord with Microsoft Project	89
Overview of Integration	89
Microsoft Project Macros	90
TrackRecord Menu	91
Initial Project Setup	91
Mapping TrackRecord Priorities to Project Priorities	93
Synchronizing a Project in TrackRecord and Microsoft Project	94
What Happens During Synchronization?	95
Using TrackRecord with Version Control Systems	97
Checking Files In and Out	100
Synchronizing TrackRecord and VCS Projects	101
Troubleshooting	101

8 Using TrackRecord WebServer

About WebServer	105
Logging In to TrackRecord WebServer	105
The WebServer Window	106
Creating a New Item	108
Opening an Outline Report	109
WebServer Window Notes	111
Logging Out	111

9 Using TrackRecord with DevPartner Studio

About TrackRecord and DevPartner Studio	113
DevPartner Studio Interaction with TrackRecord	114
Logging In to TrackRecord from DevPartner Studio Tools	114
DevPartner Studio Integration with Visual Studio	115
DevPartner Menu	116
Task Lists	116
DevPartner Toolbar Buttons	116

Automated Session Submission	117
Build Registration.	117
Instrumenting Your Code	117
Defect/Task Submissions	117
Debug Session Output	119
Submitting Debug Sessions	120
Submitting Tuning Sessions.	121
Detecting Duplicate Defects and Tasks	121
Resolving Multiple Matching Build Conflicts	122
Viewing Task Lists	122
TrackRecord and DevPartner Studio Coverage Information	122
Platform Configuration Information.	123
Viewing Captured Platform Configuration Information	124
Remote Error Capture.	125

10 Using TrackRecord with QACenter and Fault Manager

Using TrackRecord and QACenter	127
Using TrackRecord with Fault Manager	128
Copy and Paste.	129
Save As	129

Part 2: Administering TrackRecord

11 TrackRecord Configuration

The Administrator	133
Configuration Administration Tasks	134
Group Administration	135
Group Access Mechanisms.	135
Creating Groups.	136
Share Groups	137
Creating Team Members	138
Project Configuration.	141
Project Administration Privileges	141
Creating a Project.	142
Cloning Projects.	143
Multiple Projects	143
Historical Information	143
Global Preferences Administration	143
Creating Global Queries and Outline Reports.	144
Identifying Favorite Types, Reports, and Home Pages	145

Creating Global Templates.	146
Workflow Administration.	147
Creating and Modifying a Workflow.	148
Working with States.	150
Working with Transitions	151
Designing a Workflow Based on Type.	153
Adding a Status Field to a Type	153
Default Status for New Items	155
Type Administration.	156
Type Inheritance	156
Categories of Types.	157
Abbreviations	157
Identifying Duplicate Imported Items.	158
Altering Access Rights for Types and Fields.	158
ActiveLink Tags	158
Database Lock-out	159
Type Administration Tasks.	160
Creating and Modifying Types.	161
Using the Type Editor.	164
Restrictions.	172
Custom Items	173
Adding Rules That Control the Behavior of Types	174
Registering One or More Application Software Builds.	175

12 TrackRecord Database Administration

Overview.	179
Database Administration	180
New Database.	180
Backup and Restore Database.	181
Copy Database	181
Check Database	182
Rebuild Database	183
Administering WebServer.	184
WebServer Processes.	184
Permanent and Transient Sessions.	184
WebServer Log Files	185
Ending a Session.	185
WebServer Administration Menu	186
WebMonitor.	189
Troubleshooting Common WebServer Issues	189
Configuring AutoAlert	192
Installation Notes	192
Setting Up AutoAlert Users.	193

Setting Up Mail Queries	194
Using the AutoAlert Administrator	195
Adding and Removing Databases	196
Show Users	197
Options	197
Changing the Contents of Mail Messages	198
Importing and Exporting Data	199
Using the Import-Export Wizard to Export	199
Exporting Using the Command Line Interface	201
Using the Import-Export Wizard to Import	201
Using the Import and Export Menu Options	202
Using Copy and Paste to Export Data	203
Manually Creating Template Files and Importing	203

A ActiveX Programming Interface

ActiveX Interfaces	209
Basic Design	209
Simple Example	210
Object Reference	210
Application Object	210
Abbreviation Object	214
Abbreviations Object (Collection)	215
Database Object	215
Field Object	216
FieldLayout Object	217
Fields Object (Collection)	218
History Object	218
Histories Object (Collection)	220
Item Object	221
Items Object (Collection)	223
Query Object	224
Queries Object (Collections)	225
Type Object	226
Types Object (Collection)	227
User Object	227
Users Object (Collection)	229

Glossary

Index



How To Use This Manual

This manual describes how to use Compuware TrackRecord® software to track defect reports, feature requests, tasks, and other project-related information during software development cycles.

Who Should Read This Manual

This manual is intended for all TrackRecord users, TrackRecord administrators, and programmers writing modules to access a TrackRecord database.

- ◆ Users should read Part 1 (chapters 1 through 10) for information about TrackRecord's concepts and user interface and to learn how to enter information into TrackRecord, how to use predefined reports to display the contents of the TrackRecord database, and how to format custom TrackRecord reports.
- ◆ Administrators should read Part 2 (Chapters 11 and 12) for a description of administration tasks.
- ◆ Programmers should read Appendix A.

This manual assumes that you are familiar with the Microsoft Windows interface and with software defect tracking concepts.

What This Manual Covers

This manual contains two parts, one appendix, and a Glossary.

Part 1, Using TrackRecord, contains the following ten chapters:

- ◆ **Chapter 1, "Introducing TrackRecord,"** describes the concepts and components of TrackRecord, discusses how the database stores and displays information, and explains the types of reports you can generate in TrackRecord.

- ◆ **Chapter 2, “Starting TrackRecord,”** explains the TrackRecord user interface and describes how you log in to TrackRecord and how to change your login preferences.
- ◆ **Chapter 3, “Entering Information,”** explains how you enter defect reports, feature requests, and other information into TrackRecord.
- ◆ **Chapter 4, “Using Queries and Views,”** explains how to create TrackRecord reports.
- ◆ **Chapter 5, “Templates,”** describes how you can create templates to expedite information entry.
- ◆ **Chapter 6, “Customizing User Options,”** describes how to customize your startup options, favorite types, and Outline Reports.
- ◆ **Chapter 7, “TrackRecord and Other Applications,”** explains how TrackRecord integrates with text editors, graphics programs, Microsoft Project, and version control software.
- ◆ **Chapter 8, “Using TrackRecord WebServer,”** describes how to use the TrackRecord web interface to enter new information and display Outline Reports.
- ◆ **Chapter 9, “Using TrackRecord with DevPartner Studio,”** describes how to use the TrackRecord with the DevPartner Studio tuning and debugging tools.
- ◆ **Chapter 10, “Using TrackRecord with QACenter and Fault Manager,”** describes how to use TrackRecord with the Compuware QACenter automated software quality assurance and Fault Manager fault detection tools.

Part 2, *Administrating TrackRecord* contains the following two chapters:

- ◆ **Chapter 11, “TrackRecord Configuration,”** describes the tasks required to set up and manage TrackRecord for all users.
- ◆ **Chapter 12, “TrackRecord Database Administration,”** describes the tasks required to set up and maintain the TrackRecord database, how to provide Web access to the database, and how to enable e-mail notification.

Appendix A, “ActiveX Programming Interface,” provides reference material for programmers writing modules to access the TrackRecord database.

You will also find a glossary and an index at the back of this manual.

Conventions Used In This Manual

This book uses the following conventions to present information.

- ◆ Screen commands and menu names appear in **bold typeface**. For example:

Choose **Item Browser** from the **Tools** menu.

Computer commands and file names appear in monospace typeface. For example:

The *BoundsChecker Basics* manual (bc_vc.pdf) describes...

- ◆ Variables within computer commands and file names (for which you must supply values appropriate for your installation) appear in *italic monospace type*. For example:

Enter `http://servername/cgi-win/itemview.dll`

- ◆ The names of TrackRecord database items and the names of TrackRecord views and reports are capitalized.

A *Team Member* item might be connected to *Project* items...

- ◆ Screen illustrations in this manual use the default screen layouts and database schema shipped with TrackRecord. If your TrackRecord administrator modifies TrackRecord screens or system files to reflect your site's practices or terminology, the screens in the manual may not match what you see on your computer.
- ◆ User tips appear in italic type in the left column of a page.

For More Information

You can use the TrackRecord online help to learn more about TrackRecord's functions and procedures. To access the TrackRecord online help system, choose **Help** from the **Help** menu or select a screen control and press the F1 key.

Part 1: Using TrackRecord

- ◆ Chapter 1: Introducing TrackRecord
- ◆ Chapter 2: Starting TrackRecord
- ◆ Chapter 3: Entering Information
- ◆ Chapter 4: Using Queries and Views
- ◆ Chapter 5: Templates
- ◆ Chapter 6: Customizing User Options
- ◆ Chapter 7: TrackRecord and Other Applications
- ◆ Chapter 8: Using TrackRecord WebServer
- ◆ Chapter 9: Using TrackRecord with DevPartner Studio
- ◆ Chapter 10: Using TrackRecord with QACenter and Fault Manager



Chapter 1

Introducing TrackRecord



This chapter introduces you to Compuware TrackRecord®, a software development tool that helps you organize information about products you are developing or supporting. This chapter provides an overview of TrackRecord functions and concepts. You should read this chapter carefully before you use TrackRecord for the first time.

What is TrackRecord?

Compuware TrackRecord is a software development tool designed to record and report information about products being developed or supported. TrackRecord records all the information relating to a project—project team members and testers, schedules and milestones, bug reports and feature requests—in an object database. You can then use TrackRecord’s query and reporting features to retrieve and format the information you need to keep your project on track.

TrackRecord is part of the Compuware NuMega DevPartner Studio suite of software debugging tools, Compuware ABend-Aid Fault Manager, and the Compuware QACenter family of automated testing tools. Interaction with these families of products is described in Chapter 9, “Using TrackRecord with DevPartner Studio,” and Chapter 10, “Using TrackRecord with QACenter and Fault Manager.”

The Process at a Glance

You use TrackRecord to enter information about projects and to retrieve and view that information. This section provides an overview of those fundamental tasks.

There are many other tasks performed with TrackRecord, such as creating projects, assigning team members, sharing queries and reports, and so on, but the three primary tasks – entering, retrieving and viewing data – provide the framework for all other tasks.

Each of these tasks is described briefly in the following sections and is described in detail in later chapters.

Step 1: Enter Data

To enter or change information, you open an *item view*. An item view is a form on which you can enter or modify information about one item, such as one Defect or one Person. Each type of item uses a different form, since each type requires different information.

Item views use standard Windows controls, such as text fields, drop-down lists, scrolling lists, checkboxes, and radio buttons, to display and update information about specific items. They also use some unique controls, such as the Calendar and Add Item buttons.

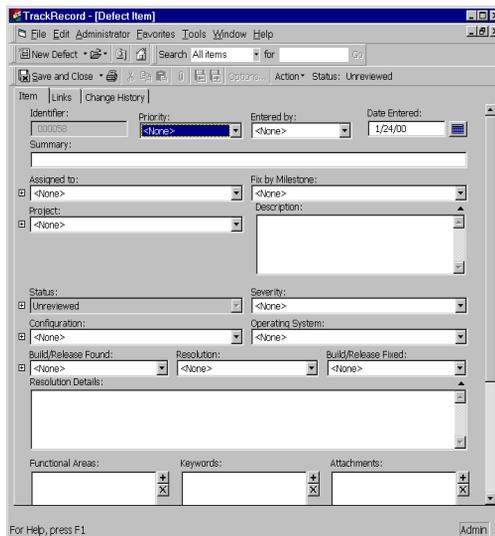


Figure 1-1 A Task Item View

You will learn more about using item views to enter information in Chapter 3, “Entering Information.”

Step 2: Retrieve Data

To retrieve information that has been entered, you create *queries*. A query indicates what information you want to retrieve. For example, the following query would find all defects in the database that are assigned to project Bullwinkle and have a Priority of 1, 2, or 3.

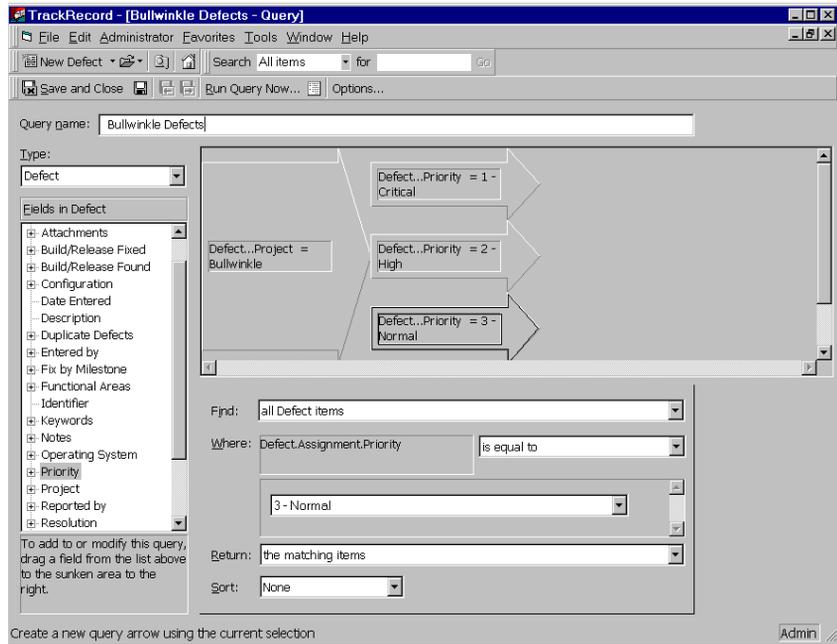


Figure 1-2 Using a Query to Identify Defects

When you set up a query, you select the type of item you want to find (Defect in the example) and the fields you want the query to search on (Project and Priority in the example), and add parameters to make your query as specific as necessary.

You can create your own queries or use queries that have been set up for you by your TrackRecord Administrator.

You will learn more about setting up and using queries in “About Queries” on page 32.

Step 3: View Data

The data retrieved by a query can be displayed in one of four formats:

- ◆ Outline Report view
- ◆ Graph view
- ◆ Milestone Status view
- ◆ Calendar view

Each of these views is described briefly in the following sections and is described in detail in Chapter 4, “Using Queries and Views.”

You can create your own views or use views that have been set up for you by your TrackRecord Administrator.

Outline Reports

Outline Reports are dynamic reports that let you view summaries of the TrackRecord data returned by a query or set of queries. For example, you could create the following Outline Report (Figure 1-3) using the example query shown in Step 2.

Defects	Summary	Assigned to
•000059: Crashes on Save	Crashes on Save	ALT,
•000060: Data files corrupted	Data files corrupted	DCH,

Figure 1-3 Using an Outline Report View to List Defects by Priority

You can select what fields appear in columns in the report. You can change the order in which the columns are displayed and the order in which the headers are displayed.

Graph View

The Graph view (Figure 1-4) lets you display a graphical summary of the information in an Outline Report. Depending on the information in the Outline Report, you can generate pie charts, simple bar graphs, clustered or stacked bar graphs, line graphs, area graphs, and trend graphs.

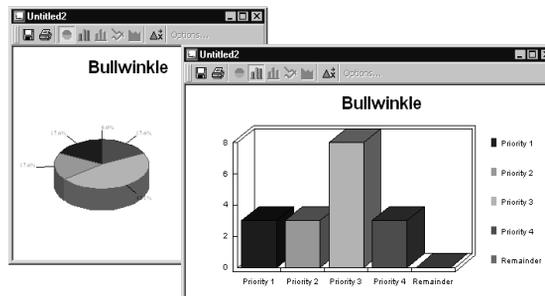


Figure 1-4 Sample Graph Views

Milestone Status View

The Milestone Status view (Figure 1-5) provides a concise, one-report summary of project status. The report captures a broad range of project status indicators, from how much the code base is changing (volatility) to how much unit testing and automated error detection has been performed. With the ability to add custom queries on the report, team leads and team members can quickly assess the true status of the project at every milestone.

Note Some parts of the Milestone Status view are product-specific. If you are not using the tool that populates a particular section of the view, that section will not contain data.

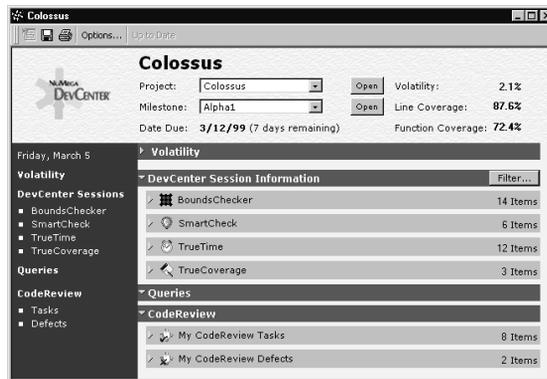


Figure 1-5 Sample Milestone Status View

Calendar View

The Calendar view (Figure 1-6 and Figure 1-7) functions as an appointment book and To Do list, in which TrackRecord automatically categorizes and displays the appropriate data by date and time.

Depending on your preferences, the daily or weekly Calendar view displays a simple list of your appointments and tasks or a tick-marked schedule of each day. By changing the Calendar view's query, you can display different data or dates. You can also maintain and display multiple calendars simultaneously.

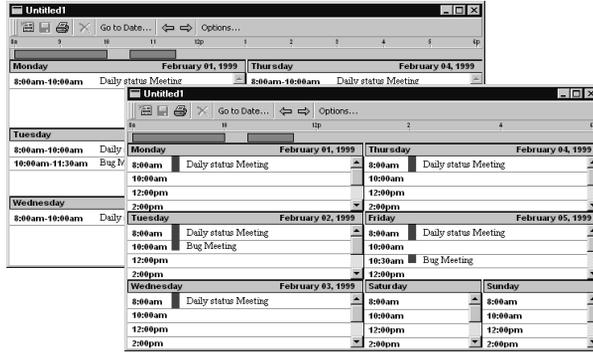


Figure 1-6 Sample Calendar by Week Views

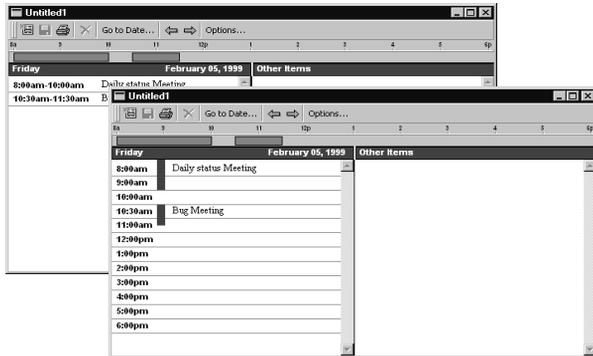


Figure 1-7 Sample Calendar by Day Views

About the TrackRecord Database

Using TrackRecord does not require an extensive understanding of database concepts. A basic understanding of how TrackRecord stores and links the information you enter, however, will improve your productivity with TrackRecord.

When you enter information using TrackRecord, the information is stored in a database. TrackRecord automatically opens its database when you start the program, and automatically closes your connection to the database when you exit. You interact with this database through item views, queries, and report views, as described in the previous section.

Types, Items, and Fields

The TrackRecord database organizes information into *types*, such as Person, Company, Defect, or Project. Each data type functions like a database table, establishing the properties (fields) TrackRecord expects for an item of that type. Visually, a data type is a form, with fields and controls to enter or display information.

A data *item* is one record of a specific type. For example, Edison Carter and Harry Truman would be two items of the type Person in a TrackRecord database.

Each data item consists of *fields* and *links*. Fields hold specific facts about an item, such as a person's address or the name of a project. Links connect one item in the TrackRecord database to another, as described in the following section.

In addition to the obvious information types, such as Projects or People, the TrackRecord database can store special types of information, such as executable commands and database queries.

Database Links

The TrackRecord database is an *object* database. Unlike flat-file databases, which store information in table-like records, the TrackRecord object database stores information as sets of discrete items. TrackRecord *links* objects of one type to objects of other types to capture the relationships between items. For example, a Team Member item (Edison Carter) might be connected to Project items (as a team member), Defect items (as the person reporting them or responsible for fixing them), and Company items (as an employee).

Figure 1-8 illustrates how a Team Member item links to a Company item and to several Project items. These links make it easy to retrieve information, such as all Team Members assigned to a Project or all Projects with open defects.

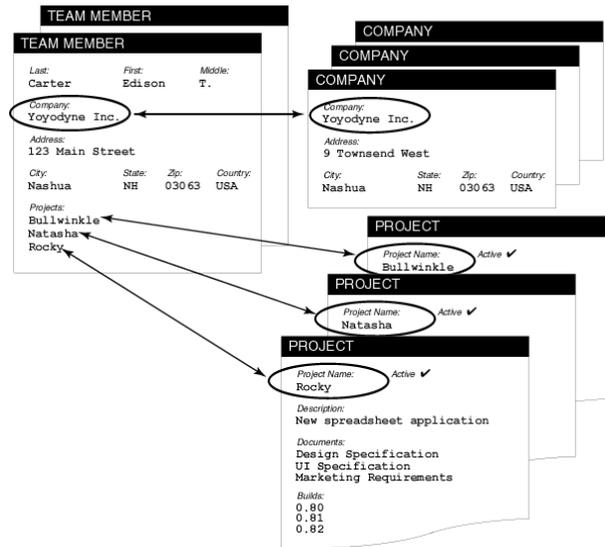


Figure 1-8 Links Between Database Items

When an object is entered in a field, that field is automatically linked to the original item. When you change the item, no matter where it appears, all references to that item change *automatically*.

For example, the default Task type includes a field called **Assigned To**. This field can contain a Person item. If you assign several Tasks to a Person, and then change the spelling of the Person's name in any of those tasks, that change is reflected in all other items that contain a link to that Person.

Data Type Inheritance

The TrackRecord database uses a data *hierarchy* to develop related data types. When one data type (the *child*) is based on another data type (the *parent*), the child type inherits all the fields and controls of the parent type. The child type can then add fields of its own, according to its specific purpose. Figure 1-9 illustrates how the Person data type is the parent to the Developer and Beta Tester data types; each child type starts with the fields it inherits from the Person type, and adds its own fields below the inherited fields.

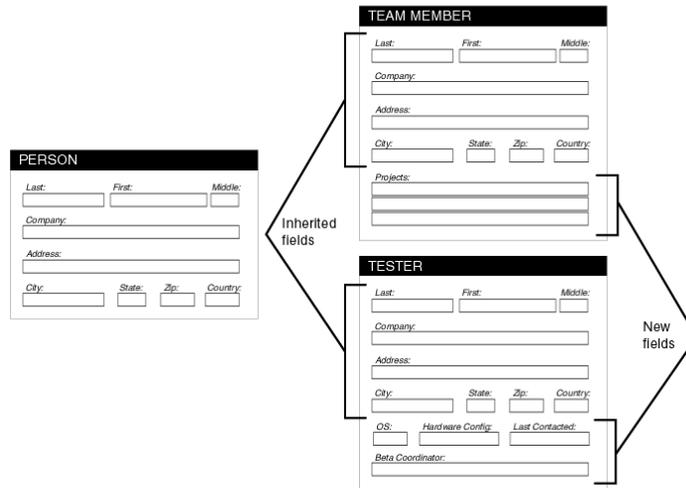


Figure 1-9 Child Items Inherit Fields from Parent

This type of inheritance allows items of any “child” type to be used anywhere in TrackRecord where an item of its “parent” type is expected. For example, the Team Member type inherits from the Person type, so all Team Member items are also Person items. Team Members and Testers can be used to fill fields where Person items are expected. Note, however, that the reverse is *not* generally true: Person items that are not Team Members cannot be used in fields where Team Members are required.

Change History

As database items are created and modified, TrackRecord captures the time and date of each change, the person adding or changing the item, and the information that was added or changed.

Figure 1-10 illustrates a Change History window for a Project item.

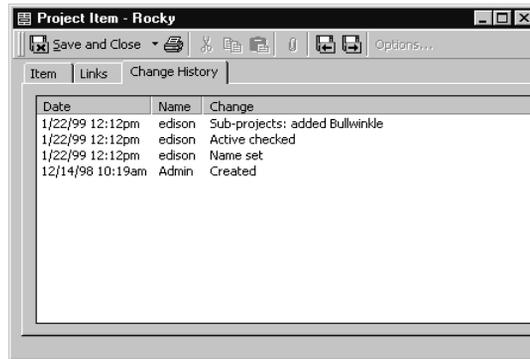


Figure 1-10 Change History Window

Client-Server Technology

The TrackRecord database resides on a central server on your network. The TrackRecord server software accepts simultaneous input from multiple users on client machines; the TrackRecord server periodically checks the database for new information during a session and forwards updated information to clients when necessary.

Users and Administrators

TrackRecord supports three types of users:

- ◆ TrackRecord Administrator — TrackRecord includes administrative functions that can only be performed by authorized users, such as setting up user groups, importing and exporting information to and from other applications, and editing the data types used by TrackRecord. When a TrackRecord administrator logs in, he or she sees an additional menu (the Administrator menu) on the TrackRecord menu bar.
- ◆ TrackRecord User — A user running the TrackRecord client software accesses the TrackRecord database by logging in from client machines anywhere on your network. Whether a user can add or modify database information, including items and reports, depends on the groups to which the user belongs. The access and editing privileges of each user is defined by the administrator.
- ◆ WebServer User — WebServer users access the database by logging in using standard Web browsers, such as Microsoft Internet Explorer or Netscape Navigator. Refer to Chapter 8, “Using TrackRecord WebServer,” for information on the WebServer interface.

Chapter 2

Starting TrackRecord



This chapter describes the TrackRecord user interface, menus, and windows. This chapter also describes how to log in to TrackRecord, how to change your login password, and how to use the TrackRecord online help system.

Note TrackRecord Administrators should refer to Part II of this book for information about setting up the TrackRecord database before users begin creating items with TrackRecord.

Before You Begin

Before you begin using TrackRecord, you will need to obtain the following information from your TrackRecord administrator:

- ◆ **What user name and password should you use?** Before you can log in to TrackRecord, you must have a valid user name and password.
- ◆ **What database should you use?** If your site has several development projects running concurrently, your TrackRecord administrator might set up separate databases. You will need the name of your database.
- ◆ **What project should you use?** A database can include multiple projects. Your TrackRecord administrator can tell you what project name you should use to display and enter project information.

Running TrackRecord

- 1 Click the **Start** button on your Windows desktop.
- 2 Select **Programs/Compuware/TrackRecord**.

Note You or your administrator may have set up your TrackRecord user information to automatically log you in to TrackRecord. If so, the login window will not be displayed.

The TrackRecord Login window (Figure 2-1) opens.

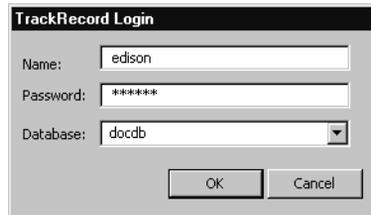


Figure 2-1 TrackRecord Login Window

- 3 Enter your TrackRecord login name and password.

Note that your TrackRecord login name and password may be different from your Windows login name and password.

Passwords are case-sensitive. For example, if your password is NuMega, you will not be able to enter Numega, numega, or NUMEGA when you log in to TrackRecord.

- 4 Identify the database you want TrackRecord to use in the Database field.

If you have used TrackRecord before, the names of the databases you have used previously appear in the **Databases** list. You can select the database you want to use from this list or enter a new database name. You do not need to enter the path to the database, just the database name.

- 5 Click **OK**.

The main TrackRecord window (Figure 2-2) opens and displays any subwindows left open from your previous session. If your TrackRecord Administrator has set up a Home page, the Home page opens.

Since various aspects of this window can be customized, your window may vary from the illustration.

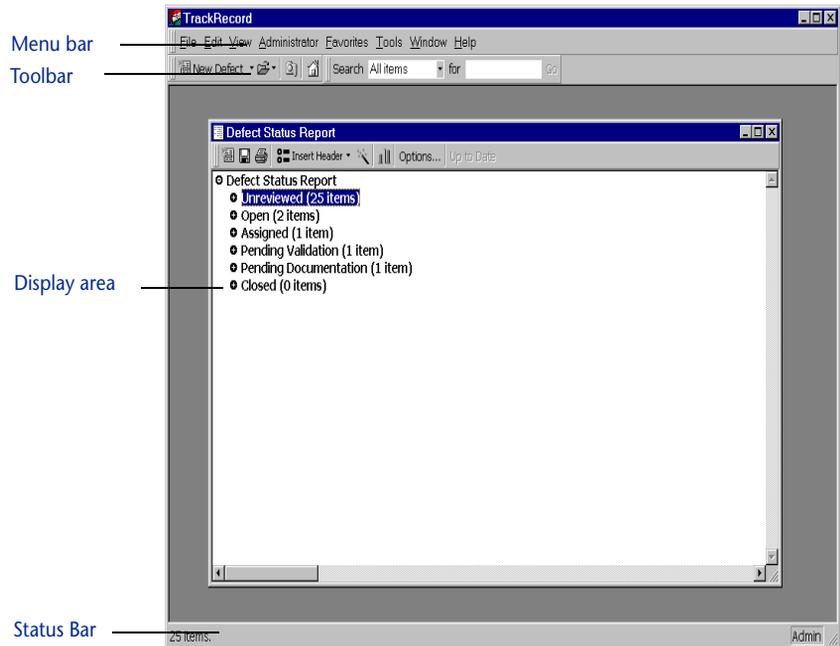


Figure 2-2 TrackRecord Main Window

Exploring TrackRecord's Main Window

Before you begin using TrackRecord, you should take a few minutes to explore its windows, toolbar, and menus. Each of the options mentioned in the following sections is described in more detail in later chapters.

The main TrackRecord window opens when you log in to TrackRecord.

Menu Bar

The main TrackRecord window displays a menu bar below the window title.

- ◆ The **File menu** lets you create new items and reports, open and manage predefined reports, queries, and templates, and lists items you have recently worked with. The File menu also includes standard File menu entries, such as Print Setup, Print, and Exit.
- ◆ The **Edit menu** lets you copy and paste text and graphics from one place to another. The contents of the Edit menu change according to which window is active within TrackRecord. For example, when you open an Outline Report, the Edit menu displays commands for adding, modifying, and deleting report headers. When you open a

graph, the Edit menu displays commands for copying the graph to the Clipboard and changing graph options.

- ◆ The **Administrator menu** (displayed only if you log in to TrackRecord with Administrator access) lets the TrackRecord administrator set up users and user groups, edit data types and user defaults, and import and export data and type descriptions.
- ◆ The **Favorites menu** lets you specify the TrackRecord reports you use most frequently. After you add a report to the Favorites menu, you can run it by choosing its name from the menu. You can also specify one report as your Home Page. This report will open automatically when you log in to TrackRecord.
- ◆ The **Tools menu** lets you search the TrackRecord database for items that meet the criteria you specify, log in as another TrackRecord user, browse TrackRecord items, and modify your TrackRecord options.
- ◆ The **Window menu** lets you position TrackRecord windows and switch from one window to another.
- ◆ The **Help menu** lets you access the TrackRecord online documentation and the About TrackRecord window, which displays registration and software release information.

Toolbar

The toolbar (Figure 2-3) contains shortcuts to your most commonly performed tasks. It lets you create a new item or view, open an existing item, display the Item Browser, open your home page, or search for items.

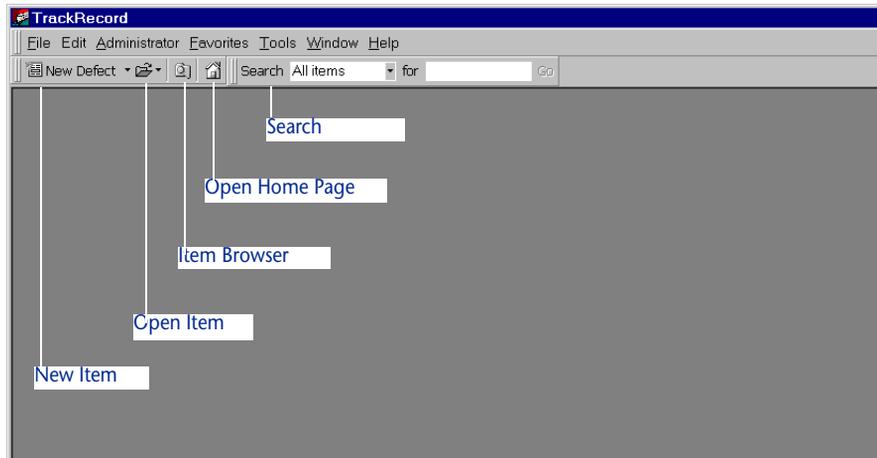


Figure 2-3 Toolbar

- ◆ **New Item** - Use this button and drop-down list to create a new item or view. The item types listed on the drop-down list are determined by your Favorite Types list (as described in “Identifying Your Favorite Data Types” on page 83).

You can control the new item type shown on the Toolbar with your Favorite Types list. For example, if you most often create new Tasks, you would move Task to the top of your Favorite Types list and the Toolbar would show New Task rather than New Defect, which is the default.

- ◆ **Open Item** - Click this button to open an existing item or view. The Open dialog is described in “Opening, Modifying, and Deleting Queries and Views” on page 66.
- ◆ **Item Browser** - Click this button to display the Item Browser. The Item Browser is described in “Using the Item Browser” on page 25.
- ◆ **Open Home Page** - Click this button to open your Home Page report. Setting a Home Page is described in “Identifying Favorite Reports and Home Pages” on page 67.
- ◆ **Search** - Lets you search items in the TrackRecord database for a specified string.

To use the Search toolbar, select the type of item you want to search from the **Search** drop-down list, enter the text string you want in the **for** field, and click **Go**.

You can control whether TrackRecord displays the Search tool by right-clicking anywhere in the Toolbar area and toggling Search off. To redisplay the Search tool, right-click in the Toolbar area and toggle Search on.

Status Bar

The left side of the Status bar (at the bottom of the screen) provides suggestions for the actions you can perform with a tool or on a selected object. If you are working in an Outline Report, the status bar identifies the sequence number of the item you have selected and the number of items in the sequence.

The right side of the status bar displays the login name of the current user.

Changing Your Login Password

To ensure the security of the information in a TrackRecord database, the first task you should perform with TrackRecord is to change your login password. (Note that if you or your Administrator has set up your user profile to allow you to automatically log in when you start TrackRecord, this task is not necessary.)

You can change your TrackRecord login password at any time. To change your password:

- 1 **Choose Change Password from the Tools menu.**

The Change Password window (Figure 2-4) opens.

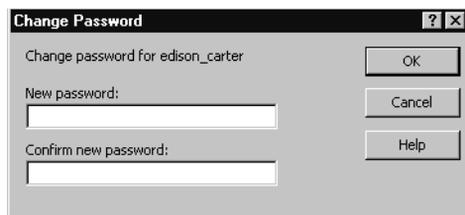


Figure 2-4 Change Password Window

- 2 **Enter your new password in the New Password field.**

To prevent anyone from observing what you type, TrackRecord displays an asterisk for each character you enter. Passwords are case-sensitive.

- 3 **Enter the identical password in the Confirm New Password field.**

Entering your password a second time ensures that you have entered it correctly.

- 4 **If you do not wish to use a password, leave the New Password and Confirm Password fields blank.**
- 5 **Click OK.**

Your new password takes effect immediately.

Using TrackRecord Online Help

TrackRecord's online help facility is a convenient way to look up information about a task you are performing, a feature you would like to know more about, or a command you want to use.

- ◆ Pressing the F1 key opens the online help system and displays a help topic appropriate for the task you are performing.

- ◆ Selecting the **TrackRecord Help** option on the **Help** menu displays a window containing TrackRecord's help system. You can navigate to the desired information through the Contents, Index, or Search tabs.

Topics may contain links to other topics or links to popup frames. To close a popup frame, click anywhere inside the popup.

Displaying the About TrackRecord Window

The About Compuware TrackRecord window (Figure 2-5) identifies the version of TrackRecord you are using, the registered user's name and serial number, the server and database TrackRecord is using, and information about the computer you are using. This information may be required should you contact Compuware Technical Support for assistance.



Figure 2-5 About Compuware TrackRecord Window

To display the TrackRecord About window, from the **Help** menu choose **About Compuware TrackRecord**. Click **OK** when you are finished.

Chapter 3

Entering Information



This chapter describes how to enter different types of information into the TrackRecord database.

Using TrackRecord Forms

For each type of information stored in the TrackRecord database — Defects, Tasks, Team Members, and so on — you will want to store different pieces of information. For example, you might want to store a name and phone number for a Team Member and a summary and description for a Defect. Each item type has its own data entry form, which is composed of fields appropriate to that item type.

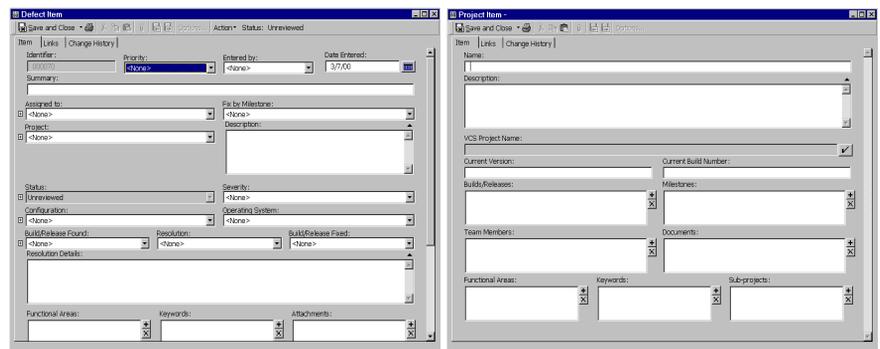


Figure 3-1 A Defect Form and a Project Form

To make data entry even easier, you can set up item templates that store default values for fields. Templates are like overlays for an overhead projector: they automatically add data to an item. Information about how to set up and use templates is provided in Chapter 5, “Templates.”

TrackRecord forms include three tabs: the Item tab, the Links tab, and the Change History tab. (The Team Member form contains additional tabs.) These three tabs are described in the following sections.

Item Tab

The Item tab on a TrackRecord form (Figure 3-2) lets you enter information about the item you are adding or updating. The fields displayed on the item tab depend on the type of item with which you are working.

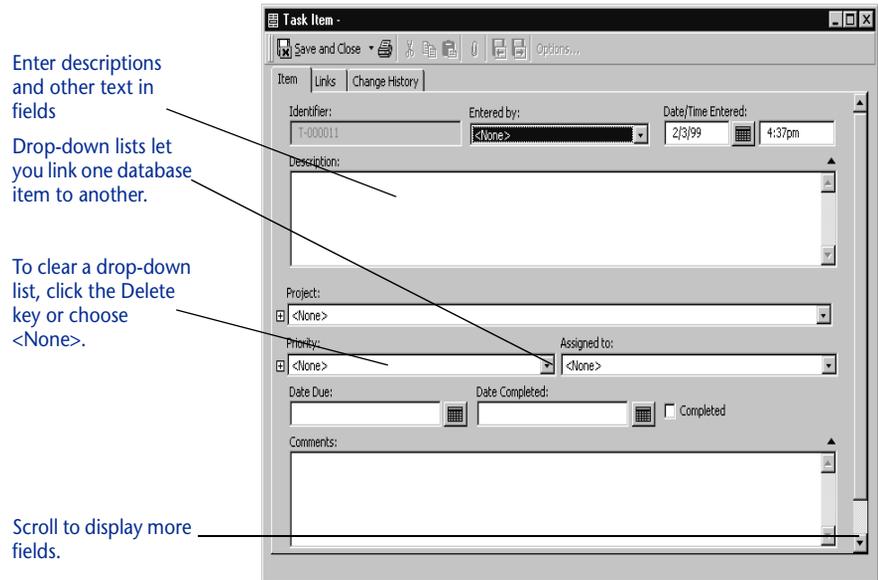


Figure 3-2 Item Tab

Links Tab

As you add information to the database, TrackRecord creates a network of links between items. The Links tab (Figure 3-3) displays a list of all the items in the TrackRecord database that refer to the item with which you are currently working.

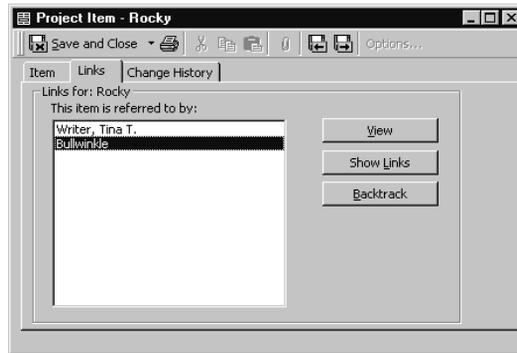


Figure 3-3 Links Tab

You can navigate through your network of linked items by selecting a link.

- ◆ Clicking the **View** button opens the item in a new window.
- ◆ Clicking the **Show Links** button follows the link to see what other database items have links to the linked-to item.
- ◆ Clicking the **Backtrack** button returns to the item you were viewing when you clicked **Show Links**.

Change History Tab

The Change History tab (Figure 3-4) provides an audit trail of the changes made to an item over time. When an item is created, modified, or deleted, an entry is added to the change history list for that item. Each entry contains the login name of the user, the field or control that was modified, the date and time when the change was made, and, if appropriate, information about the type of change that was made. If a person made several changes to an item during a single session, each change will be noted separately.

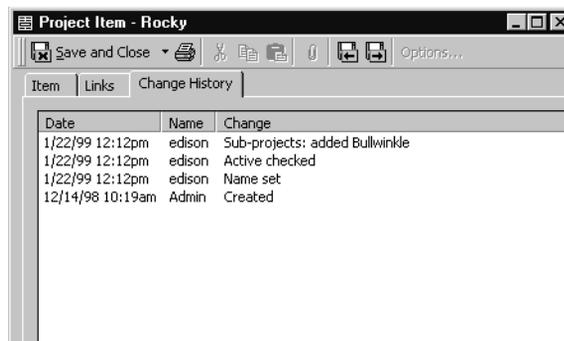


Figure 3-4 Change History Tab

TrackRecord Form Controls

In addition to standard Windows controls, such as text boxes, lists, buttons, and checkboxes, TrackRecord uses a number of special controls to help you create, display, and maintain database information. This section describes how to use each of these special controls.

Expand Item Button

As you learned in Chapter 1, fields in one type of item can be links to other items in the database. For example, in a Defect item the Project field might be a link to an item of the Project type. The Project field's drop-down list would display a list of all Project items in the database, allowing you to select an existing Project. The selected Project would then be "linked" to this defect.

The Expand Item button lets you expand a link to another item. You can display and update the fields belonging to that linked item. For example, clicking the Expand Item button beside the Priority list in Figure 3-5 causes the window to display the Name and Description fields for the selected Priority. If you have appropriate privileges, you can add new items or modify the items displayed. If you make changes, these changes will be reflected in all other items that are linked to the changed item.

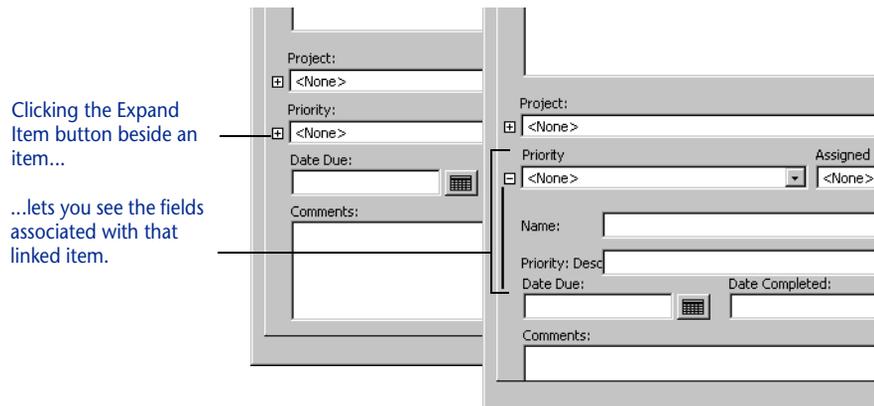


Figure 3-5 Using the Expand Item Button

Calendar Button

The Calendar button (Figure 3-6) lets you specify dates from a graphical calendar instead of entering a date with the keyboard. Click the

Calendar button to open the calendar and double-click a date to select the day you want to enter.

To choose a date on a past or future month, click the left (past) or right (future) arrows at the top of the calendar. The single triangles move one month at a time. The double triangles move one year at a time.

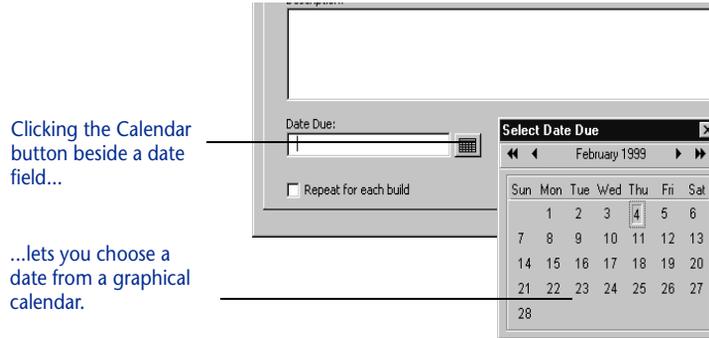


Figure 3-6 Using the Calendar Button

Add/Delete Item Buttons

The Add/Delete Item buttons (Figure 3-7) let you add and delete items in the TrackRecord database in the course of maintaining other types of items. For example, if you want to enter a new telephone number while adding or editing a Person item, you would click the **Add Item** button beside the Phone Number field, create the item you want, and click **Save and Close**. (Note that the behavior of the **Add Item** button is configurable. Some fields display a list of existing items, allowing you to pick one or to click **New Item** to create a new item of that type.)

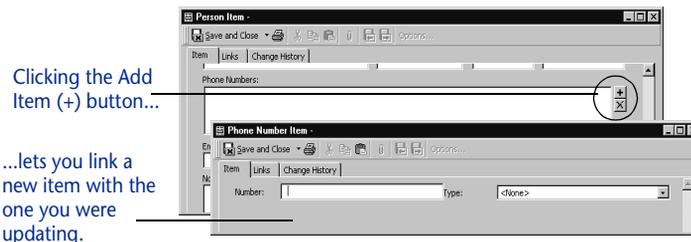


Figure 3-7 Using the Add/Delete Item Buttons

To delete an item from one of these lists, select the item and click the X button. Depending on how this field was defined when the type was created, deleting an item from a list may remove it from the database.

You will be prompted to confirm the deletion. If the field was not defined to allow deletion from the database with the X button, you can delete it from the database by right-clicking on the item and choosing **Delete *itemname*** from the shortcut menu.

Expand Text Field Button

The Expand Text Field button (Figure 3-8) identifies resizable text fields. When you click the Expand Text Field button, the text field area expands to the bottom of the window to let you display the complete text of an extended description or note about an item. You can click the Collapse Text Field button to collapse the text field to its original size.

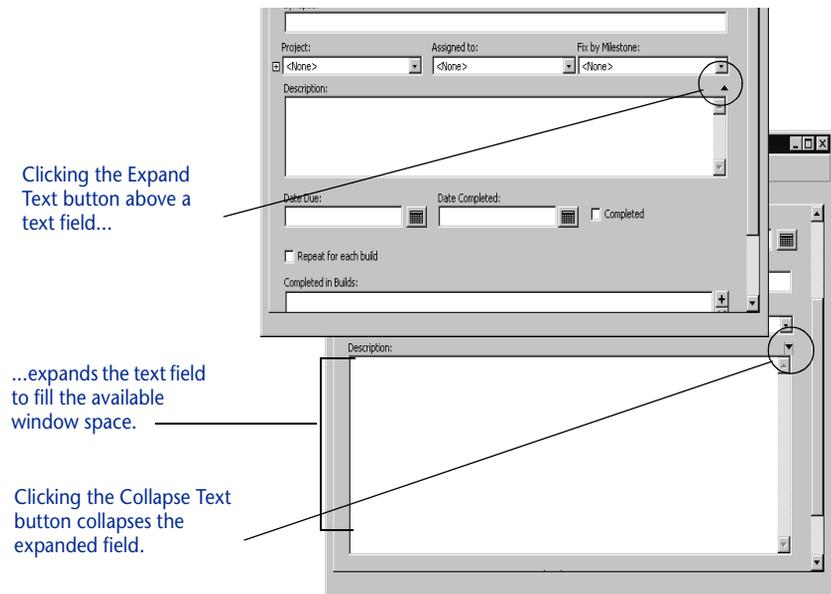


Figure 3-8 Using the Expand Text Button

Using the Item Browser

The Item Browser window (Figure 3-9) provides access to information in your TrackRecord database. With the Item Browser, you can access existing database items of any type or create new items.

The Item Browser window includes two lists:

- ◆ The Types list identifies the basic types of information in the TrackRecord database. If one type of information is a subtype of another, it appears indented below that type. For example, Team Member is a subtype of the Person data type in Figure 3-9.
- ◆ The Items list identifies all of the items of the selected type. When you select a type that has subtypes, the names of all the members in the subtypes appear. For example, if you choose Person, the Items list will include the names of all Team Member items (subtype of Person) as well as all Person items.

Note The list of types that you see might be a subset of the types available. Your access to types is controlled by the administrator. Only the administrator can view all types.

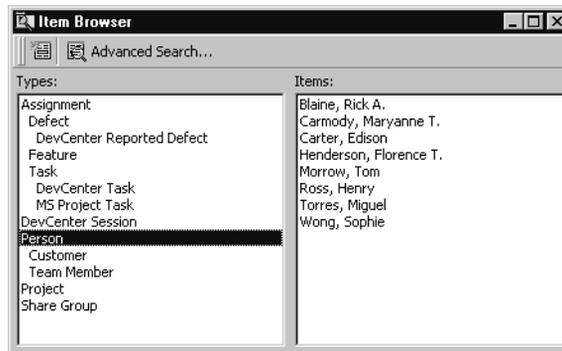


Figure 3-9 Item Browser Window

To Use the Item Browser

- 1 Click the **Item Browser** button on the **TrackRecord Toolbar** to open the **Item Browser** window.

Note TrackRecord often presents several ways to accomplish a task. For example, you can open the Item Browser with the Toolbar button, from the Tools/Item Browser option, or from the File/Open/Other option.

- 2 **Select the type of item about which you want information.**

For example, select **Person** to see the list of all **Person** items in the TrackRecord database. Note that the list will include all items of the type you selected and of all children of that type. For example, selecting **Person** will show a list of **Persons**, **Customers**, and **Team Members** in the Item list.

- 3 **Double-click an entry in the Items list to display the item's contents.**

For example, double-clicking the name of a **Person** in the Items list opens the appropriate item view, letting you view and modify the information recorded for that **Person**.

Note that if you select an item that belongs to a child type of the data type you chose, TrackRecord opens the item view for the child. For example, if you choose **Person** and then double-click the name of someone who is a **Team Member** (child of **Person**), TrackRecord opens the **Team Member** item view instead of the **Person** item view.

Example: Using a TrackRecord Defect Form

- 1 From the **File** menu, Open a new **Defect** form using one of the following methods:
 - ◇ Click the **New Defect** button on the **Toolbar**. (If you or your Administrator has changed your Favorite Types list, the **New Defect** button may not be displayed. In this case, use the **Toolbar's New Item** drop-down list to select **Defect**.)
 - ◇ Open the **Item Browser**, (use the button on the **Toolbar**), right-click **Defect**, and choose **New Defect** from the shortcut menu.
 - ◇ Choose **New/Defect** from the **File** menu.

- ◇ Choose **New/Other** from the **File** menu. When the **Choose a Type** window (Figure 3-10) opens, select **Defect** and click the **New** button.

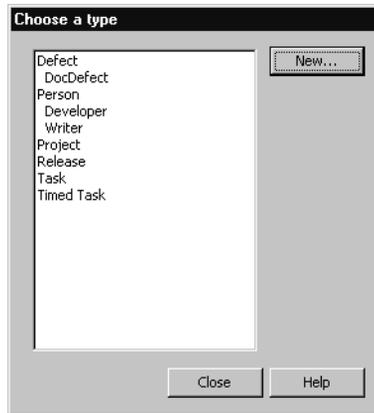


Figure 3-10 Choose a Type Window

- ◇ If an Outline Report includes entries for defects, right-click a Defect entry and choose **New Defect** from the shortcut menu.
- 2 When the Defect form (Figure 3-11) appears, enter information in the appropriate fields.**

If you are unfamiliar with the controls on this screen, refer to “TrackRecord Form Controls” on page 22.

Figure 3-11 Defect Form

If your site has implemented templates, TrackRecord may fill in some of the fields for you. Refer to “Templates” on page 73 for information on how to set up and use TrackRecord templates.

3 If you are finished entering defects, click Save and Close on the toolbar.

The drop-down menu next to the **Save and Close** button offers two additional Save options:

- ◇ If you want to add an unrelated defect, choose **Save and New** from the Save menu on the toolbar.
- ◇ If you want to duplicate the current defect information as part of a new defect, select **Save and Duplicate** from the **Save** menu on the toolbar.

Changing the Status of an Item

From the time an item is opened to the time it is officially closed, it typically undergoes a series of status changes; these changes are collectively known as a workflow. For example, a new Defect typically starts with a status of Unassigned. A project manager assigns the Defect to a developer, updating its status to Assigned. If the developer confirms that the problem exists and fixes the appropriate software, he or she would move the Defect into a Pending Validation status, flagging it for testing and verification.

A workflow defines the states an item goes through and the privileges each user has to move an item from one state to another. Your TrackRecord administrator defines the workflow for items. For example, an administrator might specify that a Defect with a status of Unresolved can be moved by someone in Quality Assurance to a status of Needs More Information or to Resolved.

Not all items follow a workflow. The Person type, for example, might undergo a series of changes over time, but these changes do not constitute a change in the status of the item. Your TrackRecord administrator establishes which types of items will follow a workflow by placing a Status field on the type form. The Person type does not follow a workflow and does not contain a status field.

To Change the Status of an Item

- 1 Open the appropriate item and enter any changes in the fields on the form.**
- 2 Use the Action button on the toolbar to select an action.**

You can only select actions that your group is authorized to perform. When you choose an action, you change the status of the item. For example, if you open a Defect with a status of Unreviewed, the Action list may offer two choices: Review and

Close. If you choose Review, TrackRecord changes the status of the item to Open. If you choose Close, TrackRecord changes the item's status to Closed.

- 3 **When you are finished, click Save and Close.**

Merging One Item with Another

If you discover that your TrackRecord database includes duplicate items, you can merge them to remove the redundant information. When you merge one item (the *source* item) into another (the *target* item), TrackRecord replaces all references to the source item with references to the target item throughout the TrackRecord database, ensuring that database links are preserved. When the merger is complete and the appropriate information from the source item is moved to the target item, TrackRecord deletes the source item.

If the two items you merge contain lists of information, such as telephone numbers, the contents of the source item's lists are added to the corresponding list for the target.

To Merge Items

- 1 **Open the target item and the source item.**

You can open items from the File menu, from the Item Browser, or from an Outline Report.

- 2 **Click inside the window of the target item to make it active.**
- 3 **Choose Merge with Another Item from the Edit menu.**

The Target window (Figure 3-12) opens. The Target window displays the type and name of the item to which information will be transferred.

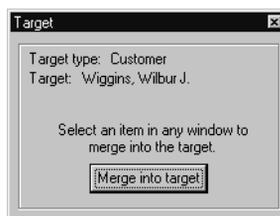


Figure 3-12 Target Window

- 4 **Click inside the window of the source item to make it active.**

- 5 **Click the Merge into Target button in the Target window.**
If the source item has information in lists that can be transferred to the target item, the Merge Items window (Figure 3-13) opens. Click **Yes** to transfer the item's list contents.

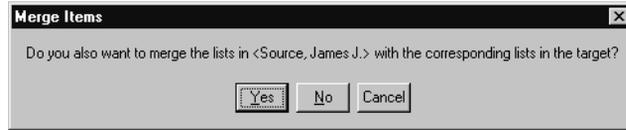


Figure 3-13 Merge Items Window

- 6 **When the Merge Complete dialog is displayed, click OK.**
TrackRecord merges the two items, closes the window for the source item, and deletes the source item from the database.
- 7 **Close the target item.**
- 8 **Close the Target dialog.**

Chapter 4

Using Queries and Views



This chapter describes how to display information from the TrackRecord database.

Queries and Views

To use information that has been entered into the TrackRecord database, you must first identify the information in which you are interested and the format in which you want the information displayed. You do this using queries and views.

- ◆ A **query** defines *what* is displayed. It lets you formulate requests for specific data from the database.
- ◆ A **view** defines *how* data is displayed. It determines the format in which the data is displayed. You can choose from four types of views:
 - ◇ The **Outline Report view** is a dynamic report in a flexible outline format that is automatically kept up to date.
 - ◇ The **Graph view** summarizes the results of Outline Reports graphically to simplify analysis of data statistics, distributions, and trends.
 - ◇ The **Milestone Status view** displays a summary of a development project's status with respect to important project milestones
 - ◇ The **Calendar view** lets you view time-critical information in a daily or weekly calendar format.

Each of these concepts is explained in more detail in this chapter.

Note If you are new to TrackRecord and want to create simple reports quickly, you can use the Header Wizard to display data without creating a query. See “Using the Header Wizard” on page 47.

About Queries

Queries let you set up criteria for retrieving information in the TrackRecord database. You select the type of item you want to find and add parameters to make your query as specific (or general) as necessary.

The query shown in Figure 4-1 returns all Defect items in which the Project field is equal to Bullwinkle.

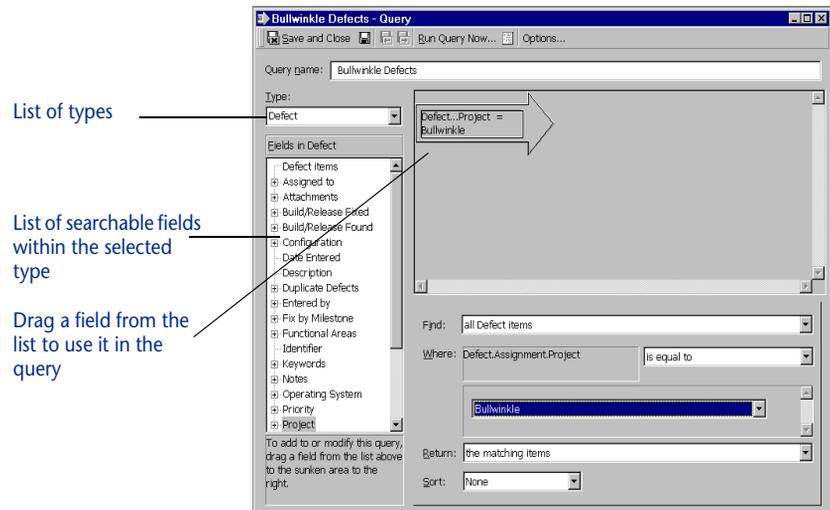


Figure 4-1 Simple Query

Each selection criterion is represented graphically as an arrow. Including more than one arrow in a query allows you to filter database information using several fields as selection criteria (Figure 4-2). The output from one arrow becomes the input set for the next arrow, until the end of the query chain is reached.

These query elements are logically “AND”ed. Items that meet both tests will be selected.

These query elements are logically “OR”ed. Items that meet either test (as well as other elements in the query) will be selected.

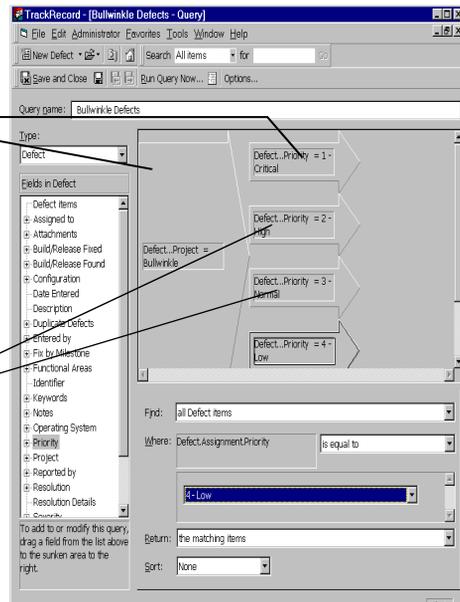


Figure 4-2 Query with Logical ORs and ANDs in Separate Arrows

When you click a query element (an arrow), the search criteria fields in the bottom right region of the Query become active. The criteria fields let you set specific search parameters (for example, *is greater than*, *starts with*, or *is equal to*). For date fields, you can use relative dates (*Today*, *1 week from today*) to make reuse of queries easier.

To Create a Query

- 1 Choose Query from the New Item drop-down list on the Toolbar.

Alternatively, you can open a new Query by choosing New/Query from the File menu (Figure 4-3).

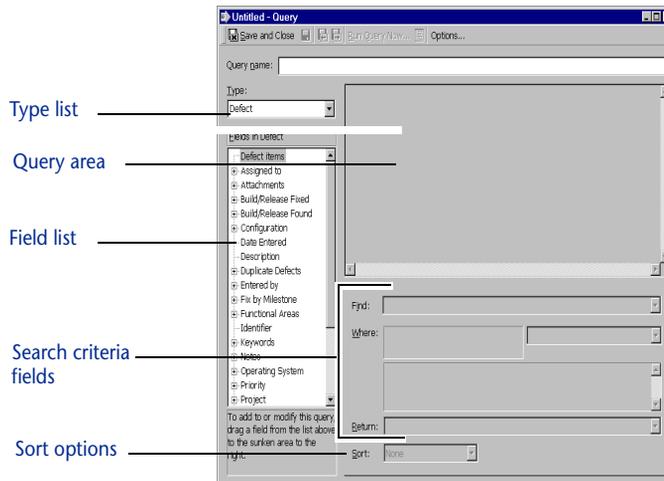


Figure 4-3 Query Form

2 Enter a name for this query in the Query Name field.

Query names can contain up to 39 characters.

3 From the Type list, select the type of item you would like to find.

For example, if you want to find all Defects in the TrackRecord database relating to a specific development project, you would choose **Defect**.

4 Drag the field representing your primary search criterion from the Field list to the Query area.

For example, to find the Defects relating to a specific project, drag **Project** to the Query area. An arrow identifying the search field (the query element) will appear in the Query area.

Notice that the search criteria fields become active.

5 Use the Find list to specify whether child types should be retrieved.

Normally, if you query a parent type such as Defect, the results include child types, such as DevPartner Studio Defect. If you want to exclude child types from the query results, choose **Excluding child types**.

6 Use the Where fields to specify the search criteria you want the query to use.

For example, to find Defects associated with the Colossus project you would select **Is Equal To** from the option list and then enter Colossus in the **Name** field. As you specify selection criteria, TrackRecord updates the text in the query element (the arrow).

7 Specify what you want TrackRecord to do when it finds an item that matches the search criteria.

Typically, you want the query to return all items that meet the search criteria you have entered. Alternatively, you can retrieve all items that link to anything meeting your search criteria.

8 Optionally, use the Sort list to specify how the query should sort the items it retrieves.

Note Alphabetization is not case sensitive.

- ◇ If you choose **Ascending**, TrackRecord sorts the selected items by the selection field you specified in alphabetical order (1, 2, 3,...x, y, z).
- ◇ If you choose **Descending**, TrackRecord sorts the selected items by the selection field you specified in reverse alphabetical order (z, y, x,...3, 2, 1).
- ◇ If you choose **None**, TrackRecord does not sort the selected items.

9 Repeat steps 3 through 8 to add additional query elements.

When using multiple query elements, the search criteria fields will reflect the currently selected element.

TrackRecord executes queries from left to right following the arrows in the queries.

10 If you want to share this query with other users or be notified when new items meet the selection criteria, click the Options button to display the Query Options window (Figure 4-4).



Figure 4-4 Query Options Window

- ◇ To share the query with other users, click the **Sharing** button. When the Sharing window opens, click **Share To** and select the share groups with whom you want to share the query. If you choose **Not Shared**, no one other than you can use the query.
- ◇ To receive notification through electronic mail when a new item meets the selection criteria for this query, click the **Email Me** check box. Note that your TrackRecord administrator must

configure AutoAlert and record your electronic mail address as part of your user profile for this feature to take effect.

If you choose to share a query and also click the **Email Me** checkbox, the e-mail will only be sent to you, not to members of Share Groups.

Click **OK** to close the Sharing dialog.

11 Optionally, click the Run Query Now button on the toolbar.

Running the query from within the Query dialog allows you to test your query. A search results window will display all items from the database that match your query. If the results do not match your expectations, you can modify the query.

12 Optionally, click the Create Outline Report button in the toolbar.

Since queries are most often used to create Outline Reports, this button provides a shortcut to that feature. Outline Reports are described in “Using Outline Reports” on page 37.

The new query is automatically saved and entered as a header in a new Outline Report. (This option is not available on Queries created from within Outline Reports.)

13 Save the new query by clicking Save and Close or by choosing Save from the File menu.

Query Composition Strategies

While TrackRecord provides great flexibility in the creation of queries, you should follow a few simple rules to ensure that your queries perform satisfactorily.

- ◆ Begin your query with the type of item you want to find. For example, if you want to find all the Defects for a particular Project, start by selecting the Defects data type, then add a query element for the Project field. If you start your query by asking for all Project items, the query would check all *Projects*, rather than Defects that *relate to* a given Project.
- ◆ Create many simple queries. You can later combine these queries into more complex data extraction operations by creating Outline Reports and attaching simple queries. (Using Queries in Outline Reports is described in “Using Outline Reports” on page 37.)
- ◆ Avoid complex queries and queries that return very large data sets.
- ◆ Choose a sorting option for a query.
- ◆ In a sequence of Query elements (arrows), start with the query elements you expect to return the fewest items.

- ◆ The fastest queries are often the ones that run on the entire database.

Using Outline Reports

Outline Reports (Figure 4-5) are dynamic reports that select, sort, and format summaries of the information in your TrackRecord database. When you create an Outline Report, you insert headers into the report and associate a query with each header. When you run the report, the queries are run and the matching information is displayed in columns in the report.

While you are viewing an Outline Report, TrackRecord periodically runs the queries associated with the report to ensure the contents of the report reflect the most current information in the database.

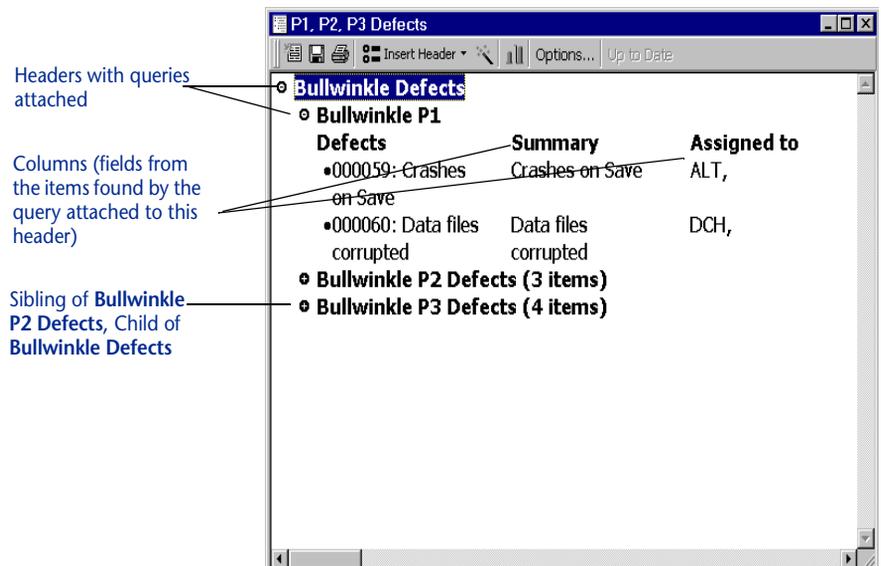


Figure 4-5 Sample Outline Report

The headers on an Outline Report are hierarchical. A child header can display only data that meets the Query conditions of all of its parent headers.

To collapse or expand a header, click the bullet to the left of the header's title. When a header is expanded to show its child headers and data items, its bullet appears as a white circle with a minus (-) sign inside it. When a header is collapsed, concealing its child headers and contents, its bullet appears as a black circle with a plus (+) sign inside. When a header is collapsed, the number of items it contains appears to the right of the header text; you can use this feature to create numeric

“summary reports” that identify data categories and item counts within each category.

You can specify what data is to be displayed in the columns of the report, rearrange the columns, and rearrange the headers in the report.

When you view a list of existing Outline Reports, the list shows both your privately-owned reports plus all other Outline Reports you are permitted to share.

The TrackRecord sample database provides several Outline Reports that demonstrate the use of headers and queries.

Creating a New Outline Report

Step 1: (Optional) Define Your Queries

If the queries you want to use in your Outline Report do not already exist, you should create them before you set up the Outline Report. Alternatively, you can create them as you add report headers. Refer to “About Queries” on page 32 for information on setting up queries.

Step 2: Add Report Headers

1 From the File menu, choose New/Outline Report.

An untitled Outline Report (Figure 4-6) opens.

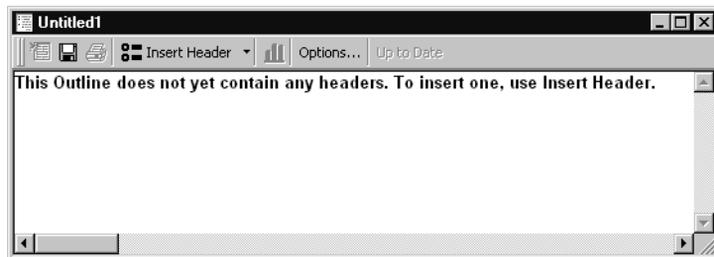


Figure 4-6 Blank Outline Report Form

- 2 Click the **Insert Header** button and select the **As a Sibling Header** option.

The Header Properties window (Figure 4-7) opens.

Note If you are new to TrackRecord or have not set up queries to be used in this report, you may prefer to use the Header Wizard. This option is described in “Using the Header Wizard” on page 47.

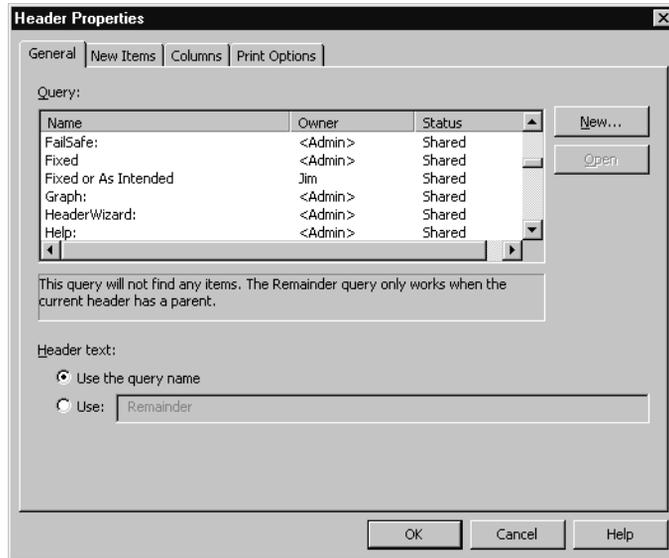


Figure 4-7 Header Properties Window

- 3 On the **General** tab, select the query you want for your top-level header or click the **New** button to create a new query.
To review the contents of an existing query, click the **Open** button. If you create a new query and use the **Run Query Now** button, note that the query results will not reflect the filters of the parent headers.
- 4 Click a **Header Text** option to use the query name or to enter more descriptive text for the header.
- 5 Click **OK** or click another tab to continue setting up the header.

Step 3: Set Up New Items Options

The options on the **New Items** tab are used to define custom behavior when a user right-clicks on a displayed header. It adds a **New Item** option to a header’s right-click menu and allows you to specify a

template to be used for new items. If you are new to creating Outline Reports, you do not have to enter anything on this tab.

- 1 **On the New Items tab, select the type of item to be added.**
- 2 **Select a template to be used when items are added.**
To learn about templates, refer to Chapter 5, “Templates” on page 73.
- 3 **Click OK or click another tab to continue setting up the header.**

Step 4: Set Up Report Columns

Use the Column tab to specify what columns are to be displayed for this header. (Alternatively, you can set up columns after a header has been added. Right-click the header and choose **Set Columns** from the shortcut menu to display the Columns tab.)

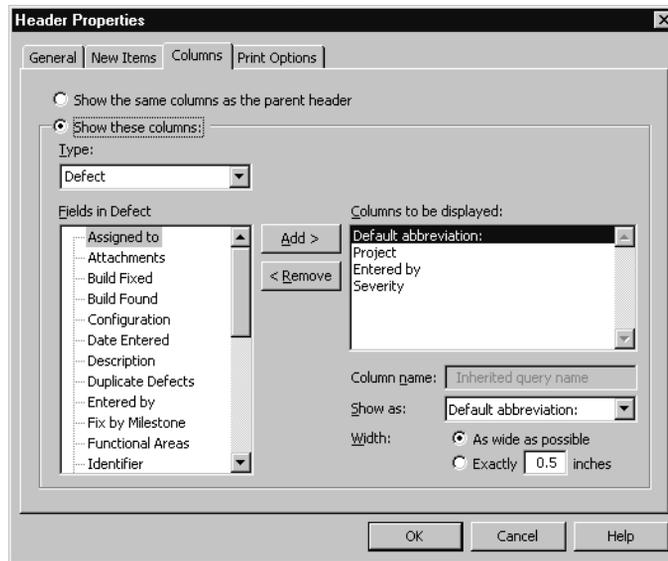


Figure 4-8 Columns Tab in the Header Properties Window

- 1 **To display the same columns as are displayed under the parent header, click the Show the Same Columns as the Parent Header option.**
- 2 **To customize the column display for the selected header, click the Show These Columns option.**
- 3 **Use the Type list to select the type of data in the header.**

- 4 **Select a field you want to display in the Outline Report and click the Add> button to move the field name into the Columns to be displayed list.**

The Default Abbreviation is always displayed. It is established by your Administrator and cannot be eliminated from the report.

- 5 **Repeat Step 4 for all columns to be displayed.**
- 6 **Specify the order in which you want columns displayed by dragging field names up and down within the Columns To Be Displayed list.**

For example, if you want Project to be displayed in the first column, drag it to the top of the list.

- 7 **Use the Column Name field to change the label the Outline Report displays at the top of the column.**
- 8 **Use the Show As list to specify how you want the column displayed.**

The available options vary depending on the field type. For a Name field, for example, you could choose to display the full name, initials, name and address, and so on.

- ◇ If the Administrator has created short and long versions of the Default Abbreviation, you can choose which to display. For example, the default abbreviation for a Defect is the identifier and summary. You could choose to display the short version, which is the identifier only, or the long version, which is the identifier, summary and full description.
- ◇ If you are setting up a column consisting of checkboxes, you can choose **Show X** to display the state of the checkbox (on or off) or **Show as lead checkbox** if you want the checkbox to appear in the first column of the report. TrackRecord allows any checkbox field from an item to appear in the Outline Report view as an active checkbox control. Clicking the checkbox in the Outline Report view toggles its state between on and off (similar to clicking the checkbox in the item or right-clicking an item and clicking the checkbox in the shortcut menu).

To include an active checkbox in an outline item, select **Leading Checkbox** in the **Show As** dropdown list. This moves the checkbox to the left column of the Outline Report.

- 9 **Specify how wide you want the selected column to be within the Outline Report.**
 - ◇ If you want TrackRecord to determine how much space to use, click the **As wide as possible** option.

- ◇ If you want the column to be a specific width within the report, click the **Exactly** option and enter the number of inches for the column width in the **Inches** text box.

10 When you have finished adding and formatting columns for the header, click OK or click another tab.

Step 5: Specify Print Options for the Report

Note When you print an Outline Report, TrackRecord prints items within open headers. If you collapse a header, the items under that header will not be printed.

You can specify print options for an Outline Report to override your default print option settings, and you can specify print options for specific headers within the report.

- ◆ To set print options for the whole report, click the **Options** button on the Outline Report toolbar and use the **Print Options** tab.
- ◆ To set print options for individual headers, right-click on the header, choose **Properties**, and use the **Print Options** tab.

The following steps are common to both report and individual header print options.

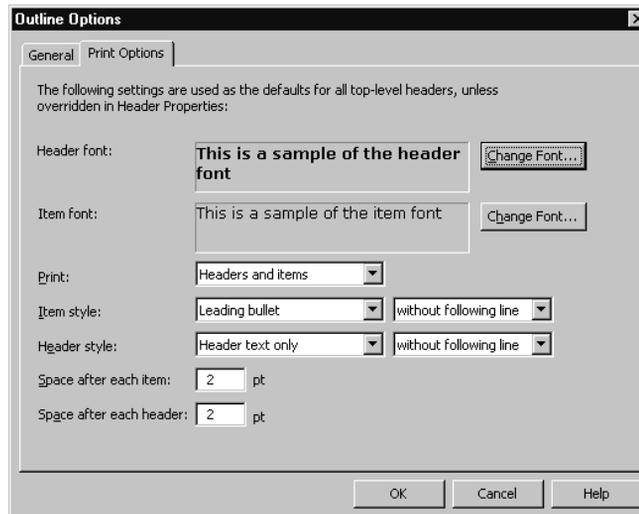


Figure 4-9 Print Options for a Header

- 1 **Select the type face and size you want to use to print report headers and data items.**

Note Headers and items must use the same type size, even if they use different type faces.

Click the **Change Font** button beside the **Header font** or the **Item font** field. When the Font window opens, select the type face, style, and size you want to use. When you are finished, click OK.

- 2 **From the Print list, specify whether the items belonging to report headers should be printed by selecting Headers and Items or Headers Only.**
- 3 **If you are printing data items, use the Item Style list to select the lead-in character for item entries.**
 - ◇ Choose **Leading Bullet** if you want each item to be preceded with a bullet (•).
 - ◇ Choose **Leading Number** if you want each item to be preceded with a sequence number. Sequence numbers restart within each header.
 - ◇ Choose **Leading Checkbox** if you want each item to be preceded with an empty checkbox. Checkboxes simplify tracking which items within a list have been discussed or completed.

Select whether to print with or without a line following each item.

- 4 **Use the Header Style list to specify how you want headers on the report printed.**

Select **Header Text Only** or **Header Text and Item Count**.

Select whether to print with or without a line following each header.

- 5 **Use the Space After Each Item and Space After Each Header fields to specify how much vertical space you want between lines.**
- 6 **If you are setting print options for an individual header, choose whether to start this header on a new page and, if this is a child header, whether to use the parent header's print options.**
- 7 **When you are finished, click OK.**

The changes will not be reflected on your screen display; they are used only when reports are printed.

Adding Multiple Headers

To add more headers to your reports, select a header in the report and click the **Insert Header** button.

- ◆ To add headers at the same level as the one you selected, choose **As a Sibling Header** from the **Insert Header** menu.
- ◆ To add child headers, choose **As a Child Header** from the **Insert Header** menu. Child headers act to further segment the data selected by the parent header. The selection criteria of the parent header limits the data available for selection by the child header.

Dragging Headers in an Outline Report

You can easily rearrange your Outline Report by dragging headers from one location to another within the report. When the cursor is over an area where the header can be dropped, the dragged image will “snap in,” showing you where the dropped header will appear. You can drag one header above or below another as a sibling, which puts them at the same level within the report. You can also drag one header below another to position it as a child or drag one header above another to make the dropped header a parent header.

If it is not possible to drop a header at a given location, the header will not snap in at that location, or will snap to the closest alternative.

Opening Items in an Outline Report

Each item entry in an Outline Report represents an item in the TrackRecord database. To open an item, double-click its entry in an Outline Report. If you want to update the information displayed for the item, make the appropriate changes and close the form; click **Yes** to save your changes.

Creating New Items from an Outline Report

You can create new items from within an Outline Report. Refer to “Step 3: Set Up New Items Options” on page 39 for information about establishing the type of items that will be created and the template to be used.

To Create a New Item from Within an Outline Report

- 1 **Select the header under which the item will be created.**

2 Right-click and select the New *itemtype* option.

The *itemtype* is the default type of item included in this header. An item form opens, allowing you to create a new item.

Using the Remainder Header

TrackRecord includes a special header called Remainder, which acts as a *None of the Above* summary item. When you include a remainder header as a child header, it gathers and summarizes all the items belonging to the parent header that have not been listed in the child headers above it. The Remainder header is typically used to identify items belonging to a parent header that fail to match criteria for sections under that parent header.

By moving the Remainder header within a parent header, you can summarize the “none of the above” items quickly and easily. Figure 4-10 illustrates how you can move the Remainder header from one location to another within an Outline Report to obtain different summaries. The first screen summarizes the unfixed bugs filed against a product, sorted by priority, with the number of bugs of each priority shown in parentheses beside the child header. The Remainder header indicates that the product has 36 unfixed bugs whose priority is not 1, 2, 3, or 4.

When the Remainder header is moved above the Priority 4 header in the second screen, the item count for Remainder changes from 36 to 120. The Remainder header is now counting the unfixed bugs it was counting before (36) plus the unfixed bugs in the Priority 4 category (84). Similarly, when the Remainder header is moved above the Priority 3 header, the item count for Remainder changes to 299, representing the number of items of Priority 3 (179) plus the number of items of priority 4 (84) plus the number of items it represented in the first screen (36).

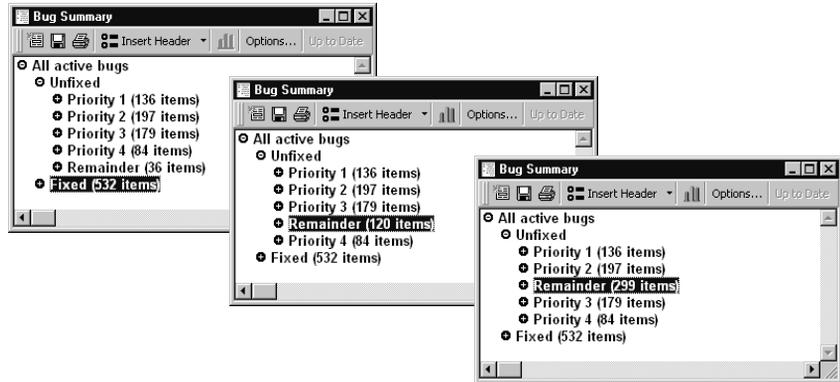


Figure 4-10 Using the Remainder Header

Using the Shortcut (Right-Click) Menu

When you are working in an Outline Report, you can use the shortcut menu to perform a variety of functions. To open the shortcut menu, right-click an outline header or item.

As Figure 4-11 illustrates, the contents of the shortcut menu change according to the type of header or item you right-click. When you click an item in an Outline Report, the shortcut menu displays options for adding a new item, opening, printing, or deleting the item you right-clicked, or merging the selected item with another item. When you click a header in an Outline Report, the shortcut menu displays options for adding a new item, opening, deleting, or changing the query associated with the header, organizing templates and report columns, graphing the data in the header, and other tasks.

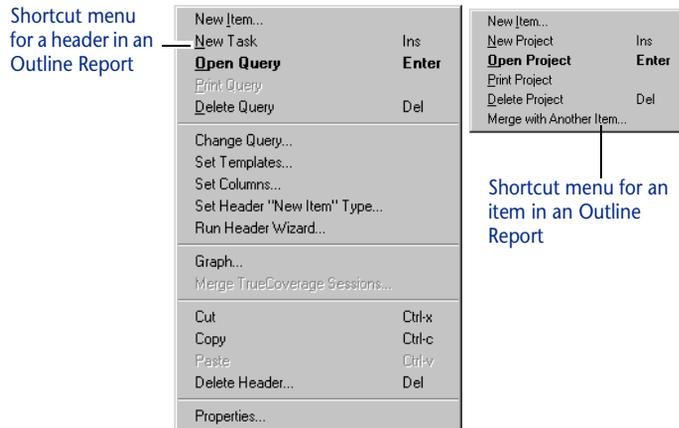


Figure 4-11 Shortcut Menus for Headers and Items

Using the Header Wizard

The Header Wizard is a TrackRecord utility that simplifies the creation of Outline Reports. The Header Wizard is most useful in two situations:

- ◆ When you are a new TrackRecord user unfamiliar with creating queries and inserting headers
- ◆ When you want to create Outline Reports that would require multiple, repetitious queries

For example, to create a report listing tasks assigned to certain team members, you would have to create a query element for each person (AssignedTo=Bob, AssignedTo=Mary, AssignedTo=Joe, etc.). To group the tasks under separate headers would be even more tedious, since you would have to create separate queries. With the Header Wizard, you could identify all team members at one time by selecting them from a list and specifying how you want the tasks grouped.

To Use the Header Wizard

- 1 **From the File menu, choose New/Outline Report.**
- 2 **Click the Insert Header button on the toolbar and choose Using Header Wizard.**

The Introduction window (Figure 4-12) opens.

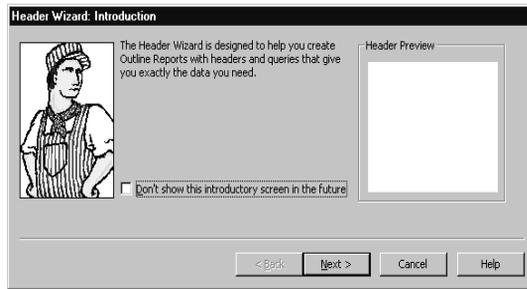


Figure 4-12 Header Wizard Introduction Window

Read the instructions on the Introduction window. If you want to skip the Introduction window when you run the Header Wizard in the future, click the **Don't Show This Introductory Screen** check box.

When you are finished, click the **Next** button.

- 3 **In the Select Item Types window (Figure 4-13), click the check box beside each type of item you want to retrieve from the TrackRecord Database.**

The Header Preview pane displays the name of each type you select.

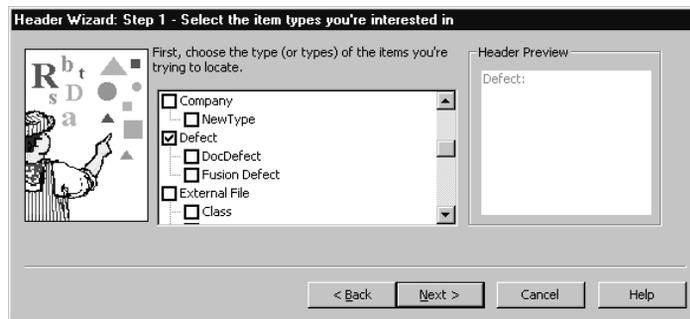


Figure 4-13 Select Item Types Window

- 4 **When you are finished, click Next.**
- 5 **In the Choose Field to Query On window (Figure 4-14), specify the field you want to use as your query criterion.**

To find all items, click the **Find All Items** option. To search for items that have a specified value in a field, click the **Query On Field Named** option and select the field you want to use. For example, if you want to look for Defects associated with a specific project, you would choose Project as the query field.

If you choose more than one type in Step 3, only those fields that are common to all chosen types are displayed.

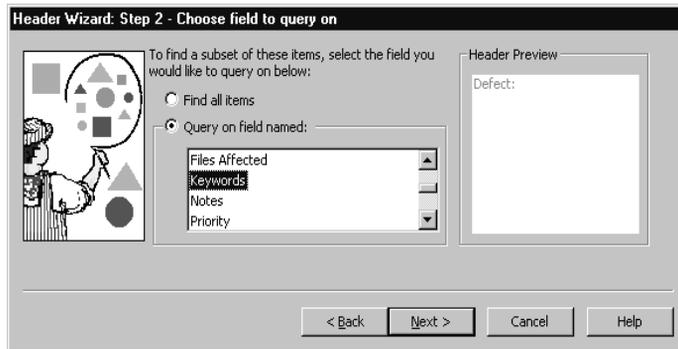


Figure 4-14 Choose Field to Query On Window

- 6 When you are finished, click Next.
- 7 If you selected a query field in Step 5, use the Choose Specific Values To Find window (Figure 4-15) to specify the values you want the query to find.

If you want the query to search for more than one value, click each value you want. For example, if you want to look for Defects associated with three specific projects, you would click each project name. To deselect an entry, click it again.

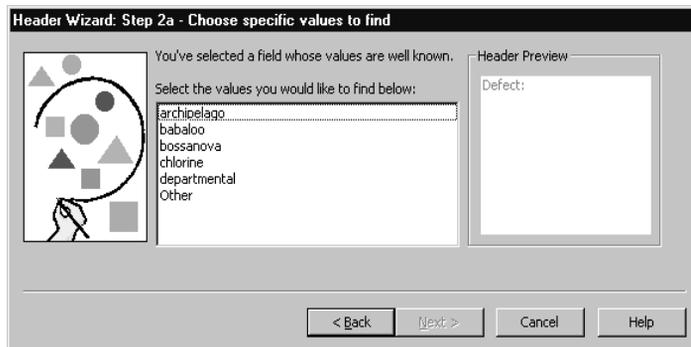


Figure 4-15 Choose Specific Values to Find Window

- 8 When you are finished, click Next.
- 9 In the Grouping window (Figure 4-16), specify how you want the query results to be grouped.
 - ◇ To group selected items under separate headers, click the **Group Them Under Multiple Headers** option.
 - ◇ To group selected items under one header, click the **Put Them All Under One Header** option.

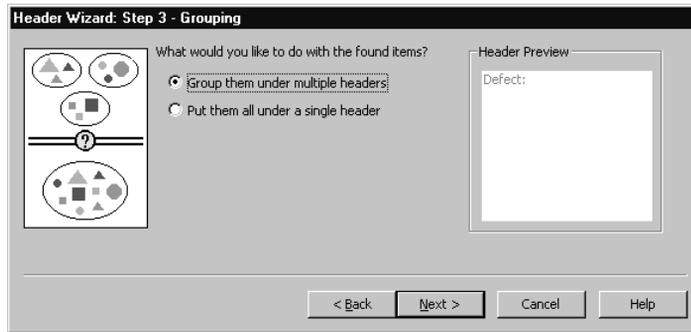


Figure 4-16 Grouping Window

10 When you are finished, click Next.

11 If you specified in Step 9 that you wanted selected items grouped under separate headers, use the Grouping Options window (Figure 4-17) to specify how TrackRecord should group items. Options are:

- ◇ Group by Type – Put all items of the same type together, but do not sort them.
- ◇ Group by Value – Put all items with the same value together, but do not sort them.
- ◇ Group by Type, Then By Value – Put all items of the same type together and sort them by value.
- ◇ Group by Value, Then By Type – Put all items with the same value together and sort them by type.

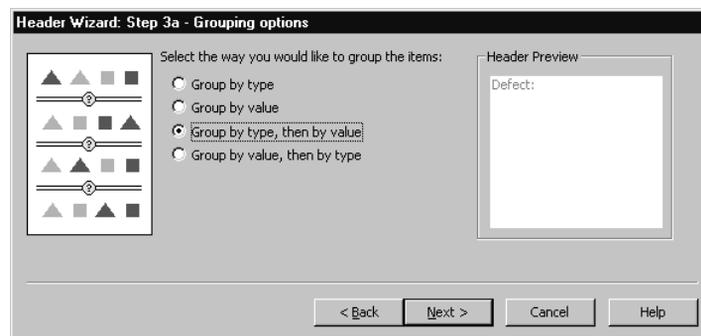


Figure 4-17 Grouping Options Window

12 When you are finished, click Next.

13 In the Placement Options window (Figure 4-18), specify where the new headers should appear on the Outline Report.

- ◇ If new headers should appear at the same level as the current header, or if these are the only headers in the report, click the **Insert as a Sibling** option.
- ◇ If new headers should appear as subheaders within the current header, click the **Insert as a Child** option.

When you are finished, click **Finish**. TrackRecord formats the Outline Report with the new headers, retrieves the items that meet the header search criteria, and displays the selected items under the appropriate headers.

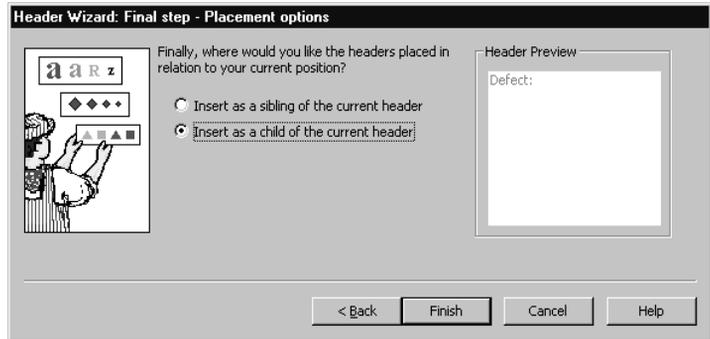


Figure 4-18 Placement Options Window

You can use the Header Wizard to insert as many headers as desired in your report.

Using the Graph View

The Graph view (Figure 4-19) lets you display a visual summary of the information in an Outline Report. Depending on the information in the Outline Report, you can generate pie charts, simple bar graphs, clustered or stacked bar graphs, line graphs, area graphs, and trend graphs. For example, you could generate a bar graph to illustrate the number of open bugs assigned to different developers or a pie chart to illustrate what proportion of open bugs have an Urgent priority.

Reports with symmetric, hierarchical headers produce the most meaningful graphs.

After a graph is generated, TrackRecord updates it automatically when its underlying data changes. However, the graph does not change if you modify the Outline Report view from which the graph was generated.

You can print and save Graph views just as you do any other view. You can copy graphs generated by TrackRecord into graphics applications, such as Microsoft Paint, or into text editors, spreadsheet programs, and

electronic mail applications that can accept bitmap images, such as Microsoft Word, Microsoft Excel, or Microsoft Outlook. For more information, refer to “TrackRecord and Other Applications” on page 87.

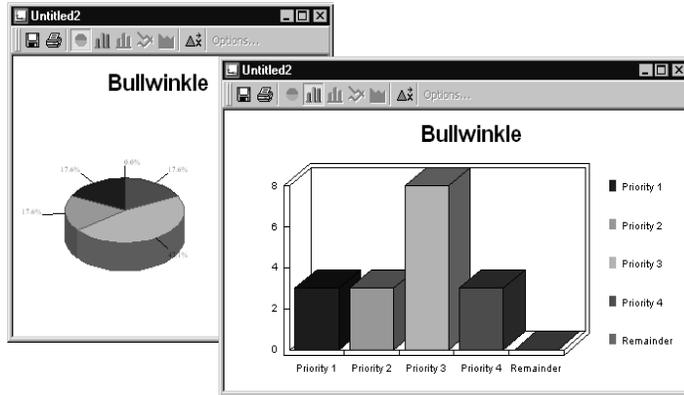


Figure 4-19 Sample Graph Views

Creating Graphs from an Outline Report

To create a graph from an Outline Report, select the appropriate header in the report and click the **Graph** button on the toolbar (Figure 4-20). Alternatively, right-click the appropriate header and choose Graph from the shortcut menu.

If the information in the header is not appropriate for graphing, the Graph button and option will not be active.

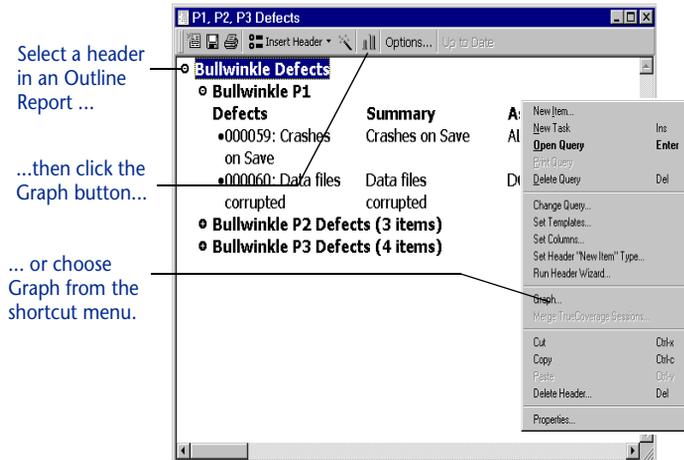


Figure 4-20 Creating a Graph from An Outline Report

Formatting Graphs

TrackRecord automatically selects a graph style based on the structure of the Outline Report and the contents of the queries in the outline. You can override the default graph style by choosing a different graph icon from the Graph toolbar (Figure 4-21). The graph style toolbar buttons (pie chart, bar graph, stacked bar graph, line graph, area graph, and trend graph) become active when the style is appropriate for the data being graphed.

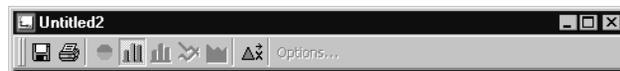


Figure 4-21 Graph Toolbar

- ◆ To select a different graph style, click a graph style button on the **Graph** toolbar or choose a graph style entry from the **View** menu.
- ◆ For line graphs, you can add a line identifying the overall trend by choosing **Best Line Fit** from the **View** menu. Similarly, you can insert vertical bars connecting graph points to the baseline by choosing **Draw Bars to Points** from the **View** menu.
- ◆ To choose whether to display a legend for your graph, choose **Legend** from the **View** menu.

Using the Milestone Status View

When project managers develop project plans, they typically break the project into manageable segments or development cycles. They identify target dates for completion of these phases of the development project and define criteria for entry and exit from each milestone. Once development is underway, milestones become the focus of the team efforts; team members work to perform the tasks associated with upcoming milestones.

The Milestone Status view (Figure 4-22) identifies where a development project stands with respect to a predefined milestone, such as an Alpha or Beta release date. Individual developers can use the Milestone Status view to identify the specific assignments they have to complete to meet an upcoming deadline. By adding database queries identifying which tasks and bugs have been assigned, individual developers can use a personalized Milestone Status view as a To Do list that highlights tasks remaining to be performed.

Similarly, project managers can use the Milestone Status view to get a summary of the overall project status. They can leverage code volatility statistics to identify how much of the code is changing from one build to the next and coverage statistics to identify how much of the project source code has been tested. By adding queries to the standard report, a project manager can determine critical information, such as the status of the project's requirements and software defects.

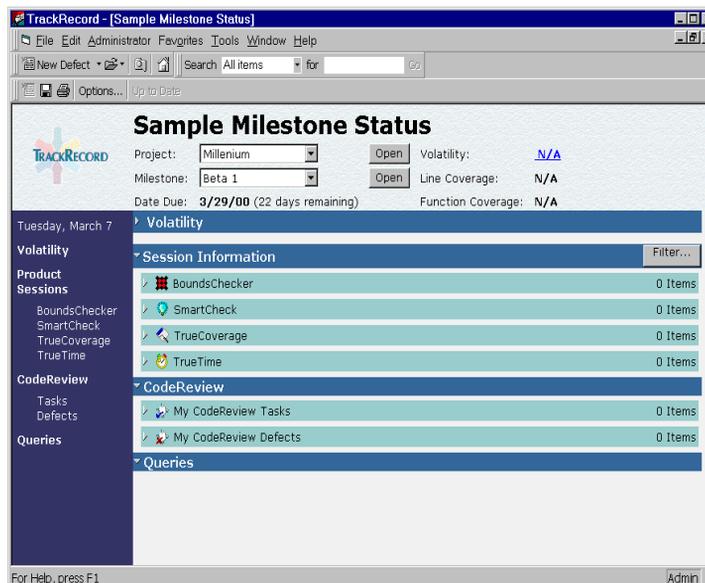


Figure 4-22 Sample Milestone Status View

Note Some parts of the Milestone Status view are product-specific. If you are not using the tool that populates a particular section of the view, that section will not contain data.

A Milestone Status view consists of five types of information:

- ◆ The **header information** section identifies the project, milestone and date due. When used with the DevPartner tools, the header also summarizes the volatility and coverage statistics for the project.
 - ◇ **Volatility** measures the number of functions that were added, removed, or changed between builds over the total number of functions in both builds.
 - ◇ **Line Coverage** measures the number of lines that have been tested over the total number of lines of code in the components that have been run under TrueCoverage.
 - ◇ **Function Coverage** measures the number of functions that have been tested over the total number of functions in the components that have been run under TrueCoverage.
- ◆ The **Volatility** section displays a graph depicting how much of the project has changed from one build to another. Click the triangle on the left side of the Volatility header to display the volatility graph for the selected milestone. This section is populated when TrackRecord is used with the DevPartner tools.
- ◆ The **Session Information** section displays testing data generated from applications that submit session information, such as NuMega BoundsChecker, SmartCheck, TrueTime, or TrueCoverage. Click the triangle on the left side of the Session Information header to display statistics for each component. Click the triangle to the left of a component name to display the items belonging to that component.
- ◆ The **CodeReview** section displays the tasks and defects identified by NuMega CodeReview.
- ◆ The **Queries** section displays the results of queries included in the Milestone Status view.

Note Milestone Status views depend on certain internal tags. If these tags have been removed or altered, the Milestone Status option might not be available. Administrators should refer to Chapter 11, “TrackRecord Configuration,” for more information on ActiveLink Tags.

To Create a New Milestone Status View

- 1 From the File menu, choose New/Milestone Status to create a blank Milestone Status view.
- 2 Use the Project list to select the project you want to profile.
Optionally, click the Open button beside the Project list to display the project details.
- 3 Use the Milestone list to select the project milestone you want to profile.
Optionally, click the Open button beside the Milestone list to display the milestone details.
- 4 Click the Options button on the Milestone Status toolbar to open the Milestone Status Options window (Figure 4-23).

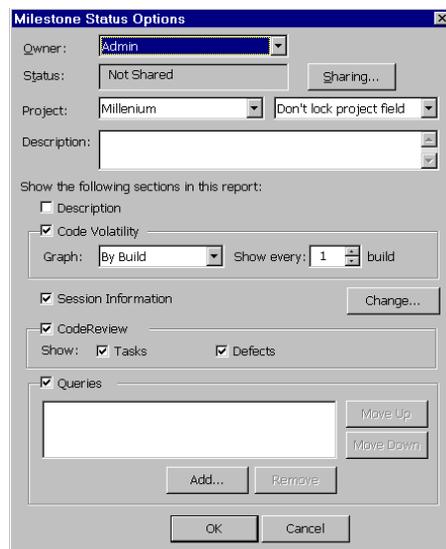


Figure 4-23 Milestone Status Options Window

- 5 To share your Milestone Status view with other users, click the **Sharing** button.

When the Sharing window (Figure 4-24) opens, click the **Share To** button and click the checkbox beside each share group with whom you want to share the report. Click **OK** when you are finished.

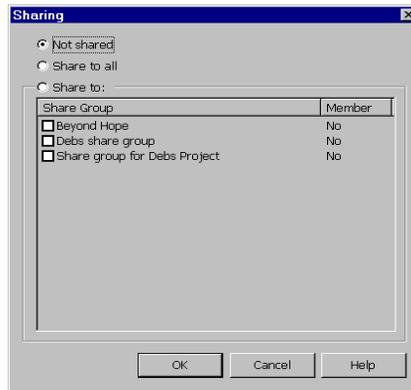


Figure 4-24 Milestone Status Sharing Window

- 6 Use the **Project list** to select the project you want to summarize in the Milestone Status view.

To prohibit other users from changing the project being displayed, choose **Lock Project** field from the list to the right of the Project list.

- 7 Enter a brief description of the report in the **Description** field.

To display the report description on the report, click the **Description** checkbox.

- 8 To display code volatility information graphically, click the **Code Volatility** checkbox and select whether the graph should depict volatility over time or across builds.

- ◇ If you choose **By Build**, you can use the **Show Every** control to specify how many builds back you want to compare to the previous build. For example, if you choose 1, TrackRecord compares the volatility of the current build against the previous build. If you choose 3, TrackRecord compares the volatility of the current build against the software three builds ago.
- ◇ If you choose **By Date**, you can specify whether you want a graph of changes in code volatility for each day, for each week, for each two-week period, or for each month.

- 9 To display information passed to TrackRecord from other applications, click the **Session Information** checkbox, click

Change and click the checkbox beside each component whose session information is to be displayed.

Click **OK** to close the Products dialog.

10 To include CodeReview information in the Milestone Status view, click the CodeReview checkbox.

To see CodeReview tasks, click the **Tasks** checkbox. To see CodeReview defects, click the **Defects** checkbox.

11 To include queries in the Milestone Status view, click the Queries checkbox.

To add a query to a Milestone Status view, click the **Add** button. When the **Open Query** window opens, choose the query you want to include in the Milestone Status view and click **Open**.

12 To change the sequence in which queries are displayed in the Milestone Status view, select a query and click the Move Up or Move Down buttons.

To delete a query from the Milestone Status view, select it and click the **Remove** button.

13 When you are finished, click OK.

Filtering Milestone Status Data

The data that accumulates over the course of a development project provides a rich source of information about the project. Answering specific questions may require the filtering of session information.

You can set up a filter to isolate specific information for display. For example, you might want to see only those sessions entered by Dan Developer.

You can filter session information by one or more of the following criteria:

- ◆ A single developer
- ◆ A particular build
- ◆ A specific application component
- ◆ A date
- ◆ An operating system
- ◆ An amount of memory

For example, you could filter the session information to discover how much debugging had been done on machines configured with more than 32 Megabytes of memory by setting the filter to **Memory is greater than 32**. TrackRecord searches the Hardware Configuration items in its database, retrieves a list of machine names whose

configurations have more than 32 megabytes of memory, and then shows only those sessions run on the retrieved list of machines.

To Filter Sessions in a Milestone Status View

- 1 **Open the appropriate Milestone Status view.**
- 2 **Click the Filter button on the Session Information header.**

The **Filter Sessions** window (Figure 4-25) opens.

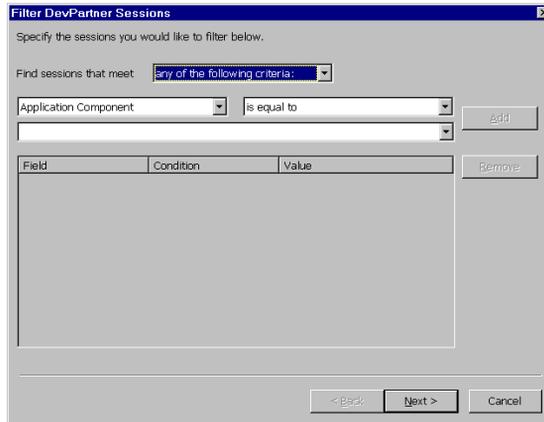


Figure 4-25 Filter DevPartner Studio Sessions Window

- 3 **Specify the types of sessions you want to retrieve and click Next.**
- 4 **On the second Filter Sessions window, select the sessions you want to include in the Milestone Status view.**
Click the **Select All** button to select all projects.
- 5 **When you are finished, click the Set Filter button.**
The Milestone Status view displays information for the sessions you selected.

Using the Volatility Statistics Windows

The Volatility field in the header section of the Milestone Status view displays a percentage value that represents the number of functions that were added, removed, or changed between the current build and the previous one. This data is available when TrackRecord is used with the NuMega DevPartner tools.

If you click the Volatility percentage, TrackRecord displays the **Volatility Information** window, which contains three tabs:

- ◆ The **Summary** tab (Figure 4-26) displays a count of the number of functions in a project that have been added, removed, and changed between the current build and the previous one.

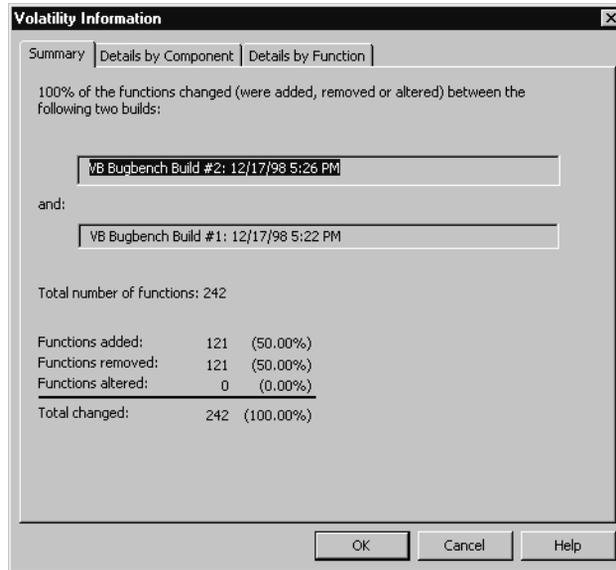


Figure 4-26 Volatility Information Window – Summary Tab

- ◆ The **Details by Component** tab (Figure 4-27) identifies the components, source files, and functions that have changed between one build and another. Volatility statistics reported here may differ from volatility statistics reported by TrueCoverage, because the volatility calculations in DevPartner Studio include new or deleted functions in the count of changed functions.

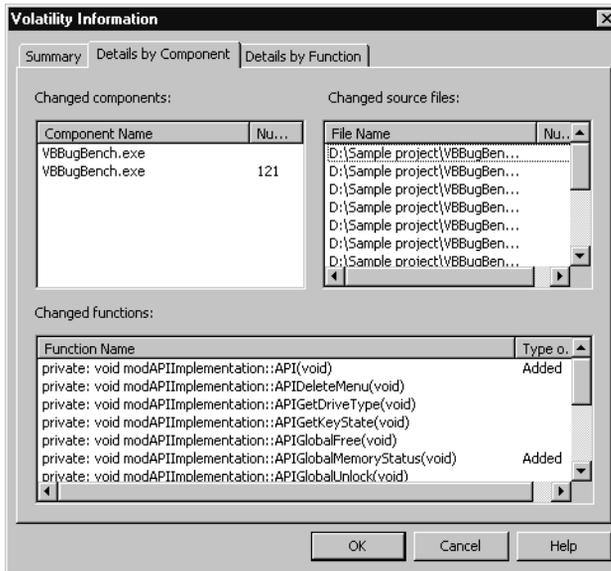


Figure 4-27 Volatility Information Window – Details by Component Tab

- ◆ The **Details by Function** tab (Figure 4-28) displays the list of functions that have been added, modified, or removed within a project’s source tree between one build and another.

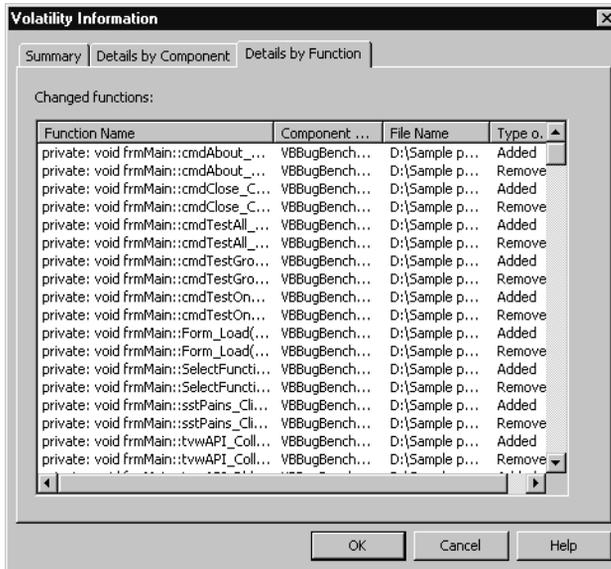


Figure 4-28 Volatility Information Window – Details by Function Tab

Interpreting the Volatility Graph

You can display a graphical summary of the volatility between builds by opening the Volatility section of the Milestone Status view. To open the Volatility section, click the triangle to the left of the Volatility header.

The point plotted above each build label represents the percent change between that build and the previous one. If you specified that you wanted to display volatility across more than one build at a time, the plot pointed above the label represents the percent change between that build and the appropriate earlier build.

Using the Calendar View

To view project information in the context of some span of time, use TrackRecord's calendar view. The Calendar view (Figure 4-29 and Figure 4-30) functions as an appointment book and To Do list, where TrackRecord automatically categorizes and displays the appropriate data by date and time.

By default, TrackRecord displays all uncompleted items for a given day or week. Timed Tasks are displayed according to their scheduled start date/time, and Tasks (including Defects and Feature Requests) are displayed according to their due date. Timed Tasks take place at specific times and for specific periods. Regular Tasks may have deadlines, but are otherwise open-ended.

The data displayed in a calendar is determined by the query run by the calendar view. You can customize the information displayed by your calendar by changing the query used by the calendar.

Depending on your preferences, the daily or weekly Calendar view displays a simple list of your appointments and tasks or a tick-marked schedule of each day. By changing the Calendar view's query, you can display different data or dates. You can also maintain and display multiple calendars simultaneously.

Drag open a new time bar to create a new item for a selected day; also works in Week view

Can display list of appointments and items or daily calendar

Displays appointments (Timed Tasks) and To Do items (Tasks)

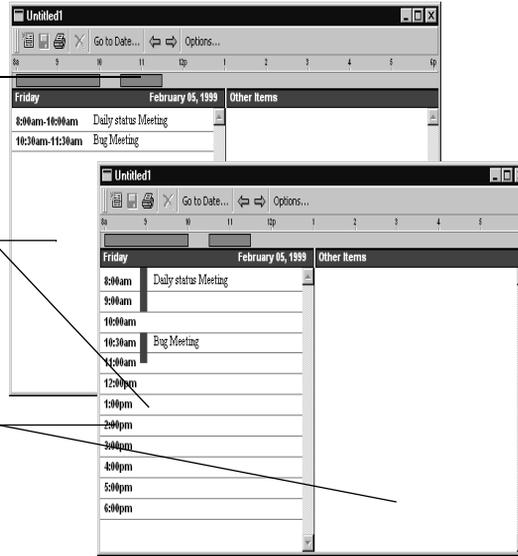


Figure 4-29 Sample Calendar by Day Views

Change appointment times dynamically by dragging handles or entire time bar

Can display list of appointments and items or weekly calendar

Double-click titlebar to display Day view

Double-click an item to see its Item view; right-click to inspect

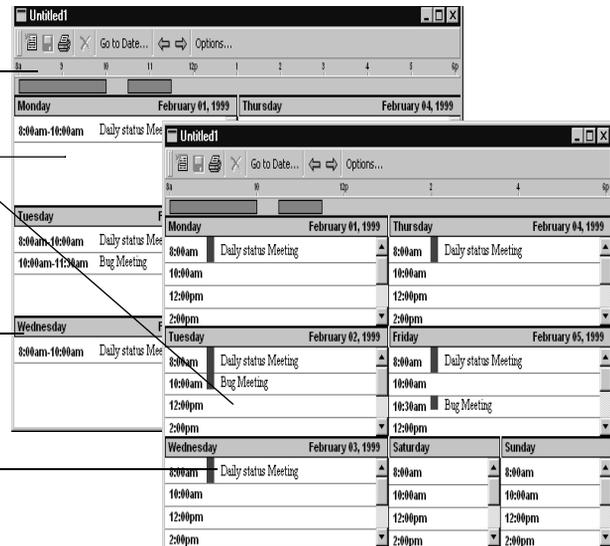


Figure 4-30 Sample Calendar by Week Views

Setting Up Your Default Calendar

The items displayed in a calendar are determined by the query run by the calendar view. Before setting up your default calendar, create a query for the items that you want displayed. Refer to “About Queries” on page 32 for information about creating queries.

To Set Up Your Default Calendar

- 1 From the File menu, choose the New/Calendar option.
- 2 Click the Options button.

The Calendar Preferences window (Figure 4-31) opens.

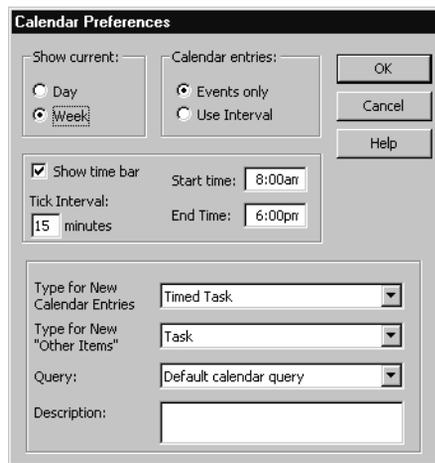


Figure 4-31 Calendar Preferences Window

- 3 Use the Show Current options to specify whether you want to see a calendar for a day (see Figure 4-29) or for a week (see Figure 4-30).
- 4 Use the Calendar Entries options to specify whether you want to see calendar slots for which nothing is scheduled.
- 5 Use the Show TimeBar checkbox to turn the time bar at the top of the calendar on or off.
- 6 Use the Tick Interval field to specify the interval for time slots and to set tick marks on the time bar.
- 7 Use the Start Time/End Time fields to specify the time range displayed in the Calendar.

You can enter tasks and appointments outside the range established by the Start Time/End Time fields.

- 8 Use the **Type for New Calendar Entries** list to specify the default type for appointments.
- 9 Use the **Type for New “Other Items”** list to specify the default type for other time-dated queries, such as due dates for project milestones.
- 10 Use the **Query** list to specify the query the calendar uses to retrieve information.

The default calendar query displays all tasks and timed tasks due on a day that have not been marked as completed. You can choose any query that scans for time or date fields. The query you select will be used every time you open a new Calendar view. For example, you might select **Tasks Due This Week** as the default query for your Calendar view.

- 11 **Optionally, enter a description of the calendar in the Description field.**
- 12 **Click OK. Your calendar view is displayed.**

TrackRecord stores your calendar view settings and uses them any time you display a new calendar.

Opening Your Calendar View

You open your TrackRecord calendar by choosing **Calendar** from the **File/New** menu. TrackRecord uses your default Calendar settings to display your current calendar.

To Add a Task to a Calendar

Note If the query specified in your Calendar Preference dialog does not search for date and time fields, you will not be able to enter an appointment in this way. The Default Calendar Query uses appropriate fields.

- 1 **From the File menu, choose the New/Calendar option to open your Calendar view.**
- 2 **Click the point on the timebar representing the starting time of the task or appointment, and drag across to the ending time. Release your mouse button.**

For example, if you are entering an appointment scheduled to start at 9 AM and end at 10 AM, you would drag from the 9 to the 10 on the timebar. The **Timed Task** form (Figure 4-32) opens.

Figure 4-32 Timed Task Form

- 3 Enter information about the appointment, such as a one-line summary (which will appear in your calendar), in the form fields.

TrackRecord fills in the starting and ending date and time fields for your appointment automatically.

- 4 When you are finished, click the Save and Close button.
The new appointment appears on your calendar.

Opening, Modifying, and Deleting Queries and Views

There are several ways to open an existing query or view. You can use the **Open** button on the Toolbar, the **Open** option on the **File** menu, or the **Manage** option on the **File** menu, which is the method used in the following steps.

To Open an Existing Query or View

- 1 From the TrackRecord File menu, select the Manage option.

- 2 From the displayed list, select the type of view, query or template to be managed.

The Manage dialog displays. Figure 4-33 illustrates the Manage Outline Reports dialog.

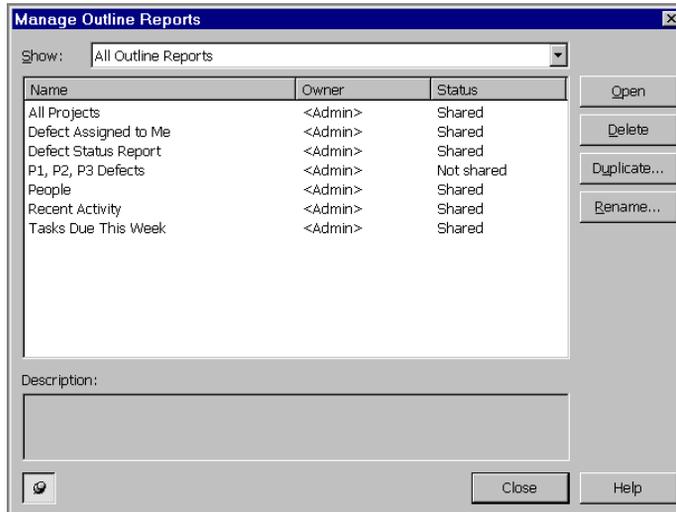


Figure 4-33 Manage Outline Reports Dialog

If the list of queries or views is large, you can use the **Show** control to display a subset of the list. You can show all, show those belonging to you, or show those you can access through share groups.

- 3 Use the buttons to **Open**, **Delete**, **Duplicate** or **Rename** the selected query or view.

Identifying Favorite Reports and Home Pages

Favorite Reports are reports or views that you use frequently. Identifying a view or report as a Favorite Report lets you open it from the **Favorites** menu instead of through the **File/Open** menu.

A Home Page is a report that opens automatically when you start TrackRecord, and can be accessed quickly through the **Home Page** button on the Toolbar. If you are using the WebServer browser interface, your Home Page is accessible through a button in the control area.

Project administrators can create one or more favorite views for new users during TrackRecord setup and can set default Home Pages. You can modify these settings.

To Add a Report To Your Favorites Menu

- 1 Open the report or view you want to identify as a favorite.
- 2 With the report window active, choose **Add to Favorites from the Favorites menu**.

The title of your report appears on the Favorites menu.

To Change Your Home Page

- 1 Open the report or view you want to identify as your Home Page.
- 2 With the report window active, from the Favorites menu select the **Set as Home Page** option.

Searching the TrackRecord Database

To find database information quickly, you can perform a search rather than create a query and view. Simple searches find text strings that occur in an item; advanced searches find text strings in specific fields.

Performing a Simple Search

A simple search lets you find items of a specified type that contain a specified text string. For example, you can search for all defect items that contain the name Edison Carter in any field.

To Perform a Simple Search

- 1 **Select Search from the Tools menu. The Search window (Figure 4-34) opens.**

You can use the **Search** tool on the main TrackRecord window to perform the same function as the **Search** menu option.

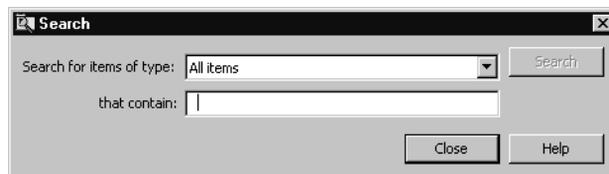


Figure 4-34 Search Window

- 2 **Select the type of item you want to find.**
 - ◇ To search for a specific type of item, pull down the **Search for Items of Type** list and choose the type you want.

- ◇ To search through the entire database, choose **All Items** from the **Search for Items of Type** list.
- 3 Enter the text string you want to find in the That Contain field.**

Any item that contains the specified text string in any field will be displayed. You can enter partial words to find wording variations. For example, you can enter **veri** to find all items that include the words *verification*, *verify*, *verified*, or *unverified*.
- 4 Click Search.**

A counter displays in the left corner of the status bar, indicating the progress of the search.

The Search window lists all items that meet your search criteria. To display the contents of one of the items, double-click the appropriate entry.
- 5 When you are finished, click Close.**

Performing an Advanced Search

Advanced searches let you refine the criteria you want TrackRecord to use to find information in the database. Unlike a standard search, an advanced search lets you specify the fields to check for text strings and lets you set up multiple search criteria. For example, you can perform an advanced search to find all defects where the project name is Forbin, the severity level is 3, and the description includes the phrase “only happens.”

To Perform an Advanced Search

- 1 Select Advanced Search from the Tools menu.**

The **Choose a Type** window (Figure 4-35) displays your list of standard database types.

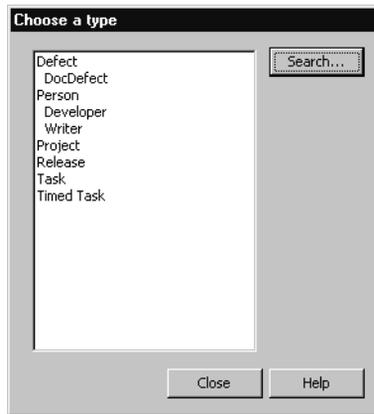


Figure 4-35 Choose a Type Window

- 2 **Select the type of information you want to find and click Search.**

TrackRecord opens the form associated with the type of information you selected. For example, if you selected Defect, TrackRecord would open the Defect form.

- 3 **Use the fields and controls on the item form to specify your search criteria.**

For example, you can use the Project list to select the project for which you want to find Defect items. Similarly, you can enter a text string in the Description field to find all defects whose description starts with those words. If you specify more than one search criterion, TrackRecord finds all items that meet *all* of the search criteria.

- 4 **Click Search.**

TrackRecord displays the **Search Results** window (Figure 4-36), which lists the items that meet your search criteria.

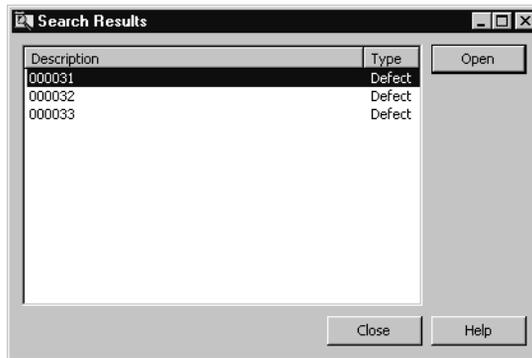


Figure 4-36 Search Results Window

5 Display the contents of a selected item.

Double-click an entry in the Search Results window or select an entry and click **Open**. TrackRecord displays the contents of the specified item.

6 When you are finished, click Close.

Chapter 5

Templates



This chapter explains TrackRecord templates and how to use them to accelerate entering project data.

About Templates

Templates simplify data entry in TrackRecord by automatically entering information in specified fields when you create a new item. For example, you can set up a template for new Defect items that enters your name, your computer configuration, and other information in the appropriate fields in the Defect form. You can create templates for any TrackRecord data type, such as Team Members, Tasks, or Defects.

Templates can be global, shared to selected work groups, or private. Private templates are applied after global and shared templates and can therefore be used to override global and shared templates.

Templates are “transparent;” when you layer one template on another, any field you leave blank can “see” the value applied by the templates behind it.

When Are Templates Applied?

TrackRecord applies templates in two circumstances: when you create new items directly and when you create new items from within an Outline Report.

When You Create Items Directly

When you create an item, TrackRecord automatically inserts values from templates for that type of item. Template values come from a nested set of sources, which are applied in the following order:

- 1 A global template for a parent of the new item

- 2 A template for a parent of the new item shared with the user's workgroup(s)
- 3 A user's personal template for a parent of the new item
- 4 A global template for that type of item
- 5 A template for that type of item shared with the user's workgroup(s)
- 6 A user's personal template for that type of item

When a value for a field is found at one level, TrackRecord ignores values specified at lower levels. For example, if a user's personal template for a Defect specifies "Colossus" as a project name and the global Defect template specifies "Millennium," TrackRecord uses Colossus.

When You Create Items From Within Outline Reports

Templates provide context-sensitive boilerplate for items created in an Outline Report section. When you use templates to create items from within an Outline Report, you can specify as many templates as you require to prefill the item's fields.

For example, Figure 5-1 illustrates an Outline Report that lists Defects by status (Open, Assigned, Pending Validation), with separate child headers for each Defect priority (1, 2, 3, 4) under each status header. You could add a Defect template to the Open status header to insert Open into the Status field when a new Defect is created from within that header. You could then add a Defect template to each child header under Open to insert the appropriate value in the Defect Priority field.

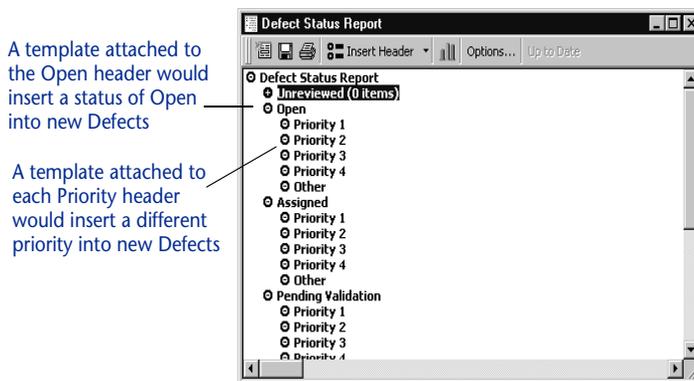


Figure 5-1 Adding Templates to Headers Simplifies Data Entry

When someone using this Outline Report wants to add a new Defect, he or she would right-click the appropriate child header and choose New Defect from the shortcut menu. TrackRecord would automatically fill the Status field with the information supplied by the parent header's template and the Priority field with the information from the child header's template.

Templates associated with child headers override templates associated with parent headers. For example, if the template linked to the parent header inserts "Millennium" in the Project field of a new Defect and the child header inserts "Vertigo," TrackRecord would insert "Vertigo" whenever a new Defect is created from within the child header.

Creating Templates

To Create a New Template

- 1 **From the File menu, choose the Manage/Templates option.**
The **Manage Templates** window (Figure 5-2) opens.

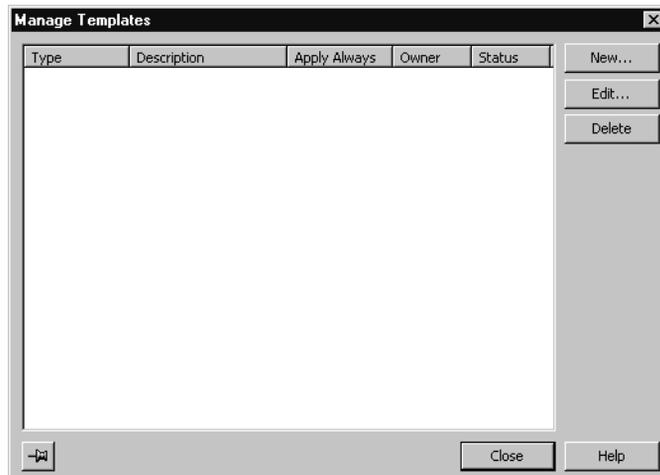
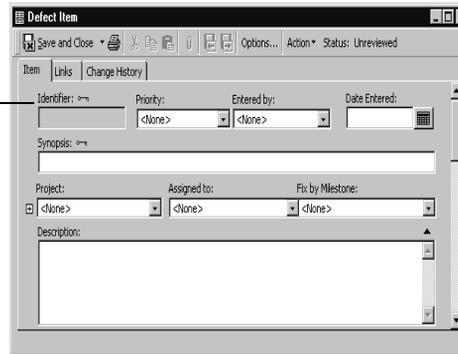


Figure 5-2 Manage Templates Window

- 2 **Click the New button.**
The **Choose a Type** window opens.
- 3 **Select the type of item for which you want to create a template and click the New button.**

An item form (Figure 5-3) for the selected type opens. The Identifier and Summary fields display a blue key icon beside the label to signify that these fields are not part of the template.

Blue key indicates field identifies or describes the template instead of establishing default information for a field in a new item



The screenshot shows a 'Defect Item' form with the following fields and controls:

- Identifier: A text input field with a blue key icon to its left.
- Priority: A dropdown menu with '<None>' selected.
- Entered by: A dropdown menu with '<None>' selected.
- Date Entered: A date picker control.
- Synopsis: A large text area.
- Project: A dropdown menu with '<None>' selected.
- Assigned to: A dropdown menu with '<None>' selected.
- Fix by Milestone: A dropdown menu with '<None>' selected.
- Description: A large text area.

At the top of the form, there are tabs for 'Item', 'Links', and 'Change History'. A toolbar contains buttons for 'Save and Close', 'Options...', and 'Action'. The status bar at the bottom indicates 'Status: Unreviewed'.

Figure 5-3 Template Form for Defect Item

- 4 Enter a description for this template in the Summary field.**
The Summary field is used to identify the template in the Manage Templates screen (that is, the Summary becomes the Template name).
- 5 Enter information in the fields you want the template to prefill.**
- 6 Click the Options button to open the Template Options window (Figure 5-4).**



The screenshot shows the 'Template Options' dialog box with the following fields and controls:

- Owner: A text input field containing 'edison'.
- Status: A dropdown menu with 'Not Shared' selected, and a 'Sharing...' button to its right.
- Apply this template to all new items: A checkbox that is currently unchecked.
- Buttons: 'OK', 'Cancel', and 'Help' buttons at the bottom.

Figure 5-4 Template Options Window

- 7 Optionally, specify whether you want to share this template with other users.**
Click the **Sharing** button to open the Sharing window (Figure 5-5). Click the **Share To** button and then click the checkbox next to each group with which you want to share the template.

Template is shared with team members in the Colossus and Millennium workgroups but not team members in the Vertigo workgroup.

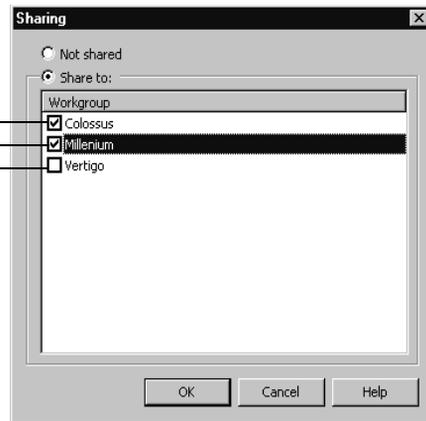


Figure 5-5 Sharing Window

- 8 If you have opened the Sharing window, click OK to return to the Template Options window.
- 9 If you want to use this template whenever you open a new item of this type, click the Apply This Template to New Items checkbox.
- 10 When you are finished, click OK.
You return to the template form.
- 11 If you are finished, click Save and Close.

Associating Templates with a Header

To Associate a Template with an Outline Report Header

- 1 Open the Outline Report to which you want to link a template.
- 2 Right-click the header to which you want to add a template and choose Set Templates.

The New Items tab on the Header Properties window (Figure 5-6) opens.

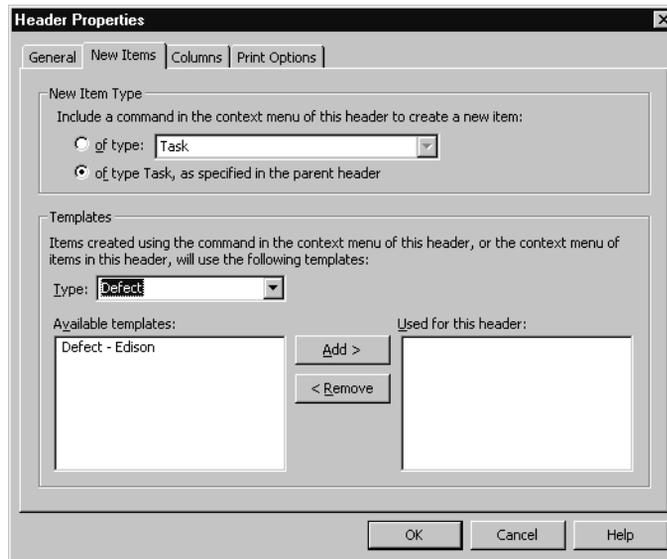


Figure 5-6 Header Properties Window – New Items Tab

- 3 **Use the New Item Type controls to specify the item types that you want to be able to add by right-clicking the header.**
 - ◇ Click the **Of type** option and choose the data type from the list on the right
 - ◇ To open the same types from this header as you can from the parent header, click the **Of type <type>, as specified in the parent header** button
- 4 **Identify the templates you want to use when creating new items from within the selected header.**

Use the **Type** list to select the type of item for which you want to set up templates. When TrackRecord displays the list of templates applicable to the type you select in the Available Templates list, select the one you want to use and click the **Add>** button.

You can use more than one template when you create items from within an Outline Report header. For example, you can set up one template to insert your name and a second template to insert a project name. This lets you reuse the first template when you set up templates for other headers in the Outline Report.

When you set up more than one template for an Outline Report header, TrackRecord uses information from the templates at the top of the list before using information from lower templates. For example, if the first template in the list specified a project name of “Colossus” and a lower template specified a project name of

“Millennium,” TrackRecord would use “Colossus” when creating a new item.

- 5 **When you are finished, click OK.**

Chapter 6

Customizing User Options



This chapter describes how to customize your startup options, favorite data types and Outline Reports, and how to log on as a different user.

Customizing TrackRecord

After you become familiar with TrackRecord, you will probably want to customize your user options to suit your style of work. TrackRecord user preferences fall into three categories:

- ◆ General – Controls how TrackRecord behaves at startup, how frequently TrackRecord polls its database, and how TrackRecord interacts with DevPartner, QACenter, and other Compuware tools.
- ◆ Favorite Types – Controls which data types appear in the File/New submenu, in the New list on the Toolbar, and in the Search Types list.
- ◆ Outline Reports – Controls the default type face and size for new Outline Reports and what headers in a report do when you double-click them.

Configuring General Options

The General tab on the TrackRecord Options window controls how TrackRecord behaves at startup, how frequently TrackRecord polls its database, and how TrackRecord interacts with DevPartner tools.

To Configure General Options

- 1 **From the Tools menu, choose TrackRecord Options.**

The TrackRecord Options window (Figure 6-1) opens.



Figure 6-1 TrackRecord Options Window – General Tab

2 On the General tab, specify what TrackRecord should do when you open a new session.

- ◇ To preserve your window layout from one session to the next, click **Open the windows that were open at the end of the last session**.
- ◇ To display your Home Page at the start of a session, click **Open the current Home Page**.

3 If you want to log in to TrackRecord without entering your user name and password, click Automatically log in at startup.

If you activate this option, whenever you start TrackRecord you will be automatically logged in and, since you will not be prompted for a database name, you will always access the same database. Anyone running TrackRecord on your computer will automatically be logged in to the last TrackRecord account and database that you used from TrackRecord or from one of the integrated tools.

4 Use the Check the Database for Changes slider to specify how frequently TrackRecord should poll its database for changes.

Polling the database frequently ensures that TrackRecord is working with current information but may slow other operations on your computer. If your network link is slow or if you are using Outline Reports that are not likely to change frequently, you may want to check the database less often.

5 If you are using the DevPartner tools and want TrackRecord to record information from DevPartner tools automatically, click Automatically Submit Sessions from DevPartner Tools.

6 Click OK.

Identifying Your Favorite Data Types

TrackRecord lets you set up a list of your favorite data types to simplify data entry. The list of your favorite types appears in three places in TrackRecord:

- ◆ When you choose **New** from the **File** menu, your list of favorite types appears at the top of the resulting submenu.
- ◆ When you use the **New Item** list on the Toolbar, your list of favorite types appears at the top of the resulting list.
- ◆ When you use the **Search Types** list, your list of favorite types lets you select a specific type to search.

Note Do not confuse Favorite Data Types with the Favorites menu, which lets you save shortcuts to frequently-used views and reports.

To Identify a Data Type as a Favorite

- 1 **From the Tools menu, choose TrackRecord Options.**
- 2 **When the TrackRecord Options window opens, click the Favorite Types tab.**

The Favorite Types panel (Figure 6-2) appears.

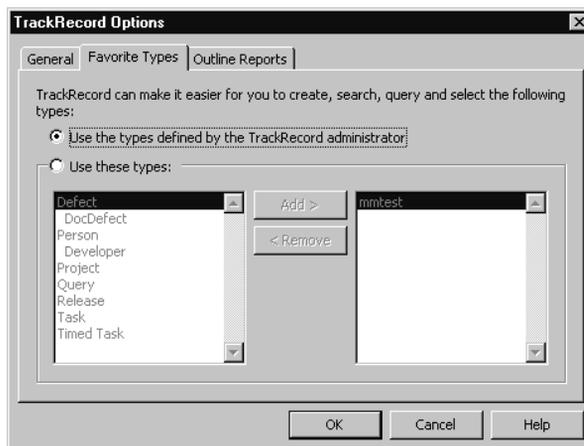


Figure 6-2 TrackRecord Options Window – Favorite Types Tab

- 3 **Identify the data types you want to appear in your Favorite Types list.**
 - ◇ To use the default list of favorite data types, click **Use the types defined by the TrackRecord administrator.**

- ◇ To customize your favorite types list, click **Use these types**, select each data type you want in your list from the list on the left, and click **Add >**. To remove an entry from your list of favorite types, select its name in the list on the right and click **<Remove**.
- 4 **Specify the order for favorite types by dragging entries in the list on the right up and down.**
The first entry in the Favorite Types list appears as the label for **New Item** on the Toolbar.
 - 5 **When you are finished, click OK.**

Setting Up Outline Reports

The Outline Reports tab on the TrackRecord Options window controls the default type face and size for new Outline Reports and what headers in a report do when you double-click them. You can override these default settings for specific reports or for specific headers within a report.

- 1 **From the Tools menu, choose TrackRecord Options.**

When the TrackRecord Options window opens, click the **Outline Reports** tab. The Outline Reports panel (Figure 6-3) appears.

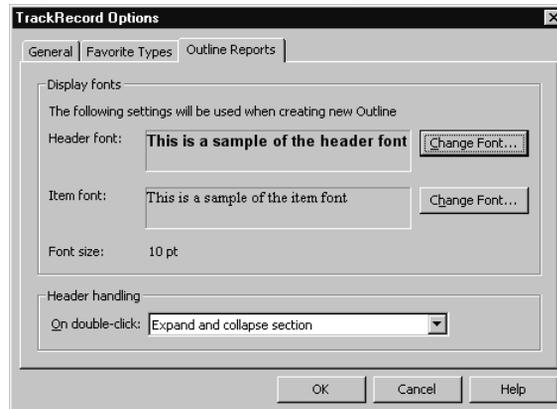


Figure 6-3 TrackRecord Options Window – Outline Reports Tab

- 2 **To change the font used to display report headers:**
 - ◇ Click the **Change Font** button to the right of the header font sample.

- ◇ When the Font window opens, select the type face, type style, and size you want TrackRecord headers to use.

Note You can use different type faces and styles for headers and items, but the header font and the item font must use the same type size. Resetting the size for one automatically changes the size for the other.

- 3 To change the font used to display items in an Outline Report:**
 - ◇ Click the **Change Font** button to the right of the item font sample.
 - ◇ When the Font window opens, select the type face, type style, and size you want TrackRecord items to use.
- 4 Specify what you want TrackRecord to do when you double-click an Outline Report header.**
 - ◇ Select **Expand and collapse section** if you want TrackRecord to display and hide the entries under a header when you double-click it.
 - ◇ Select **Open Header Properties** if you want TrackRecord to display the property form for a header when you double-click it.
 - ◇ Select **Open Query** if you want TrackRecord to display the query associated with a header when you double-click it.
- 5 Click OK.**

Logging In as Someone Else

Under some circumstances, you may want to log in to TrackRecord as a different user. For example, you might want to log in as a TrackRecord administrator if you want to export data to another application.

To Log In as Someone Else

- 1 From the Tools menu, choose Change Login.**
The Change Login window (Figure 6-4) opens.

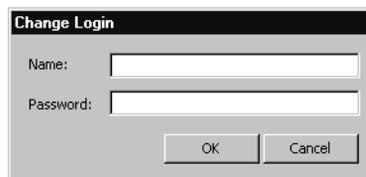


Figure 6-4 Change Login Window

- 2 **Enter the user name under which you want to log in to TrackRecord in the Name field.**

Note You cannot change databases when you log in as someone else. To change databases, you must exit and restart TrackRecord.

- 3 **Enter the password associated with that user name in the Password field.**

- 4 **Click OK.**

The user name displayed in the lower right corner of the TrackRecord main window identifies the user name of the current user.

Chapter 7

TrackRecord and Other Applications



This chapter describes how to integrate TrackRecord with text editors, graphics programs, Microsoft Project, and version control systems.

Using TrackRecord with Text Editors

TrackRecord makes it easy to copy data from Outline Reports into other applications, including text editors, spreadsheet programs, and electronic mail applications. When you paste data from an Outline Report into an application that supports Rich Text Format (RTF), such as Microsoft Word or Microsoft Outlook, the report's columns, fonts, and format are preserved in the new document.

Note Information copied from TrackRecord to other applications is embedded statically. It will not change when the data in TrackRecord changes.

To Embed an Outline Report in a Text Document

- 1 **Select the headers or items you want to copy and choose Copy from the Edit menu to copy the Outline Report data to the Windows Clipboard.**
- 2 **Switch to the document window for your target application and choose Paste from the Edit menu.**

If you copied an Outline Report header, the header you selected, its child headers, and any sibling headers below the one you copied appear in the target document window. If you selected an Outline Report item, that item is pasted to the target document.

- 3 **Optionally, update the style information for pasted data to make it conform to your document standards.**

TrackRecord information pasted into other applications uses two styles: *TR Header* and *TR Item*. After you have pasted information from TrackRecord into a text editor, you can quickly adjust the typefaces, type sizes, and other properties of the embedded information by changing settings for these styles.

Using TrackRecord with Graphics Applications

You can copy graphs generated by TrackRecord into graphics applications, such as Microsoft Paint. After you paste a TrackRecord graph, you can modify the graph using the application's tools.

Additionally, you can paste graphs generated by TrackRecord into text editors, spreadsheet programs, and electronic mail applications that can accept bitmap images. For example, you can paste a bar graph from TrackRecord into Microsoft Word, Microsoft Excel, or Microsoft Outlook.

Note Graphs copied from TrackRecord to other applications will not change when the data in TrackRecord changes.

To Paste a TrackRecord Graph into Another Application

- 1 **Generate the graph from the appropriate Outline Report.**
Use the TrackRecord graph controls to format the graph. Refer to "Using the Graph View" on page 51 for information on formatting TrackRecord graphs.
- 2 **Choose Copy from the Edit menu to copy the TrackRecord graph to the Windows Clipboard.**
- 3 **Switch to the window for your target application and choose Paste from the Edit menu.**
- 4 **Optionally, if the target application lets you edit bitmap images, change the colors or layout of the TrackRecord graph.**

Using TrackRecord with Microsoft Project

You can integrate TrackRecord with Microsoft Project to let managers and development team members exchange up-to-date project status information quickly and easily. Integrating TrackRecord and Project lets you link development schedules to items in the TrackRecord database. Periodically synchronizing Project information and TrackRecord information lets you update information about project dates and milestones, status and priorities of tasks, and the availability of project team members.

Managers can track actual schedules against baseline schedules easily; each time a team member marks a task as complete in TrackRecord, its status and dates are adjusted in the project plan in Project. Similarly, a project team member can use TrackRecord to obtain a list of current tasks (which may include tasks defined in TrackRecord or Project). When a team member completes an assignment, TrackRecord automatically updates the project database and passes the new assignment information to Project.

Overview of Integration

The following provides an overview of the process of integrating TrackRecord and Microsoft Project information. Each bullet is described fully in the following sections. These tasks are generally performed by a project manager.

- ◆ Install the Microsoft Project TrackRecord integration macros.
- ◆ Perform initial project setup, which connects a Microsoft Project-based schedule with a TrackRecord project. The TrackRecord project team members are pulled into Microsoft Project and made available as resources in Microsoft Project.
- ◆ Map Project priorities to TrackRecord priorities.
- ◆ Synchronize the two applications. TrackRecord creates a new Microsoft Project Task item for each task listed in Microsoft Project, and synchronizes task status.
- ◆ Synchronize the applications periodically, whenever you want to update project information. Refer to “What Happens During Synchronization?” on page 95 for an explanation of how fields are synchronized.

Microsoft Project Macros

To integrate TrackRecord and Microsoft Project information, begin by installing the TrackRecord macros in Microsoft Project.

Installing the TrackRecord macros adds a TrackRecord menu to the Microsoft Project toolbar.

Note You can enable two TrackRecord toolbar buttons, one for synchronization and the other for opening a task item, by right-clicking on the Microsoft Project toolbar and clicking on TrackRecord.

To Install the Microsoft Project Macros

- 1 **Double-click the TrackRecord Integration.mpt file in the TrackRecord folder.**

By default, the TrackRecord Integration.mpt file is installed in the C:\Program Files\Compuware\TrackRecord folder. Microsoft Project opens.

- 2 **If the Microsoft Project Macro warning dialog (Figure 7-1) opens, click Enable Macros.**

This dialog will not display if you have cleared the **Always ask before opening projects with macros** checkbox.

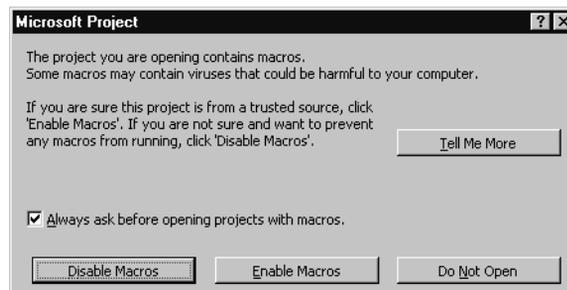


Figure 7-1 Microsoft Project Window

- 3 **When the TrackRecord Integration Installed window appears, click OK.**

If the Office Assistant is open, TrackRecord displays status and error messages through the Office Assistant.

TrackRecord Menu

There are three options on the TrackRecord menu added to Microsoft Project by the integration macros.

- ◆ **Synchronize Project Now** – Choosing **Synchronize Project Now** causes TrackRecord and Microsoft Project to exchange project information. Refer to Table 7-1 for information on what types of information are transferred during synchronization.
- ◆ **Open Current Item in TrackRecord** – Selecting an item, such as a task, milestone, or resource, in Microsoft Project and choosing **Open Current Item in TrackRecord** causes TrackRecord to run and open the selected item.
- ◆ **Options** – Choosing **Options** opens the TrackRecord Options window, which lets you change the user ID, database, project, or priority map.

Initial Project Setup

Before TrackRecord and Microsoft Project can exchange project information, you must log in to TrackRecord from within Project and identify the database and TrackRecord project associated with the current schedule.

To Perform Initial Project Setup

- 1 **Open the Microsoft Project project file you want to use.**
- 2 **From the TrackRecord menu, choose Options.**

The TrackRecord Options window (Figure 7-2) opens.

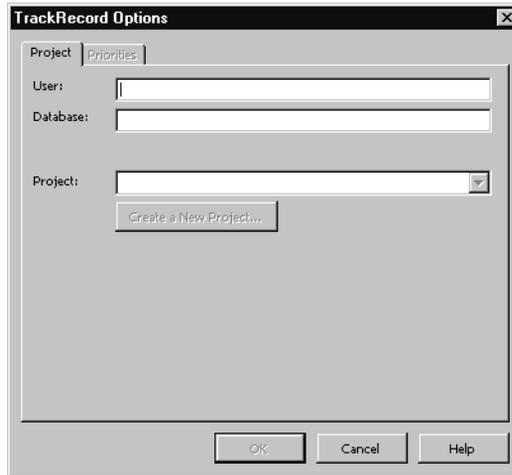


Figure 7-2 TrackRecord Options Window

- 3 Enter your TrackRecord login name in the User field.
 - 4 Identify the database you want TrackRecord to use in the Database field.
 - 5 To specify a project, do one of the following:
 - ◇ To use an existing project, click the Project list.
 - ◇ To create a new TrackRecord project, click Create New Project.
- The TrackRecord Login window (Figure 7-3) opens.



Figure 7-3 TrackRecord Login Window

- 6 Enter your login password in the Password field and click OK. TrackRecord passwords are case-sensitive. You return to the TrackRecord Options window after you log in to TrackRecord.
- 7 If you are using an existing TrackRecord project, select the project you want from the Project list.
- 8 If you are creating a new project, select the resources (Team Members) you want to assign to the new TrackRecord project by clicking the checkbox beside each Team Member you want to add as a Microsoft Project resource (Figure 7-4). Click OK.

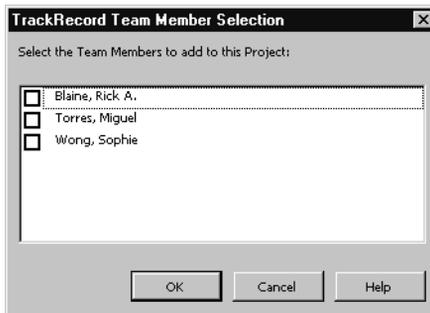


Figure 7-4 TrackRecord Team Member Selection Window

9 When you are finished, click OK.

Mapping TrackRecord Priorities to Project Priorities

The priority assigned to a task indicates the task's importance within a project. When schedules grow tight, tasks with low priorities are typically deferred or eliminated. TrackRecord and Project have different default priority settings. Before you can use resource leveling in Project, you must map the priority levels in TrackRecord to the corresponding priority levels in Project.

To Map Priorities

- 1 Run Microsoft Project.
- 2 Open a project you have previously set up for TrackRecord/Project integration.
- 3 Open the **Priorities** tab.

Choose **Options** from the **TrackRecord** menu in Project. When the TrackRecord Options window opens, click the **Priorities** tab. The **Priorities** tab (Figure 7-5) opens.

Note If your TrackRecord administrator has modified the default priorities in TrackRecord, your **Priorities** tab may differ from the one shown in Figure 7-5.

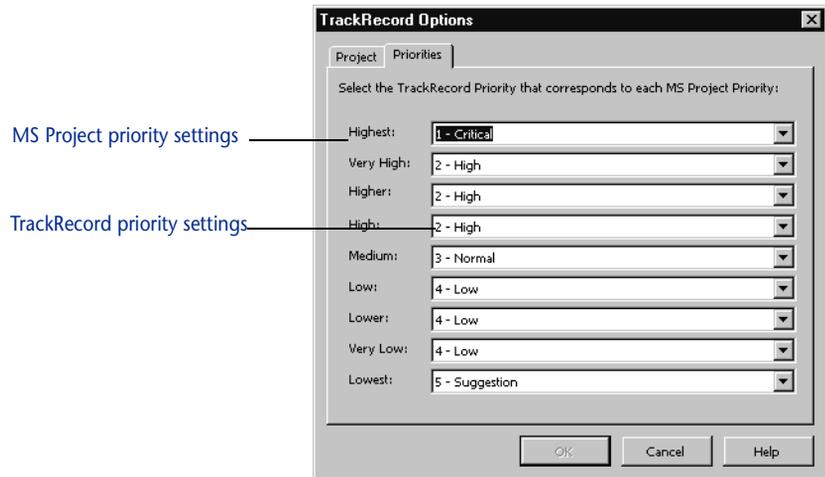


Figure 7-5 TrackRecord Options window – Priorities Tab

4 Assign a TrackRecord priority to each priority in Project.

For example, map the TrackRecord priority “1 – Critical” to the Project priority “Highest” by selecting 1 – Critical in the Highest list.

5 When you are finished, click OK.

Synchronizing a Project in TrackRecord and Microsoft Project

Refer to “What Happens During Synchronization?” on page 95 for a detailed list of which fields are overwritten during synchronization.

Note Tasks are not synchronized if no team member has been assigned to the task. If you have not assigned a TrackRecord resource name (team member) for each task in Microsoft Project, do so now.

To Synchronize a Project

1 Open the Microsoft Project file for the project you want to synchronize.

2 From the TrackRecord menu in Project, choose **Synchronize Project Now**.

The TrackRecord Synchronization window (Figure 7-6) indicates the progress as TrackRecord and Project exchange information.

Note If the Office Assistant is running, TrackRecord uses it to display synchronization summary information.



Figure 7-6 TrackRecord Synchronization Window

3 When the TrackRecord Synchronization Complete window appears, click OK.

What Happens During Synchronization?

When you choose **Synchronize Project Now**, TrackRecord and Microsoft Project exchange information about Team Members, Projects, Tasks, and Milestones in the project.

TrackRecord adds a Microsoft Project section to the bottom of the Task item form. Checking the **Auto Update Summary** field in the Microsoft Project section of a Task item form will cause the Summary field to be read-only in TrackRecord, allowing it to be changed through Project only. Unchecking this option allows you to change the Summary in TrackRecord, but your changes will not be reflected in Microsoft Project.

Table 7-1 identifies what fields in TrackRecord are overwritten by information from Microsoft Project and what fields in Microsoft Project are overwritten by information from TrackRecord.

TrackRecord Item	TrackRecord Field	Synch Direction	MS Project Item	Project Field	Comment
Team Member	Name	-->	Resource	Name	Project members are defined in TrackRecord and sent to MS Project.
	Initials (Abbreviation)	-->		Initials	
	Email Address	-->		Email	
Project	Name	<-- (New items only)	Project	Name	Project Name and Description can be defined in Project and sent to TrackRecord on first synchronization.
	Description	<-- (New items only)		Notes	
	Team Members	--> (new or changed)		Resources	
Task	Summary	<--	Task	Name	Tasks are created in MS Project. Completion status is captured in TrackRecord.
	Description	<--		Notes	
	Assigned To	<--		Resource	
	Priority	<--		Priority (mapped)	
	Date Due	<--		Finish (baseline)	
	Date Completed	-->		Finish (Actual)	
	Completed	-->		% Complete	
	Identifier (autogenerated)	-->		WBS field	
Milestone	Name	<--	Task	Name	Milestones and Planned dates are defined in MS Project. Actual dates are captured in TrackRecord.
	Description	<--		Notes	
	Project	<--		Associated Project	
	Projected End Date	<--		Finish (baseline)	
	Start Date	-->		Project start date	
	Actual End Date	-->		Finish (actual)	

Table 7-1 TrackRecord-Microsoft Project Synchronization

Using TrackRecord with Version Control Systems

Version control system (VCS) applications help you manage projects by controlling and tracking source and other project files. You can integrate TrackRecord with VCS applications, such as Microsoft Visual SourceSafe, Merant PVCS, and MKS Source Integrity. Integrating TrackRecord with a version control system lets you check files in and out from within TrackRecord, open read-only versions of source files, and synchronize TrackRecord releases with VCS labels.

When you integrate TrackRecord with a VCS application, each tool tracks parallel sets of changes to your software projects: TrackRecord tracks change requests and their resolution, such as defects that have been fixed or features that have been added, and your VCS application tracks changes to files. By integrating the tools, you can identify which source file changes were associated with specific change requests.

For example, a developer fixing a defect checks out source files, makes changes to them, and checks them in again. If a problem with the defect fix is subsequently identified, you can use the VCS files to identify and roll back the files that were modified, making it easier to resolve the problem.

For each file, release, and project in a VCS application, TrackRecord maintains a corresponding item in the database. This item references the source file. In most situations, TrackRecord creates items automatically when you check files in or out. However, you must update information for milestones, projects, and for file items that already exist in TrackRecord.

To Associate a TrackRecord Project with a VCS Project

- 1 **Open a new or existing project in TrackRecord.**
 - ◇ To open a new project, from the **File** menu choose **New/Other**. From the **Choose a Type** window, select **Project** and click **New**.
 - ◇ To open an existing project, from the **File** menu choose **Open/Other**. From the **Item Browser**, select **Project** and double-click the appropriate project name in the **Items** list.
- 2 **When the Project form opens, click the button beside the VCS Project Name field (Figure 7-7).**

Click the VCS button to log in to your version control system.

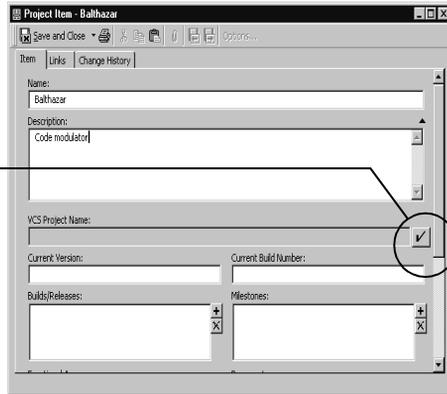


Figure 7-7 Browse Project Button on Project Form

- 3 Enter your user name and password in the Username and Password fields of the version control system login window (Figure 7-8).

Note Depending on the version control system you are using, your dialog may differ from those shown.

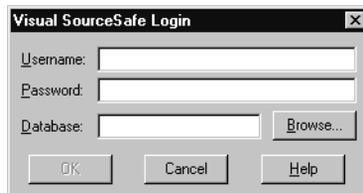


Figure 7-8 Version Control System Login Window

- 4 Use the Database field to specify the version control database you want to use.

If you know the name of the database you want to use, enter it in the **Database** field. If you are not sure what name to enter, click the **Browse** button, select the appropriate database from the Open Database window (Figure 7-9), and click **Open**.

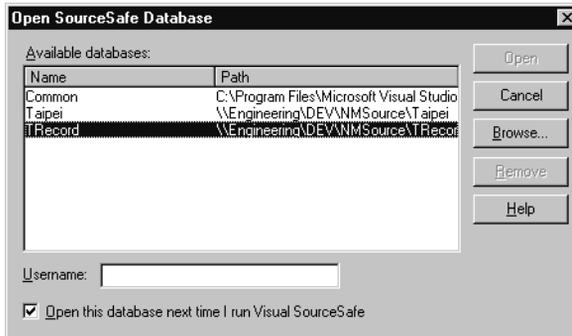


Figure 7-9 Open Database Window

5 Click OK.

A dialog is displayed in which you specify the VCS project to use, or enter the name of a new project.

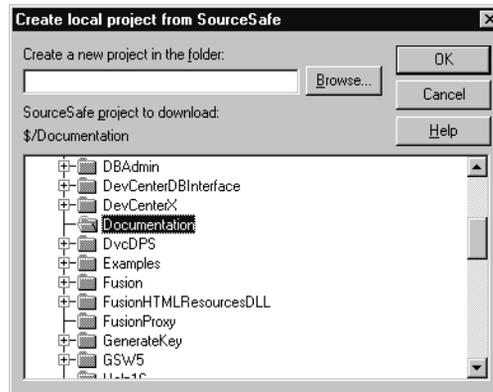


Figure 7-10 Choose Local Project From SourceSafe Window

6 Specify where you want to store the new project.

Enter the name of the folder where you want to store the VCS project or click the **Browse** button (Figure 7-11) to select the folder you want to use.

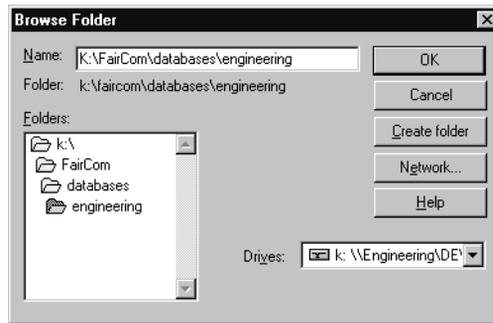


Figure 7-11 Browse Folder Window

7 Click OK.

The name of the project you select appears in the **VCS Project Name** field in the Project form.

If you do not already have **Source File** and **Data File** items that you want to connect with your VCS files, TrackRecord will automatically create these items the first time you check the file in or out from within TrackRecord. When the items are created, you will see a status message, and have the opportunity to enter a description of the new file.

Checking Files In and Out

You typically check source files in and out of a VCS system in the course of working on a specific TrackRecord Defect or Feature. When you check files in and out of a VCS system from within TrackRecord, TrackRecord automatically records the file name and current VCS version number in the item's change history.

To Check a File In or Out from TrackRecord

- 1 **Open the appropriate Defect or Feature item.**
- 2 **Verify that the project identified in the Project field is associated with a project in your version control system application.**
- 3 **Choose Check Out or Check In from the Edit|Source Control menu.**

You will be presented with a list of files from your VCS project, which can be checked in or out as appropriate.

TrackRecord copies any descriptive comment you enter when you check files in to the Resolution Description field of the Defect.

Synchronizing TrackRecord and VCS Projects

Labels are user-defined identifiers used by VCS applications to represent a specific version of a project or file. When you label a project, all the files in that project inherit the label. For example, if you assign a label of “Beta1” to a project, you can retrieve the appropriate version of files within that project by using the project label.

Note If TrackRecord cannot find the VCS project, verify that the Milestone item is contained in a Project item that contains a valid VCS Project field. To add the release to the Project, open the Project item, scroll to the **Milestones** field, and click **Add Item**. After the Milestone item links to the Project item, you should be able to display the list of your VCS labels.

Connecting your TrackRecord releases with your VCS releases works the same way as connections to projects and file names, with one exception: you can create a new VCS label from within TrackRecord, at the same time you create a TrackRecord release.

To Associate a TrackRecord Release with a VCS Label

- 1 **Open the Milestone item in TrackRecord.**
- 2 **Click the button beside the VCS Release/Label field.**
TrackRecord displays a list of your VCS labels.
- 3 **To use an existing label, select the appropriate entry in the list. To create a new label from within TrackRecord, click the New button.**

To create a new VCS label from within TrackRecord, press the checkbox next to the **VCS Release** field, and click **New** in the window that appears. TrackRecord will use the name of the release as the default value for the label; you can change this value when prompted for the release name. When TrackRecord creates the release, it applies the label to the entire project associated with the Release item.

- 4 **Click OK.**

Troubleshooting

If you are using TrackRecord with one of the supported version control systems, use the following troubleshooting and configuration information if you encounter difficulties.

All Supported VCS Systems

Make sure the TrackRecord user has a VCS user ID and password in the User Administration dialog. To bring up the User Administration dialog, log in as an Administrator and choose Administrator/User Administration. Double-click on the user name.

Microsoft Visual SourceSafe 4.0 and 5.0

- ◆ SourceSafe must be installed before TrackRecord. If SourceSafe is installed after TrackRecord, run the SETUP.EXE for TrackRecord to enable VCS support. (None of your views or data files will be affected by rerunning setup.) The following lines are automatically added to the Windows registry under HKEY_CURRENT_USER/ SOFTWARE/COMPUWARE/TRACKRECORD:

```
sourceControl=SourceSafe scPath=<path_to_SSEX.P.EXE>  
scCommandPath=<path_to_SS.EXE>
```

If these lines are not present and SourceSafe was installed prior to the TrackRecord installation, rerun SourceSafe setup and ensure that the “Enable SourceSafe Integration” checkbox is checked. Then run TrackRecord’s setup again.

- ◆ If you are still having trouble with the integration, verify TrackRecord is correctly reading your SourceSafe information:
 - ◇ Check the Windows registry for the previously mentioned scPath line. This will guide you to your main SourceSafe directory.

For example: You may have

```
scPath=D:\VSS5\WIN95\SSEX.P.EXE. In this case, D:\VSS5 is  
your main SourceSafe directory.
```

From here, open a file called SRCSAFE.INI and look for a Users_Txt variable. Note that in some cases, SRCSAFE.INI might only contain an “include” statement that may point to another SRCSAFE.INI. In this case, follow along to the new location until you find the Users_Txt variable. Open the specified file referenced in your Users_Txt variable (usually this is USERS.TXT). The file, in turn, specifies a SS.INI file for each SourceSafe user. Look in the referenced SS.INI file for the user that is having problems and make sure that there is a section for each project, with the default source directory specified as follows (for example, for a project called \$/test):

```
[$/test] Dir (PC) = C:\SOURCE
```

- ◆ If these sections do not exist (which will be the case if you created the projects with the 32 bit version and the 16 bit version was not installed at the time), create them manually.
- ◆ If you are getting “Checkin failed” messages, the most common cause is that some other user checked out the file. Use SourceSafe to see who checked out the file, then have that user check the file back in before continuing.

PVCS

Please note that only the 32 bit version of PVCS (either PVCS for Windows 95 or PVCS for Windows NT), version 5.2.10 or later, is supported. In addition, the PVCS interface for Visual C++ must also be installed. You can get these interface files from the Merant Web page (www.merant.com). Currently, the URL for the interface files is <http://www.merant.com/pvcs/download/eval/evlic-pvcs-vmccci.asp>.

Install the interface before you install TrackRecord. Otherwise, you must rerun the SETUP.EXE for TrackRecord to enable the VCS support. (None of your views or data files will be affected by rerunning setup.) The following lines are automatically added to the Windows registry under HKEY_CURRENT_USER/SOFTWARE/COMPUWARE/TRACKRECORD:

```
sourceControl=PVCS scPath=<path_to_PVCSVMNT.EXE>  
scCommandPath=<PVCS_path>
```

Also, TrackRecord uses the command line PVCS tools to obtain some information. Be sure that those tools are set up to log in to PVCS as the correct user, because TrackRecord cannot pass the user information directly into PVCS in these cases. For more information on changing the user, see the PVCS documentation.

Finally, make sure the PVCS directory (WIN95, WINNT, or WIN32 depending on your version of PVCS) is in your path. The PVCS installation puts it there, and the command line tools will not work properly without it.

MKS Source Integrity

Source Integrity support requires Source Integrity version 7.2 or above, and the Visual C++ hooks for VC++ 4.0. Currently, you can get the Source Integrity hooks from the MKS Web page at <http://www.mks.com/support/packaged/si/download.htm> Be sure to download the Visual C++ 4.0 hooks for your operating system (16 or 32 bit).

Install the hooks before you install TrackRecord. If you install the hooks after you install TrackRecord, you must run the SETUP.EXE for TrackRecord to enable VCS support. (None of your views or data files will be affected by rerunning setup.) When setup is complete, the following lines should be present in your Windows registry under HKEY_CURRENT_USER/SOFTWARE/COMPUWARE/TRACKRECORD:

```
sourceControl=MKS scPath=<path_to_MKSSI32.EXE> scCom-  
mandPath=<path_to_PJ.EXE>
```

Also, make sure that the MKS MKSSI directory is in your path. Source Integrity normally sets the path during installation.

Source Integrity requires each TrackRecord user to select the Source Integrity project file (their sandbox file) the first time that project is used for an operation in TrackRecord. TrackRecord automatically prompts for the sandbox name when it is needed, and stores it for future use.

Chapter 8

Using TrackRecord WebServer



This chapter describes how to use the WebServer browser interface to the TrackRecord database to add, maintain, and display data items and reports.

About WebServer

WebServer allows you to access the TrackRecord database through a Web browser, such as Microsoft Internet Explorer or Netscape Navigator/Communicator.

When you log in to TrackRecord through the WebServer, TrackRecord accesses your user information in the database just as though you had logged in through the TrackRecord Windows client. If you have set preferences such as a Home Page or favorite reports through your Windows client, those items will be maintained in the WebServer client as well. The WebServer also recognizes your membership in Groups and Share Groups and allows you to run the same reports you run through the Windows client.

While using the WebServer, you can create new items, view and modify existing items, and view existing Outline Reports. You cannot create new queries, Outline Reports, or other reports.

Logging In to TrackRecord WebServer

To Log In To TrackRecord WebServer

- 1 **Run a Web browser such as Microsoft Internet Explorer or Netscape Navigator.**

- 2 **Enter** `http://servername/trackrecord/welcome.html` **in the Destination field of your Web browser and press Enter.**
Your TrackRecord administrator can tell you what value to enter for *servername*.

The TrackRecord WebServer Login page (Figure 8-1) opens.



Figure 8-1 TrackRecord WebServer Login Page

- 3 **Enter your user name, password, and the name of your TrackRecord database.**
- 4 **Click OK.**

The TrackRecord WebServer main window (Figure 8-2) opens.

The WebServer Window

The WebServer window (Figure 8-2) is your Web interface to the TrackRecord database. You use this window to create and modify items and view reports.

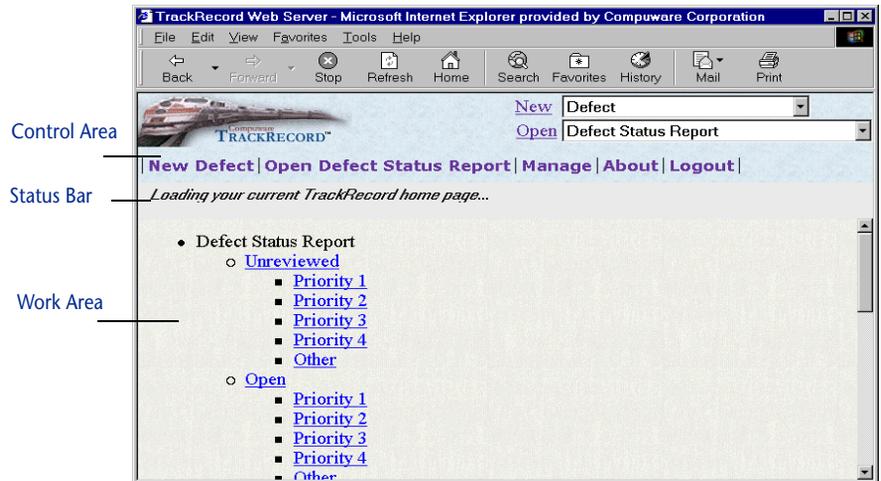


Figure 8-2 TrackRecord WebServer Main Window

The WebServer window consists of three areas:

- ◆ In the Control area, you select the task you want to perform.
 - The two drop-down menus at the top right of this area allow you to choose to create a new item or to open a report.
 - The links are shortcuts to your most common tasks. The links on your screen may differ from the illustration, since they depend on your user settings.
 - ◇ The first link (New Defect by default) allows you to create a new item of your favorite type. If your favorite type is set to Task, for example, this link will be New Task.
 - ◇ The second link is a shortcut to your Home Page report, if you or your administrator has specified a Home Page in TrackRecord. If a Home Page has not been specified, this link is not displayed.
 - ◇ For administrators, the third link is the Manage option. This link is not displayed if you are not logged on as an Administrator.
 - ◇ The Logout link allows you to log out of the WebServer.
 - ◇ The About link displays the TrackRecord WebServer version number, the name of your database and your username.
 - ◇ The Help link displays information about using the WebServer.
- ◆ The Status Bar contains either of the following:

- ◇ A status message, such as an error or a message indicating communication with the server, or a series of dots that are displayed to indicate that the WebServer is performing an action
- ◇ Information about the item or report you are currently viewing, including the title of the page and buttons that operate on the current item or report

The contents of this area vary. For example, when you are working on a Defect, the status bar contains a Cancel button and a Save button.

- ◆ The Work Area displays the selected item or report. When you first access the WebServer, your TrackRecord Home Page is displayed in the work area by default. If you have not set a Home Page, the work area will be empty.

Creating a New Item

You can use the WebServer interface to create new TrackRecord items. The WebServer uses the same item form as the TrackRecord Windows client. You enter information just as you would in the Windows client with three exceptions:

- ◆ You must manually enter dates in date fields; the Calendar icon is not functional through the WebServer.
- ◆ Always enter date information using the same date format as that being used by the computer running the WebServer. For example, if the WebServer machine is configured to use dates in the format dd/mm/yyyy, you should enter a date 28/10/1999, not 10/28/1999. Note that alphabetic characters are not supported in dates.

To update an existing TrackRecord item from the WebServer interface, you must open an Outline Report that includes that item and then click the item's link.

To Create a New TrackRecord Item

- 1 **Open the TrackRecord WebServer main window as described in “Logging In to TrackRecord WebServer” on page 105.**

- 2 From the New drop-down menu, select the type of item you want to create.

Alternatively, click the first link (New Defect by default) to create an item of that type.

The Work Area displays a form allowing you to enter the new item. The Status Bar displays buttons appropriate for this item, such as Cancel and Save.

The screenshot shows a web browser window titled "TrackRecord Web Server - Microsoft Internet Explorer provided by Compuware Corporation". The browser's address bar shows "New Defect" and "Open Defect Status Report". The page content includes a navigation bar with "New Defect", "Open Defect Status Report", "Manage", "About", and "Logout". Below this is a "Defect" form with the following fields and values:

Field	Value	Action
Identifier	000057	Open
Priority	None	Open
Entered by	None	Open
Date Entered	1/21/00	
Assigned to	None	Open
Fix by Milestone	None	Open
Project	None	Open
Status	Unreviewed	Open
Severity	None	Open

The form also includes a "Summary" text area, a "Description" text area, and "Cancel" and "Save" buttons at the bottom right.

Figure 8-3 New Defect Form in a WebServer Window

- 3 Complete the fields in the form.

Required fields are identified with red field labels.

- 4 When you are finished, click the Save or Cancel button in the Status Bar.

Note that the Save button is also located at the bottom of the form for your convenience.

Opening an Outline Report

Any Outline Report that is available to you through the Windows client is also available through the WebServer. When accessing an Outline Report through the WebServer, note the following differences from Outline Reports in the Windows client.

- ◆ You can open and close all headers at once with the **Expand All** and **Collapse All** buttons in the Status Bar. Note that **Expand All**

may take a some time to complete, depending on the contents of the headers.

- ◆ If a report contains more than ten items in a header, when the header is expanded the first ten items are displayed. Use the **Next** button to display the next set of ten items.
- ◆ When you open a header, the report scrolls to position the open header at the top of the Work Area.
- ◆ If you click on a drop-down list that causes TrackRecord to run rules, TrackRecord must access the database to run those rules. The item will be partially saved when the database is accessed. Once the item has been partially saved, clicking the **Cancel** button will display a message that a version of the item has been saved and you cannot cancel your changes. You must manually restore the values that you want saved.

To Open an Outline Report

- 1 **Open the TrackRecord WebServer main window as described in “Logging In to TrackRecord WebServer” on page 105.**
- 2 **Select the Outline Report you want to display from the Open list.**

If you have identified a Home Page report, you can open your Home Page by clicking on the link in the Control Area.

The report is displayed.

- 3 **Click a report header to show the items within that header.**
You can click the **Expand All** button to display all items in the report, or click **Collapse All** to hide items. You cannot collapse individual headers.
- 4 **To update an item in the TrackRecord database, click its reference in the Outline Report.**
- 5 **To close the report, open another item or report.**

WebServer Window Notes

Note the following when using the WebServer window.

- ◆ TrackRecord spawns a separate browser window in which to run the Web application. This window should be used exclusively for running TrackRecord. If you use this window to access another URL or if you close the window, your TrackRecord session will be terminated.
- ◆ The context menu (right-click menu) **Open Item in New Window** option is not supported. Although a window will be opened, it will not be functional. To display two items simultaneously, open another browser window, enter the URL for the welcome.html page, and log in to TrackRecord from this window.

Logging Out

To log out of TrackRecord, click the **Logout** link in the Control Area of the WebServer window.

You are automatically logged off if you close the browser or access a different URL. You should use this browser window exclusively for TrackRecord tasks. Use another browser window to access other URLs.

metrics, to help developers and testers resolve problems quickly and consistently.

DevPartner Studio Enterprise Edition tightly integrates its debugging and tuning tools with TrackRecord. The DevPartner Studio tools can send data directly to a TrackRecord project database. The project database is accessible to developers from within the DevPartner Studio tool such as BoundsChecker or TrueTime, and from within Visual Studio. This integration provides users with the information they need from within the tools they use.

DevPartner Studio Interaction with TrackRecord

DevPartner Studio Enterprise Edition derives much of its power from its underlying *ActiveLink* connection to TrackRecord. ActiveLink automates the submission of debugging/tuning tool information to a database. This accumulated information provides:

- ◆ Platform configuration data, including hardware resources and version information about installed software components
- ◆ Coverage data gathered from all developers using TrueCoverage
- ◆ Code volatility information
- ◆ Build matching to registered builds
- ◆ Task lists for each developer
- ◆ Detailed error messages from the NuMega error detection tools

You can choose to submit DevPartner Studio session information from DevPartner Studio tools automatically. For more information, refer to “Configuring General Options” on page 81.

In addition to session submissions, developers can use the DevPartner Studio menu or special toolbar buttons to submit any Defect and Task items from within the debugging/tuning tool. Defect items describe a problem with an application component, while Task items describe a particular activity requiring action.

Logging In to TrackRecord from DevPartner Studio Tools

When starting a DevPartner Studio tool, you are prompted to log in to a TrackRecord database. Logging in at this time does not start TrackRecord, but allows access to tasks and defects within the database.

If you are already logged in to TrackRecord, you can change your login identity from within any of the DevPartner tools.

To Log In to a TrackRecord Database

- 1 **From the DevPartner menu, click Change Login.**
The TrackRecord Login window (Figure 9-1) opens.

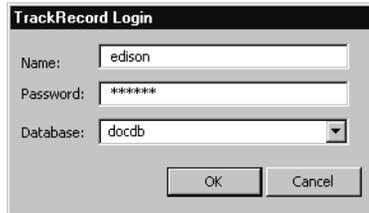


Figure 9-1 TrackRecord Login Window

- 2 **Enter your login name in the Name field.**
- 3 **Enter your login password in the Password field.**
Passwords are case-sensitive.
- 4 **Identify the database you want to use in the Database field.**
The names of the databases you have used recently appear in the **Databases** list. You can select the database you want to use from this list or enter a different database name.
- 5 **Click OK.**

DevPartner Studio Integration with Visual Studio

Developers can access DevPartner Studio components, including TrackRecord, from Microsoft Visual Studio.

Each DevPartner debug component treats Microsoft Visual Studio integration differently. The Visual Basic and Visual C++ integration for TrueTime and TrueCoverage simply launches those tools to allow you to check your test application. Submitting defects to TrackRecord from these tools works just as if you had launched TrueTime or TrueCoverage from the Windows Start menu. SmartCheck integration in Visual Basic works similarly, by launching a copy of the SmartCheck tool.

TrackRecord installation adds BoundsChecker to the integrated development environment (IDE) of Visual C++ to provide integrated debugging. The TrackRecord **Defect** and **Task** buttons appear on the BoundsChecker toolbar within Visual C. Menu selections for submitting items to TrackRecord appear on the **TrackRecord** sub-menu on the **BoundsChecker** menu. The TrackRecord **Defect** and **Task** buttons become active only when you select a BoundsChecker event while viewing the BoundsChecker **Results** window.

JCheck resides inside the Microsoft Visual J++ IDE. As with BoundsChecker, the **Submit** buttons appear on the JCheck toolbar, and the **TrackRecord** sub-menu appears within the **JCheck** menu, which itself appears as a sub-menu on the Visual J++ **View** menu.

CodeReview detects problems in Visual Basic source code. Defect submission involves selecting an item in the CodeReview window and using the TrackRecord **Defect** and **Task** buttons, or the right mouse button Context menu. The **Submit TrackRecord Defect** menu selection on the Visual Basic IDE's **DevPartner Studio** menu allows the user to key in a TrackRecord defect or task. CodeReview automatically submits defects or tasks detected by CodeReview.

DevPartner Menu

The DevPartner menu provides the following commands:

- ◆ Submit DevPartner Defect
- ◆ Submit DevPartner Task
- ◆ Submit Session Information
- ◆ List Tasks
- ◆ Change Login

Menu commands are disabled when no data exchange operation is possible.

Task Lists

The task list provided by DevPartner Studio Enterprise Edition lets you view and close tasks. Clicking the **Open Task List** option on the TrackRecord menu will create a task list based on one of the default queries shipped with DevPartner Studio Enterprise Edition. If you are running Visual Studio, the query MyTasks is run. If you are running one of the DevPartner tools, the query specific to that tool is run, such as MyTrueTimeTasks.

DevPartner Toolbar Buttons



The DevPartner toolbar buttons let you enter DevPartner Defects or Tasks. Clicking on these buttons opens a defect or task form, allowing you to key information into the TrackRecord database.

Automated Session Submission

Each time a developer runs a debugging/tuning tool, DevPartner Studio prompts for the submission of that session to a project database. A session submission captures information about the error detection session that was run. For example, TrueCoverage sessions capture information about the software build and the machine configuration the session was run on. This information can help teams assess their code testing status and code readiness.

Build Registration

TrackRecord logs DevPartner Studio developer tool session information against a specific project build. Most development teams will appoint a single person to manage the build creation process. Refer to “Registering One or More Application Software Builds” on page 175 for information about how to register builds.

Instrumenting Your Code

DevPartner Studio’s CodeReview, BoundsChecker, and SmartCheck tools can diagnose errors without instrumenting an application’s source code. Other DevPartner Studio tools may require code instrumentation.

Refer to the documentation for each DevPartner Studio developer tool for information about how to instrument code for use with that tool.

Defect/Task Submissions

Submitting a TrackRecord Defect or TrackRecord Task starts with selecting an item from a DevPartner Studio tool window that displays debug results. This window will have a different title depending on the tool: for BoundsChecker, SmartCheck, and JCheck, a three-pane window displays debug results; for JCheck, the left hand pane provides a Results tab and a Transcript tab from which you can select information; for TrueTime and TrueCoverage, locate results in the two-pane window that provides a tab for the right hand pane. Developers can select lines from either the Function list or Source tab.

When you submit defects (using the TrackRecord Defect type shipped in the sample database), TrackRecord pre-fills the field tagged for Description with the debug results you select from a results window. For BoundsChecker and SmartCheck, memory leaks and errors provide stack data that is copied into the Defect item’s Stack field. For JCheck, debug results include either error events or advisories. JCheck debug

results also pre-fill the Stack field with call stack data. Defect submission from TrueTime or TrueCoverage does not pre-fill the Stack field.

When you test an application under development with these debug tools, TrackRecord collects several categories of information about the session, including:

- ◆ The software and hardware configuration of the submitter's platform, including detailed information about all DLL and OCX files referenced by the process being tested
- ◆ The project and build, if the build has been registered

Defects entered from a DevPartner Studio developer tool can pertain to a particular tool-reported error, or they can be unrelated to the current activity within the tool. Use the **Submit DevPartner Defect** command to enter a defect that does not relate to a specific tool-related error.

To Enter a Defect from a Developer Tool

1 Choose **Submit DevPartner Defect** from the **Visual Studio DevPartner Studio** menu.

TrackRecord opens a blank Defect form.

2 Enter information into the defect report.

3 Click **Save and Close**.

Use toolbar buttons within developer tools to enter defects that relate to a specific error or problem. For example, CodeReview analyzes Visual Basic source code and reports a variety of logic, portability, performance, standards compliance, and usability problems. From within CodeReview, developers can enter Defect or Task items that contain problems identified by CodeReview. This submission capability lets a developer step through Visual Basic code and enter relevant defects and tasks without having to switch applications.

To Submit CodeReview Problems as Defects or Tasks

1 Open a project in Visual Basic.

2 Choose **Start CodeReview** from the **DevPartner** menu.

3 If prompted, save the current CodeReview session.

4 Click **Start Review** in the **CodeReview** dialog.

See the CodeReview on-line help for information about choosing settings or modules.

5 Highlight an error in the **CodeReview Results** window.

- 6 **Click the DevPartner toolbar icon to enter a defect or task.**
This action submits a defect or task for the highlighted problem. Note that the problem description is copied into the Description field of the task or defect item.
- 7 **Click Save and Close to save the task or defect.**
- 8 **Repeat for each applicable CodeReview problem.**

Debug Session Output

TrackRecord accepts the submission of developer tool set session runs from TrueTime, TrueCoverage, BoundsChecker, and SmartCheck. TrueTime and TrueCoverage can submit runs for Visual C++ or Visual Basic applications, but not Java, unless the Java application contains modules written in Visual C++ or Visual Basic.

At the end of a DevPartner tool debug session, TrackRecord's ActiveLink initiates the run submission process. First, TrackRecord examines the EXE, DLL, and OCX components your application loaded to determine if any of them match a registered build. If TrackRecord discovers that the components match more than one build, a **Multiple Matching Builds Found** dialog lets you choose the build to log the run against. If no builds match, the **Build Not Found** dialog lets you register the build, if you have access rights to register new builds.

Once TrackRecord locates a registered build, it displays a **Submit Session** dialog that lists tasks in an upper pane and has an edit box at the bottom. The edit box lets a developer log comments about the run, such as "Testing the new customer dialog, logged various bugs." The list box at the top shows a list of tasks. These tasks use types and queries provided in the sample database shipped with TrackRecord, and the queries refer specifically to a field labeled DevPartner, which will contain the name of a DevPartner developer tool. The queries, named My SmartCheck Tasks, My BoundsChecker Tasks, My TrueCoverage Tasks, or My TrueTime Tasks, return a list of tasks preceded by a check box. Developers can click in the check box to indicate completion of the task. TrackRecord will then mark the tasks as Completed.

The sample database queries return tasks that:

- ◆ Refer to a specific DevPartner developer tool, or to none of the tools
- ◆ Contain an assignment to the current user
- ◆ Contain no assignment
- ◆ Are not yet completed

TrackRecord removes any tasks that have the **Repeat for Each Build** option checked if those tasks were already completed for that particular build.

The sample database queries provide examples that developers can customize. Alternatively, a developer might create a general-purpose query with the name My TrackRecord tasks. If TrackRecord does not find the product-specific or the My TrackRecord Tasks query, it uses a built-in query to retrieve the information that the sample queries would otherwise locate.

In addition to the task list that appears on the submit session dialog, TrackRecord provides a **Task List** command on the DevPartner menu. This list mirrors that in the submit session dialog, and allows a developer to review a task before marking it completed.

When a developer checks off a task and clicks **Submit**, TrackRecord updates the tasks as follows:

- ◆ Tasks that need not be repeated for each build are marked **Completed**. The **Completed in Build** field adds the build to the list. The **Completed in Run** field adds the run to the list. The project gets set to the project from the build. The **Date Completed** field gets set to the current date.
- ◆ Upon run submission, TrackRecord creates a session object in the database. For TrueTime and TrueCoverage, TrackRecord attaches the session file.

Submitting Debug Sessions

BoundsChecker and SmartCheck detect a variety of runtime errors, including memory leaks, stack and heap memory problems, and API call correctness.

To Submit a Debug Session

- 1 **Run your application under BoundsChecker or SmartCheck.**
At the conclusion of the run, a window prompts you to submit the session.
- 2 **Enter a brief description of your activity during this session in the Description field.**
- 3 **Click any completed tasks that appear in the upper pane.**
- 4 **Click Submit.**

Submitting Tuning Sessions

TrueTime and TrueCoverage let developers and testers profile the performance of an application.

To Submit a Tuning Session

- 1 Run your application under TrueTime or TrueCoverage.**

At the conclusion of the run, a window prompts you to submit the session.

- 2 Enter a brief description of your activity during this session in the Description field.**
- 3 Click any completed tasks that appear in the upper pane.**
- 4 Click Submit.**

Detecting Duplicate Defects and Tasks

For leaks and errors in BoundsChecker and SmartCheck, results in JCheck, or any event in CodeReview, TrackRecord checks for possible duplicates when receiving a new submission. TrackRecord uses both the Defect description and the location of the error in the source code to develop a list of possible duplicate TrackRecord Defect items.

The **Potential Duplicate Items Found** dialog's **Open** button launches TrackRecord to view any potential duplicate entry. A user can inspect the contents of a potential duplicate and decide if the new submission duplicates existing information. After determination that the new submission duplicates an existing item, a user can modify the existing defect to add any new information that might be relevant. If the user decides that the submission is a new, unique defect, he can click the **Submit Anyway** button.

To Examine a Duplicate Submission

- 1 Select the DevPartner Defect submission you want to examine and click Open.**

The DevPartner Defect item opens, letting you evaluate its contents.

- 2 Click Submit Anyway to submit the pending item as a new defect.**

Resolving Multiple Matching Build Conflicts

When DevPartner Studio determines that a run or defect/task submission matches more than one build, it asks the developer to choose the build against which to submit the session.

To Resolve Multiple Matching Build Conflicts

- 1 **Highlight a build listed in the Multiple Matching Builds Found window.**
- 2 **Click Select.**

Viewing Task Lists

TrackRecord can provide developers with a list of tasks. These task lists result from the issuance of queries that search for the tasks assigned to the current user.

To View a Task List

- 1 **Choose Open Task List from the DevPartner Studio Enterprise menu.**
- 2 **Click any tasks you have completed.**
You cannot open tasks from this window.
- 3 **When you are finished, click OK.**

TrackRecord and DevPartner Studio Coverage Information

TrueCoverage users can merge session files that accumulate within their private work space. These merged sessions indicate how much testing that developer's code received over time.

With TrackRecord and TrueCoverage, coverage sessions can be merged and filtered across users and environments. Merging coverage sessions from *all* the developers working on an application lets a manager or test coordinator determine how much of an application's total code base has been exercised by test programs.

Refer to the TrueCoverage documentation and online help for information about how to use TrueCoverage.

Merging coverage sessions entails two steps: creating a coverage merge set and merging the sessions. A developer typically chooses what coverage sessions should be merged and what sessions should be

excluded. Criteria for identifying sessions to merge can include the following.

- ◆ Application component
- ◆ Build
- ◆ Date
- ◆ Memory
- ◆ Milestone
- ◆ Operating System
- ◆ Person
- ◆ Project

You can match one of these selections to a specific value, to any value, or to any value except one you specify.

To Create the Criteria for a Merge Coverage Operation

- 1 **Select a target from the left-most list box.**
- 2 **Select a match criteria from the right-most list box.**
- 3 **If you selected “is equal to” in Step 2, select a value from the bottom list box.**

For example, if you selected “Operating System is equal to” in the two top lists, you would select a value, such as “Windows 98,” from the lower list.

- 4 **Click Add.**
- 5 **Click Next to view the sessions that met your criteria.**

To Merge Coverage Sessions

- 1 **Click the check box next to a session to toggle it on or off.**

When checked, that session will be merged with the other files selected. If unchecked, that session will not be merged with the other selected files.

- 2 **Click Merge.**

The TrueCoverage main window opens and displays a bar graph and statistics about the amount of lines and functions exercised by your unit tests.

Platform Configuration Information

When an application runs under one of the DevPartner tools, TrackRecord records hardware and software configuration information about the computer on which the tested application executes. The

hardware configuration information is stored in an item called Hardware Configuration and is associated to the defect or session via the Defect's **Configuration** field. Software configuration information is stored as a list of Executable File objects in a Defect or TrackRecord Session item's **Loaded Components** field.

The Configuration type contains fields for information such as the type of processor, amount of memory, screen resolution, and the operating system.

A **Loaded Components** field in an item tracks the details of DLL, OCX, and the EXE files used by the application being debugged. These details include the link date and time and version resource strings. Many software vendors, including Compuware and Microsoft, use version resource strings to provide unique version numbers of components.

Platform configuration information, including software component version numbers, can aid the process of finding errors caused by component incompatibility.

Viewing Captured Platform Configuration Information

Suppose a tester reports a defect, but a developer cannot duplicate the defect. One cause for discrepancies of this nature is mismatched versions of essential software components, such as dynamic link libraries and ActiveX custom controls. TrackRecord's Advanced Search capability gives team members one tool for uncovering such discrepancies.

To View Sessions on Hosts That Use a Specific Component

- 1 Select Advanced Search from the TrackRecord Tools menu.**
The **Choose a Type** window opens.
- 2 Select DevPartner Session and click Search.**
The DevPartner Session Item search window opens.
- 3 Double-click the Loaded Components field.**
The Item Browser displays a list of Application Component items.
- 4 Select the component from the Application Component list and click OK.**
- 5 Click Search on the DevPartner Session Item Search Dialog toolbar.**
The Search Results window opens.
- 6 Select any listed item and click Open.**

- 7 Click Close to close the Search Results window.
- 8 Click Cancel to close the DevPartner Session Item search window.

Remote Error Capture

Even the best crafted and well tested application will contain some errors. When sent out for external tests, a development team can track these errors using DevPartner Studio's FailSafe combined with TrackRecord AutoAlert.

Development groups can deploy applications to external sites for testing, and can instrument those applications with FailSafe. When errors occur, FailSafe instrumentation creates an e-mail message containing detailed error information, and sends it to TrackRecord's AutoAlert server. AutoAlert will then create a defect on behalf of FailSafe, containing the relevant information.

Refer to "Configuring AutoAlert" on page 192 for information about setting up an AutoAlert server.

There are two uses of FailSafe with DevPartner Studio:

- ◆ DevPartner Studio Standard Edition e-mail to a named user
- ◆ DevPartner Studio Enterprise Edition e-mail to a TrackRecord AutoAlert server that routes the data to a TrackRecord database

If you add TrackRecord to an existing DevPartner Studio installation, you must copy three files named `FailSafe.*` from `\Program Files\Compuware\FailSafe\VB5\TrackRecord` to the `\ProgramFiles\Compuware\FailSafe\VB5` directory to enable integration with AutoAlert. If you installed DevPartner Studio Enterprise Edition, but you want to disable AutoAlert integration, you must copy the same three files from the `\ProgramFiles\Compuware\FailSafe\VB5\DPS` directory to `\ProgramFiles\Compuware\FailSafe\VB5`.

DevPartner Studio provides two ways to set up FailSafe for use with DevPartner Studio:

- ◆ Using the ClientINI Editor on the FailSafe Tools menu (recommended)
- ◆ Manually editing the `FSCLIENT.INI` shipped with your application

To Prepare an Application for Deployment with FailSafe

Note If you need to enable/disable TrackRecord AutoAlert integration, copy the needed files *before* you instrument your application.

1 Instrument your application with FailSafe.

Use the FailSafe Autoprocessor Wizard or run FailSafe from within Visual Basic.

2 Click ClientINI Editor on the FailSafe Tools menu, open your project's Client.INI file, and click the TrackRecord tab.

3 Enter the following information into the fields on the TrackRecord tab.

- ◇ The e-mail address of the machine at your location on which AutoAlert is running
- ◇ The name of the TrackRecord database for which the data is bound
- ◇ The name of the TrackRecord project to which the information will be directed
- ◇ The name of a TrackRecord Team Member to whom the information will be assigned
- ◇ The option to attach the FailSafe log (set this option to 1 to attach the log)

Note You can manually edit the [TrackRecord] section of the FSCLIENT.INI file you deploy with your application instead of using the above procedure.

Chapter 10

Using TrackRecord with QACenter and Fault Manager



This chapter describes how to use TrackRecord in conjunction with QADirector and Fault Manager to track defects in your test application.

Using TrackRecord and QACenter

The Compuware QACenter product family offers sophisticated automated testing tools for mainframe, client/server and Web development projects. QACenter tools include the following:

- ◆ QARun provides record/playback capabilities for testing client/server applications.
- ◆ QALoad performs load testing for client/server applications.
- ◆ QADirector helps manage the testing process across the enterprise.
- ◆ QAHiperstation provides record/playback capabilities for testing VTAM-based applications.
- ◆ QABatch provides regression testing for MVS batch applications.
- ◆ QAPlayback provides advanced capabilities for testing CICS-based applications.
- ◆ QAHiperstation+ tests mainframe applications from the desktop.
- ◆ QAPlayback+ provides a Windows-based interface for testing mainframe-based applications from the desktop.

TrackRecord can capture and track defects identified with these tools and manage the resolution process. QACenter's QADirector provides a menu for submitting defects directly to TrackRecord and for viewing submitted defects from within QADirector. For example, if a failed QADirector test reveals a defect in the application, you can submit the defect to TrackRecord from QADirector, and QADirector will automatically enter relevant information about the failed test in the new Defect item.

When TrackRecord and QACenter run in the same development environment, they can share a single project database. Defects entered

through QADirector can become part of TrackRecord Milestone Status reports or Outline Reports.

QACenter integration requires no configuration.

You can submit a Defect for a particular test procedure from within QADirector. Alternatively, you can submit a Defect for a test class.

To Submit a Defect

- 1 **Run QADirector.**
- 2 **Open the Summary pane and double-click the summary of the job that failed.**
The Suite pane displays the **Result** tab for the selected job.
- 3 **Double-click the test class or test procedure that failed.**
- 4 **From the QACenter menu, choose TrackRecord/Submit Defect.**
A new Defect window opens in TrackRecord. Information about the failed test appears in the appropriate fields of the new Defect.
If necessary, enter additional information about the failed test in the Defect window.
- 5 **When you are finished, click Save and Close to save the new Defect.**

Using TrackRecord with Fault Manager

Fault Manager makes available extensive information about mainframe application defects. The following sections describe how to access and manipulate the Fault Manager data collected by TrackRecord.

To Visit a Single Data Item

- 1 **From the File menu, click Open, then select Outline Report.**
- 2 **Double-click an Outline Report entitled Fault Defect.**
- 3 **Double-click a header to expand it.**
- 4 **Double-click outline levels as necessary to present the base data item, which is identified by an ID number assigned in the order it was added to the TrackRecord database.**
- 5 **Double-click the base data item to access the Fault Defect Item window.**
The window shows all fields and values associated with the selected fault defect.
- 6 **Exit by clicking the x button at upper right.**

- 7 Collapse outline levels by double clicking them.

To Create a Graph

- 1 Select an outline level suitable for comprehensive graphical review.
- 2 Click the Graph icon on the Fault Defect window menu bar.
- 3 Use any available formatting tools on the menu bar of the graph window.
Save and Print functions are also available on the menu bar.

Copy and Paste

You can rearrange levels and items in an Outline Report by selecting and then dragging with the mouse or by using the Edit menu **Cut** and **Paste**. A paste will be inserted below the selected outline level. You may want to combine data from different Outline Reports.

To Copy and Paste

- 1 Select the outline level containing the data you want to use
- 2 Choose Cut or Copy from the Edit menu.
- 3 Open a second Outline Report.
- 4 Select a level and then choose Paste
The data will be inserted below the selected level.
- 5 For a graphical view that incorporates the data of both reports, select the next higher outline level and click the Graph icon.

Save As

In addition to the Save command on the window menu bar, you have a variety of saving and sharing options offered by the TrackRecord Save As function.

To Use Save As

- 1 From the File menu, select the Save As option.
Save As may be applied to both Outline Reports and graphs.

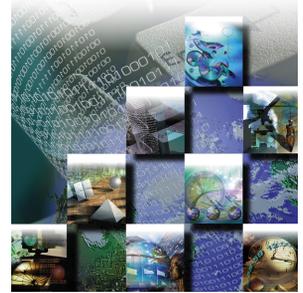
Part 2: Administering TrackRecord

- ◆ Chapter 11: TrackRecord Configuration
- ◆ Chapter 12: TrackRecord Database Administration



Chapter 11

TrackRecord Configuration



Upon completion of TrackRecord client and server installation, an administrator must perform various configuration tasks before TrackRecord can be used. This chapter explains these configuration tasks. (Note that licensing issues are addressed in the *TrackRecord Installation Guide*, not in this chapter.)

The tasks described in this chapter require administrative privileges to the database.

This chapter assumes familiarity with basic TrackRecord concepts. For an overview of TrackRecord, refer to Chapter 1, “Introducing TrackRecord.”

The Administrator

To guarantee the security of project data, TrackRecord restricts certain activities to users with administrative privileges. Users with no administrative privileges cannot perform restricted tasks.

Administrators can:

- ◆ See and use the Administrator menu in TrackRecord
- ◆ Globally share queries, Outline Reports, Milestone Status views, and templates
- ◆ Delete the queries, Outline Reports, Milestone Status views, and templates owned by other users
- ◆ Change the owner of a template
- ◆ View all the types defined for a given database with the Item Browser
- ◆ Set the Home Page and default favorite reports and types for users
- ◆ Define workflow for defects, tasks, and other items

This chapter describes tasks that require administrative privileges.

To Perform Administration Tasks

- 1 Click the Windows Start menu, point to Programs, then to Compuware, and click TrackRecord.
- 2 Enter “admin” in both the Name and Password fields (lowercase only for Password).
- 3 Enter a database name.

The default database name was specified during TrackRecord installation.

If you are using multiple databases, you will perform Administrative tasks for each database independently.

- 4 Click OK.

See “Creating Team Members” on page 138 for information about limiting administrative access to TrackRecord databases.

Note You should change the administrator password immediately, as described in “Changing Your Login Password” on page 16.

Configuration Administration Tasks

The key components of a working TrackRecord project include:

- ◆ *Groups* that specify access privileges for sets of people
- ◆ *Share groups* that gather people into logical units for sharing information
- ◆ *People* who are assigned to groups and share groups
- ◆ *Projects* that are a set of milestones, tasks and other items for different development efforts
- ◆ *Default user preferences*, queries, and reports
- ◆ *Workflow* which is the set of actions users can perform on an item and the set of states that result from those actions
- ◆ *Types* which are the information structures, and corresponding forms, that provide your database infrastructure

A TrackRecord configuration administrator is usually responsible for creating these components. The administrator creates these components by completing the series of configuration tasks described in this chapter.

Although the order in which you perform these tasks is flexible, groups should be created before other tasks are performed. If you are new to TrackRecord, perform administrative tasks in the order presented in this chapter.

Group Administration

Membership in a group determines *access rights* (privileges) to the information in a TrackRecord database. These rights include:

- ◆ Read/write access
- ◆ Read and Add (cannot edit an existing item, but can create new ones)
- ◆ Read-only access
- ◆ Hide (a field-level privilege setting that allows making a field invisible on a group-by-group basis)

For example, members of a group of software developers might require read/write access to certain database entries, while a group of customer support engineers needs read-only access to those entries.

The first step in controlling access to TrackRecord information involves creating groups and establishing the access rights (privileges) for those groups.

Note that Groups are not the same as Departments, although the two might often coincide. Team Members belonging to several departments can belong to one Group if their database access rights will always be equal.

Group Access Mechanisms

Groups provide the mechanism to control access rights to TrackRecord information. When you create a group, you assign access rights to that group. Later, when you create team members, you can assign them to one or more groups. This group assignment controls what kind of access that user has to TrackRecord information. Not assigning a user to any group grants that user full read and write privileges across the database. Assigning a user to more than one group gives the user the least restrictive access of those groups.

You can set access permissions at three levels: *global*, the default permissions for a group; *type*, which can be different for each type; and *field*, which restricts access to, or even hides, individual fields within types.

For example, suppose you create a group named Doc Reviewer and assign it Read/Write privileges. Users assigned to this group will be able to modify and add Defect reports as they review documentation. You could modify those rights at the type level. You might edit the Defect type to change the access privileges to Read/Add for the Doc Reviewer group. This change would allow users in this group to add new defect items, but prevent editing any existing Defect items. You can further

restrict the rights by field. For example, you could ‘hide’ defect details on the defect form from the Doc Reviewer group.

You set type and field permissions as part of type administration (see “Type Administration” on page 156).

Users can belong to up to fifteen groups and their database access privileges become those of their least restrictive group. Read-only access rights derived from membership in one group will be superseded by membership in another group that has read and write access.

When you add a new group with read/write access to TrackRecord, everyone in that group will have full access to all information types unless you edit the types to reference the new group.

Creating Groups

To Add or Modify a Group

- 1 **Click the Administrator menu, and then select Group Administration.**

The Group Administration dialog opens.

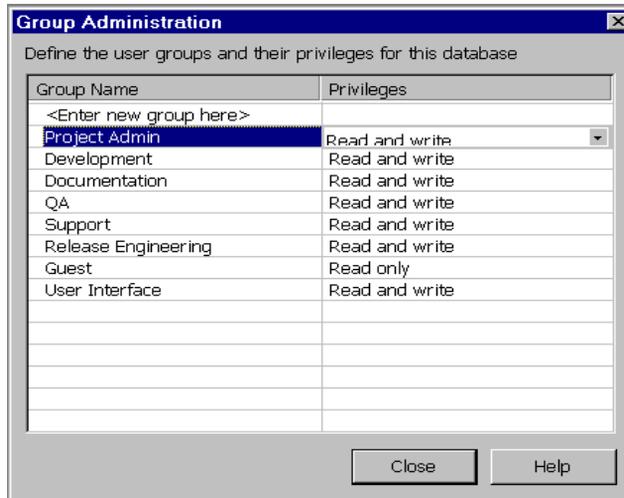


Figure 11-1 Group Administration

- 2 **Select <Enter a new group here> and enter a new group name, or select an existing group.**
- 3 **Click the drop-down arrow in the Privileges column to view the list of choices:**
 - ◇ Read/Write – Provides full access
 - ◇ Read Only – Allows viewing, but not making changes

- ◇ Read and Add – Prevents editing of existing items, but allows adding new items

This setting determines the group's access rights to the TrackRecord database. You can set more restrictive access to specific types and fields within the database (such as hiding a field) by editing types, as described later in this chapter.

- 4 Click on the desired privilege type and press Enter.
- 5 Click Close.

To delete a group, highlight the name and press the **Delete** key on the keyboard. Use caution when deleting groups, since eliminating a group eliminates access to the database for anyone assigned only to that group.

Share Groups

Share Groups define logical groups with whom members can share queries, Outline Reports, Milestone Status views, and templates. Users can belong to multiple share groups.

Use of share groups is optional, but useful. For example, assume you have set up a Developers group, QA group, and Writers group. A project team may consist of a few members from each of these groups. If you want the project team to share a common set of Outline Reports, you would create a share group containing the project's team members, create the reports and allow the share group access to these reports.

TrackRecord provides three types of sharing:

- ◆ Not shared (private)
- ◆ Shared with all members of a share group
- ◆ Globally shared (administrators only)

When TrackRecord users share a query, template, or view of database data, they can share it with members of any of the share groups to which they belong. An administrator can share those same items with all groups. An administrator can also change the owner of a report or query, and can share anyone's report and queries with everyone.

TrackRecord's sample database contains a Share Group type. Administrators can use this type to create share groups items to mirror the software development process within their company or department.

To Create a Share Group

- 1 From the Toolbar, click the Item Browser button.**
Alternatively, from the **Tools** menu select the **Item Browser** option.
- 2 Select Share Group and click the New button on the toolbar.**
The Share Group item form opens.
- 3 Enter a name for the share group.**
- 4 Enter a description of the share group.**
- 5 Click the + button at the top right of the Members field to add Team Members to the group.**
If Team Members have not yet been created, you can add members to Share Groups when you create the member items, as described in the next section, or you can create Team Members now by clicking the **New** button.
- 6 Click on the members to be added to this share group.**
- 7 Click OK to dismiss the Item Browser dialog.**
- 8 Click Save and Close.**

To Share Information with Share Groups

After you have created Queries, Outline Reports and Milestone Status views, you can determine what users can access them by assigning them a sharing property. You can share them to all users or to specific share groups.

- 1 Open a Query, Outline Report, or Milestone Status view.**
- 2 Click Options, and then click Sharing.**
- 3 Click the option Share To and then check the boxes next to the names of the share groups with whom you want the information shared.**
- 4 Click OK.**

Creating Team Members

For each project listed in the TrackRecord database, a set of Team Members should be assigned to work on the project.

The sample database supplies a Team Member type that you can use to create your Team Members. This sample type contains links to projects and groups. This association allows some fields to pre-fill automatically with information appropriate for that Team Member.

Before creating Team Member items, ensure that the Team Member type is suitable to your needs. If not, edit the Team Member type, as described in “Type Administration” on page 156, before creating Team Members.

A Team Member does not automatically have access to TrackRecord. Team Members are simply data items within a TrackRecord database. Access to TrackRecord is controlled by usernames, which are created by the TrackRecord Administrator. The Administrator can associate a username (i.e., LoginID) with a Team Member (data item).

Non-administrator users can create Team Member items, but they cannot create usernames, and therefore cannot authorize access to the TrackRecord database for these Team Members. If non-administrator users create Team Member items, an administrator must edit these Team Member items to add a login username to authorize access to the TrackRecord database.

Creating Team Members without access to the database can be useful in certain circumstances. For example, you might want to associate an external vendor to a project by creating a Team Member item for that vendor so you could then assign defects to them, but they would not have authority to access your database.

To Create Team Members

- 1 Click the Toolbar’s Item Browser button, select Team Member and click the New button in the Item Browser’s toolbar.

A Team Member item opens, as shown in Figure 11-2.

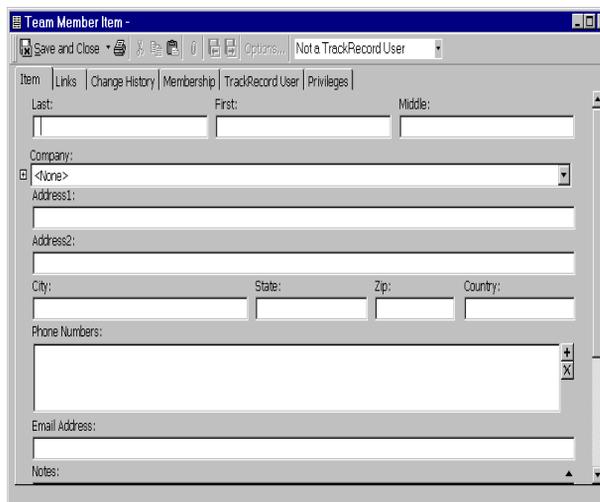
The image shows a screenshot of a software window titled "Team Member Item -". The window has a menu bar with "Save and Close", "Options...", and "Not a TrackRecord User". Below the menu bar is a tabbed interface with tabs for "Item", "Links", "Change History", "Membership", "TrackRecord User", and "Privileges". The "Item" tab is active, showing a form with the following fields: "Last:" (text input), "First:" (text input), "Middle:" (text input), "Company:" (dropdown menu with "<None>" selected), "Address1:" (text input), "Address2:" (text input), "City:" (text input), "State:" (text input), "Zip:" (text input), "Country:" (text input), "Phone Numbers:" (a list box with a "+" button to add and an "X" button to delete), "Email Address:" (text input), and "Notes:" (text input). The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

Figure 11-2 A Team Member Item

2 On the Item tab, enter information to identify an individual team member.

During the creation of the first Team Member item, fill in Company information to create a Company item. When creating subsequent Team Members you can add company information by selecting it.

3 The Links and Change History tabs are informational; do not edit these tabs.

After the Team Member is created, the Links tab will show all items to which this Team Member is linked. The Change History tab will display an audit trail, allowing you to track changes that have been made to this Team Member item.

4 On the Membership tab, click on the Projects and Share Groups with which this Team Member is associated.

If you have not created Projects or Share Groups, you can create them now with the New button, or you can add Team Members when you create Projects and Share Groups at a later time. Creating Projects is described in “Project Configuration” on page 141. Creating Share Groups is described in “Share Groups” on page 137. Some users, such as managers who do not work on a specific project but who occasionally log in to TrackRecord to view information, may not require a project membership.

5 Click on the TrackRecord User tab.

The fields of this tab are inactive because, by default, Team Member items are not authorized TrackRecord users. Use the TrackRecord User control in the toolbar to select this user's status.

Note If your Team Member Item window is not wide enough, the TrackRecord User control will not be visible on your toolbar. Resize the window until the control is displayed.

The user status options are:

- ◇ Normal – Active with no administrative authority
- ◇ Administrator – Active with administrative authority
- ◇ Inactive – Used only to retain user information in the database while locking out that username and password for TrackRecord logins

- 6 **For TrackRecord users (Normal and Administrator), enter this user's login name in the name field.**
Optionally, enter and confirm a password. To allow users to log in without a password or to change their password the first time they log in, you may leave the Password field blank.
- 7 **Optionally, enter an e-mail address for this user.**
An e-mail address is required if the user will be using the AutoAlert utility, as described in “Configuring AutoAlert” on page 192.
- 8 **Enter a Version Control System user name and password.**
This step is required if the username is different from the name just entered and if you intend to integrate with a version control system.
- 9 **Click on the Privileges tab and select the groups to which this Team Member belongs.**
If the Team Member is not a TrackRecord user (Step 5) they cannot belong to any group; the checkboxes will be inactive.
Each group's database privileges are displayed for your information. Users can belong to up to fifteen groups and their database access privileges become those of their least restrictive group.
- 10 **Click Save and Close when finished.**

Project Configuration

Before a team starts using TrackRecord, an administrator must define a project. A project consists mainly of the tasks required to build a software application, the people who work on those tasks, and the dates on which the tasks must finish. A project, therefore, provides an organizational framework for completing a software development deliverable.

Project Administration Privileges

Project administration differs from other administrative duties in one important respect—project administration requires *no* administrative privileges, but *does* require (when you use the sample database as your starting point) that the designated project administrator belong to the Project Admin group. *The sample database restricts access to several information types to members of this group.*

Creating a Project

TrackRecord supplies a Project type in its sample database. You can use this type as shipped, or make modifications to create your own Project type. If modifications are required, make changes before creating your projects. Refer to “Type Administration” on page 156 for information about modifying types.

TrackRecord also supplies a sample project called `YourProject`. You can edit this item to create Project items for your organization or start with a new project.

To Create a New Project

- 1 From the Toolbar, click the Item Browser button and select Project.**

- 2 Click the New button in the Item Browser toolbar.**

A Project Item form opens.

- 3 Enter a name and description for the project.**

- 4 If you are using a version control system, click on the check mark in the VCS Project Name field. Log in to the source database and select the folder and project name.**

You need to associate the TrackRecord project with the VCS project to make check in/out actions meaningful.

- 5 Optionally, enter a Current Version and Current Build Number.**

The DevPartner Studio Build Registration utility uses this number as a starting point to construct build names that also contain an incrementing build number. Refer to “Registering One or More Application Software Builds” on page 175 for information on build registration.

- 6 Optionally, enter Builds/Releases, Milestones, Team Members, Documents, Functional Areas, Keywords and Sub-projects.**

Remember that a form for one type can ‘contain’ forms for other types. In the sample Project form, the fields listed in this step are separate types. Clicking the Add (+) symbol allows you to choose from the existing items of this type and to create new items of this type.

You cannot create meaningful Milestone Status views unless at least one milestone exists.

Note that you can modify the project later to add additional items. If Team Members have not yet been created, you can add them to

Projects when the member items are created, as described in “Creating Team Members” on page 138.

- 7 **Click the Save and Close toolbar button.**

Cloning Projects

When you create projects, TrackRecord lets you duplicate and reuse the milestones from an existing project. This feature allows development teams to create multiple projects that all use the same milestone events. Using identical milestones promotes consistency across projects.

To clone a Project, open the Project to be cloned (in the Toolbar, click the **Item Browser** button, select **Project** and select the Project), then select **Save and Duplicate** from the Save drop down menu in the toolbar.

Multiple Projects

Projects may share people, code, and other resources. By tracking multiple projects in a single database, a team member can see all the tasks that require completion on a given day, regardless of the project with which the task is associated. A single database makes it easier to track problems in shared code, as well.

Data from multiple projects can reveal potential problem areas. For example, if testers of Project A are finding many installation bugs, developers of Project B may want to emphasize the test and development effort on their installation module.

Historical Information

Keeping information around after you finish one project and start another gives you historical data. For example, when building a second version of a product, you could look back at the first version’s bugs and analyze which parts of the program gave you the most trouble during development. That knowledge could influence how you staff or schedule that part of the second version’s development.

Global Preferences Administration

The tasks of global preferences administration include:

- ◆ Creating Queries and Outline Reports for all TrackRecord users
- ◆ Identifying favorite types, reports and a Home Page for all users
- ◆ Creating global templates

Creating Global Queries and Outline Reports

To make the members of a project team productive as quickly as possible, administrators should provide a representative set of queries and Outline Reports to TrackRecord users. These data-extraction tools will allow users to accomplish tasks without spending time learning the intricacies of TrackRecord. Only administrators can create queries and Outline Reports that are globally shared with every TrackRecord user.

The sample database shipped with TrackRecord provides a limited set of global queries and Outline Reports that administrators can study to learn how these constructs behave. If you are unfamiliar with queries and Outline Reports, review Chapter 4, “Using Queries and Views.”

To Create a New Global Query

- 1 **From the File/New option, select Queries.**

Alternatively, if the Query already exists, use the **File/Manage** option and select the query.

The **New Query** dialog or, for existing Queries, the **Manage Query** dialog, opens.

- 2 **Create or modify the Query as described in Chapter 4, “Using Queries and Views.”**
- 3 **Click Options, click the Sharing button and click the Share to All option.**

Alternatively, you could select share groups to share this query, as described in “Share Groups” on page 137.

- 4 **Click OK to dismiss the Sharing dialog.**
- 5 **Click OK to dismiss the Options dialog.**
- 6 **Click Save and Close on the toolbar to close the query.**

To Create a New Global Outline Report

- 1 **Select File/New and choose Outline Report.**

Alternatively, if the Outline Report already exists, use the **File/Manage** option to select the Outline Report.

An Outline Report window opens.

- 2 **Create or modify the Outline Report as described in “Using Outline Reports” on page 37.**
- 3 **Click Options.**
The **Outline Options** dialog opens.
- 4 **Click the Sharing button.**

- 5 Click **Share to All** or select share groups.
- 6 Click **OK** to dismiss the **Sharing** dialog. Click **OK** again to dismiss the **Options** dialog.
- 7 Click the **Save** icon when finished.

Identifying Favorite Types, Reports, and Home Pages

TrackRecord supports many types of items and many reports, but most users will generally use only a few of these types and reports. It is helpful to make it easy for the user to find the types and reports in which they are most likely interested.

For example, when the user selects **New** from the **File** menu or from the **Toolbar**, it is helpful to list the types they are most likely to use rather than forcing the user to select from a long list of all available types.

To assist the user, you can establish global favorite types and reports or views. These types and reports will be displayed for all TrackRecord users.

Of the favorite reports and views, you can select one to be the default **Home Page** for all users. The **Home Page** is the report or view that will be displayed by default when the user logs in to TrackRecord. The user can always return to this report or view easily by clicking the **Home Page** icon on the **Toolbar**.

Individual users can set up their own favorite types, favorite reports and home page, which will override the global favorites.

To Create Global Favorite Types, Reports, and Home Page

- 1 **From the Administrator menu, select Default User Options.**
The **Default User Options** dialog opens.
- 2 **On the Favorite Types tab, select a type from the All Types list and click Add to place it in the Favorite Types list.**
This action adds the type to the **Favorite Types** list for every user of the current database.
- 3 **Use the Remove button to delete a global favorite type.**
- 4 **On the Favorites tab, click Add to add a new global favorite report. Choose the type of report - Outline, Milestone or Graph.**
The existing reports of that type are listed.

- 5 **Click a report to be listed on the Favorites menu for all users and click Open.**
The report will be added to the list of favorites.
- 6 **Use the Set as Default Home drop down list to select one of the favorite reports to be used as a default Home Page for all users.**
- 7 **Click OK when finished.**

Creating Global Templates

TrackRecord uses templates to supply default field information for new items. Global templates supply default values for every user. Users can create their own templates and make them private or share them with members of the groups to which they belong.

Creating Templates is described in Chapter 5, “Templates.”

To Create a Global Template

- 1 **Select File/Manage/Templates.**
The **Manage Templates** dialog opens.
- 2 **If a template already exists, select it. If not, click New and click the type.**
The template form opens.
- 3 **In the Summary field, enter a descriptive name for this template.**
This name will identify this template in the **Manage Templates** dialog. It will not be displayed when the template is used to enter items.
- 4 **Enter the information to be displayed to all users.**
- 5 **Click Options.**
The **Template Options** dialog opens.
- 6 **Check the Apply This Template to All New Items box.**
- 7 **Click the Sharing button and check the Share to All option. Click OK.**
- 8 **When all template items have been set, click Save and Close.**

Workflow Administration

A Defect, Task, or other item can pass through a series of states between the time it is entered and the time it is closed. For example, when a Defect is entered it is generally in an Unreviewed state. It might then be assigned to someone, resolved, and validated. This series of actions, and the states resulting from these actions, is called a workflow.

As TrackRecord Administrator, you will establish the workflow for the items created with TrackRecord.

Tracking the state of items allows for more accurate assessment of a project's progress. For example, program managers might determine how to best allocate resources by examining all Defects in the Pending Validation state or those in the Unassigned state.

Before implementing a workflow, you must think about your defect or task resolution process – what states items can be in, who has authority to change each state, and so on. You use TrackRecord groups to define who can move an item from one state to the next. You can also create different workflow paths and privileges for specific groups.

Once you have thought through your task management plan, implementing a workflow is a two-step process:

- ◆ Creating the workflow, as described in this section.
- ◆ Including a Status field in a type, as described in “Adding a Status Field to a Type” on page 153.

The presence of a status field in a type causes an Action button and Action list to be displayed on the type form. Users will use the Action button to indicate the actions they perform on the item. TrackRecord will then automatically set the item's state. The actions listed in the Action list, and the state to which the item is set after each action, is determined by the workflow transitions you define.

The sample TrackRecord database includes a Status field on the Defect type. To apply workflow rules to other types, such as Tasks, you must add a Status field to their type definitions.

Each TrackRecord database uses only one workflow, however you can design the workflow to use separate rules for different types. Refer to “Designing a Workflow Based on Type” on page 153 for more information.

You can further customize your workflow by implementing rules that disable the standard workflow in certain situations. Disabling the workflow causes the Action button to be disabled, which will prevent a user from changing the status of an item even if the standard workflow

would normally permit them to do so. Refer to “Adding Rules That Control the Behavior of Types” on page 174 and to the online help for information on creating rules.

The TrackRecord sample database contains types that you can use to implement workflow for your project data. The sample database building blocks for workflow include the types:

- ◆ Action – Enter, Open, Validate, Close, and others
- ◆ Status – Unreviewed, Open, Closed, and others

The following procedures describe setting up workflow using the building blocks from the sample database.

Creating and Modifying a Workflow

A workflow is created and modified with the Workflow Editor utility.

The Workflow Editor assumes the existence of certain types and tags in your database. If your database does not contain these types and tags (for example, if you are using an older database or if you have modified the types in the sample database), the warnings described in Step 2 below are displayed to allow the Workflow Editor to make minor adjustments to your database.

To Display the Workflow Editor

- 1 From the Administrator menu, select the Edit Workflow option.**
- 2 Two warnings may be displayed advising you to backup the database and to request users to log off.**

If these warnings are displayed, close TrackRecord. The warning dialogs, and, subsequently, the Workflow Editor, will remain active.

Refer to “Backup and Restore Database” on page 181 for backup instructions.

Refer to “Database Lock-out” on page 159 for information about the Active User dialog. When all users have logged off, the dialog is dismissed, and the Workflow Editor makes database modifications and unlocks the database. The Workflow Editor opens and displays the current workflow.

Figure 11-3 illustrates the Workflow Editor and the default workflow for the sample database shipped with TrackRecord.

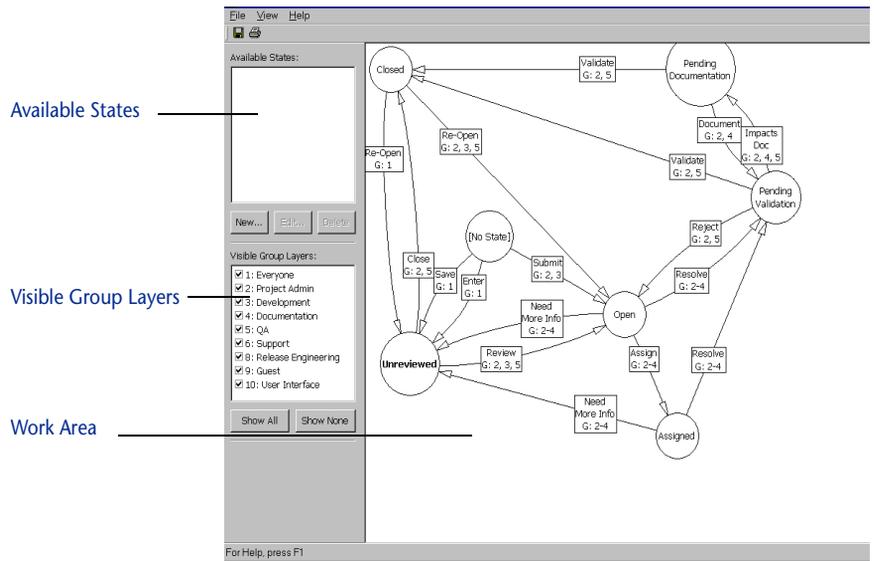


Figure 11-3 Workflow Editor

In addition to the standard menu bar and toolbar, the Workflow Editor consists of three frames:

Available States: Lists states that have been defined but are not yet used in the workflow. Using the buttons, you can create a new state, edit a state definition, or delete a state, as described in following sections.

Visible Group Layers: Lists all groups that have been defined. Checkmarks indicate the groups whose workflow is visible. The buttons allow you to quickly display the workflow for all or no groups. By default, the workflow for all groups is displayed. To view the workflow of a subset of groups, uncheck the groups that are not to be displayed or use the **Show None** button to remove all groups, then check the group(s) to be displayed. The workflow for Everyone is always visible (as a watermark if not selected) to prevent the addition of unnecessary transitions.

Work Area: The work area is where you create the workflow. Circles represent states; rectangles represent actions; arrows indicate the state transition caused by that action. For example, if a defect is in the Unreviewed state and the action Review is performed, the defect state will change to Open.

The action text boxes also indicate the groups that are permitted to perform that action. The numbers correspond to the numbers in the Visible Group Layers frame.

Moving the cursor over a state or action will display a tooltip describing the state or action.

Working with States

To change a state, select it by clicking on the state circle in the work area. You can then do one of the following:

- ◆ Move the state. Moving a state may increase the readability of your workflow. All actions remain attached. To move a state circle, click in the circle to select it, drag the circle to the desired location and release.
- ◆ Remove or delete the state. Right click on the state circle and select a menu option to either remove the state from the workflow or to delete the state.

If the state is removed, it will be listed in the Available States frame and can be reinserted into the workflow. If it is deleted, the state will no longer be available. (You can recreate the state if necessary using the **New** button in the Available States frame.) You will be prompted to confirm the delete operation. All transitions leading into or out of this state will also be deleted.

- ◆ Create a new transition that allows an item to leave this state, as described in “To Create a New Transition” on page 152.

To Create a New State

- 1 **Click on the New button in the Available States frame.**

A State Properties dialog is displayed, as shown in Figure 11-4.

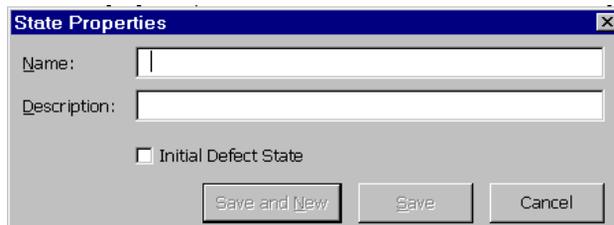


Figure 11-4 State Properties Dialog

- 2 **Enter the name of the new state.**

You should generally not create a state with the same name as an existing state.

- 3 **Enter a description for this state.**

4 If all Defect items should be placed in this state by default, check the Initial Defect State option.

Note that the default state for other types, such as Tasks, is set through the use of templates, as described in “Default Status for New Items” on page 155.

5 Click Save and New to continue creating new states or click Save if you have completed creating new states.

The state is listed in the Available States frame.

To use the new state, click on the name of the state in the Available States frame and drag it onto the work area. Create transitions that cause the item to leave this state, as described in “To Create a New Transition” on page 152. Click in other states and create transitions that cause items to enter this state.

Working with Transitions

Each state can have multiple transitions. For example, you might want to have two possible actions for an item in the Open state: Assign and Reject. Further, you might want the Reject action to result in a different state depending on who entered that action. You would create three transitions:

- ◆ The Assign transition leading to the Assigned state.
- ◆ A Reject transition leading to the Closed state if it is entered by a member of the Project Admin group.
- ◆ A Reject transition leading to the Unreviewed state if it is entered by a member of any other group, which would allow the defect to be reviewed again.

To change a transition, click on any part of the transition line. Handles (small black squares) are displayed on both ends of the transition line. Click on a handle and drag it to a new state. You can change the state in which the transition is generated or the state in which the transition results.

To change the properties of a transition, double click on any part of the transition line. A Transition Properties dialog is displayed. You can change the name of the action or the groups that can perform the action.

To delete a transition, right click on the action and select **Delete** from the menu. The transition is deleted from the work area. The action associated with the transition, however, is not actually deleted. Once

an action name has been created, it is never removed from the drop-down list in the Transition Properties dialog.

To Create a New Transition

- 1 **Click in a state circle to select it.**

When you move the cursor, an arrow is displayed.

- 2 **Click on the handle (the small black square) at the end of the arrow, drag it to another state, and release.**

You must release in another state circle or the transition will not be created.

A Transition Properties dialog is displayed, as shown in Figure 11-5.

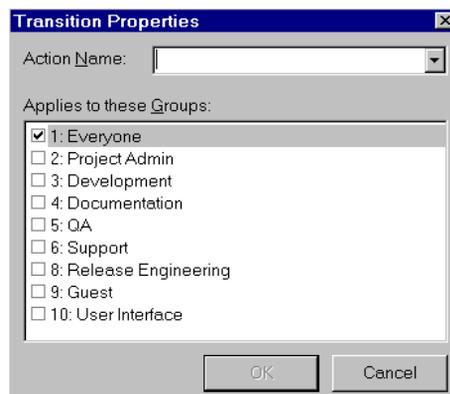


Figure 11-5 Transition Properties Dialog

- 3 **Select an action from the drop-down list or enter the name of a new action.**

Note Do not create two identical actions from the same state for a given group that result in different states.

- 4 **Check the groups that will be allowed to perform this action from this state.**

Users belonging to more than one group will be able to perform actions permitted for all groups to which they belong.

- 5 **Click OK.**

A transition line is created in the work area. The path of the transition line, the location of the text box and the point at which the line intersects the state circle are determined automatically and cannot be altered.

Designing a Workflow Based on Type

Each TrackRecord database can have only one workflow, but you can design that workflow to allow different behavior based on the item type. To accomplish this, define a separate set of states and transitions for each type. For example, Figure 11-6 shows a separate set of states and transitions for Defects and Tasks.

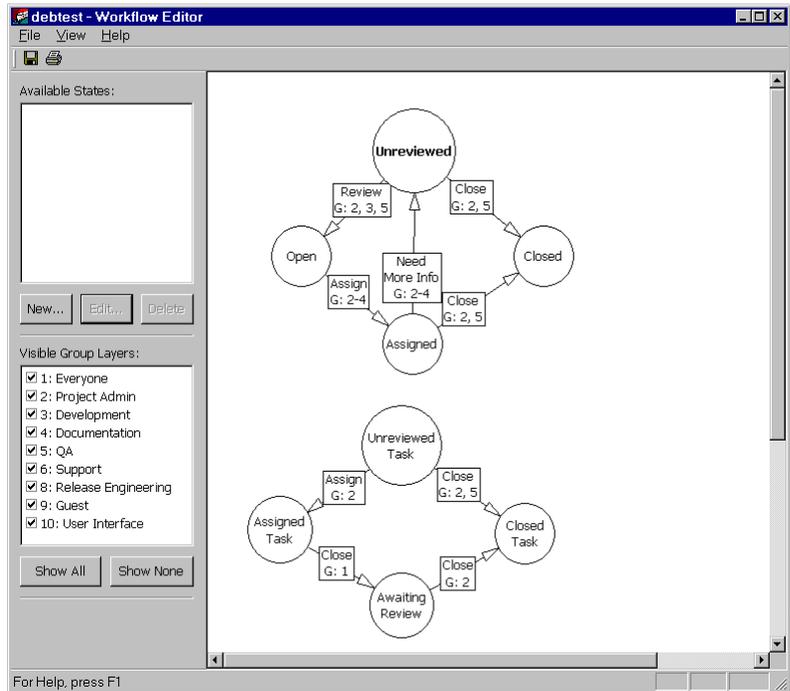


Figure 11-6 Separate Workflow Based on Type

Each type's initial state determines which set of workflow rules (states and transitions) the type will follow. The initial state can be set from within the Workflow Editor only for the Defect type. For all other types, you can set an initial state by creating a template for the type and prefilling the Status field, as described in "Default Status for New Items" on page 155.

Each set of states should use unique state names.

Adding a Status Field to a Type

The TrackRecord sample database includes a status field on the Defect type. To apply workflow rules to other types, such as Tasks, you must add a status field to their type definitions.

Refer to “Creating and Modifying Types” on page 161 for a general discussion of adding fields to types. The following steps are specific to adding status fields.

To Add a Status Field to a Type

1 Log on to TrackRecord as an Administrator.

2 Use the Administrator menu’s Edit Types option to open the Type Editor.

If users are logged on using this database, a dialog will be displayed listing current users. Send a message requesting users to log off. (Refer to “Database Lock-out” on page 159 if you would like a description of this dialog.) When no users are logged on using this database, the **Choose a Type** dialog opens.

3 Open the type to which you want to add a Status field.

4 Click Insert Field and choose Single Item Combo Box.

5 Move the pointer around the form. The pointer indicates where the field will be added.

If the type inherits from a parent type and you place the new field in the inherited section, the new field will be added to all types inheriting from that parent.

6 Click the mouse to drop the new field onto the Type Editor layout region.

The **Field Properties** dialog opens.

7 Enter a name for the field.

The name will be the label for the field. A logical field name might be Status or State.

8 Click Tags to open the Tags dialog.

9 Select TR_Item_State and click Add and OK.

10 From the Type drop-down list, select Status.

11 Check any additional options for this field.

12 Click OK.

The field will be displayed on the type form. Other fields will be resized or repositioned to accommodate the new field.

To change the field’s properties, right-click on the new field and select **Open**. To change the location of this or any other field, click on the field and drag it to another location.

13 Click OK or Save and Close to close the Type Editor.

14 Click Close to close the Choose a Type dialog.

15 Create a template to set a default status, as described in “Default Status for New Items” on page 155.

A type should have only one Status field. If a type contains more than one status field, the first one encountered in the form will be active. The second will be ignored.

Default Status for New Items

For all types that include a status field, you should specify a default state to which new items will be set. If you do not set a default state, users will be able to select any available status when they first create an item of this type, and the Action button will not be displayed until a status is set and the item reopened.

For Defect types, you can set the default state through the Workflow Editor, as described in “To Create a New State” on page 150.

For other types, you must set the default state through a template.

To Set the Default State Through a Template

1 Select File/Manage/Templates.

The **Manage Templates** dialog opens. If necessary, use the **Show** drop down menu to view a subset of templates.

2 If an appropriate template already exists, select it and click Edit to modify.

If an appropriate template does not exist, click **New** and click the type.

The template form opens.

3 In the Summary field, enter a descriptive name for this template.

This name will identify this template in the Manage Templates dialog.

4 In the Status field, choose a default value for Status, such as Unreviewed or Open.

5 Click Options to open the Template Options dialog.

6 Make sure the Apply This Template to All New Items box is checked.

7 Click the Sharing button and check the Share to All option. Click OK.

8 When all template items have been set, click Save and Close.

When a user creates a new item of this type, the initial status will be set.

Type Administration

TrackRecord information types, such as the Person type, Defect type, and so on, provide the framework on which the TrackRecord database is built. As an analogy, think of these types as the data entry forms of a traditional database—with two important differences. First, types can inherit the properties of other types. Second, the fields on a form can contain other forms.

You can use the default types provided with TrackRecord, you can modify these default types, and you can create new types. Before modifying or creating new types, be sure to assess the impact these types will have on your development process. Once types have been used by TrackRecord users, the changes you can make to them will be restricted, as described in “Restrictions” on page 172.

Type Inheritance

When one type inherits from another, the inheriting type (the child) receives a copy of all the fields of the original type (the parent). These fields always appear before any new fields that the child type adds. Child types also inherit all the field-level privileges of the parent type, but not any type-level privileges. (Field- and Type-level privileges are described in “Altering Access Rights for Types and Fields” on page 158.)

Type inheritance provides another benefit—you can add another identity to an item of one type. This new identity must be that of an item of one of the original item’s child types. For example, a person in a database (based on the sample TrackRecord database) whose name appears when you highlight the Person type in the Item Browser can be transformed into a Customer or a Team Member, which are two child types of the Person type. Or a Customer could be transformed into a Team Member. This property becomes valuable when someone in your database changes roles on a project—an administrator does not need to remove and reenter information about that individual.

Type inheritance also permits querying on a parent type to retrieve information about all items of that type, including its child types. For example, a query that looks for Assignments will retrieve items of the Assignment type, and the child types of Assignment, namely Defect, Feature, and Task.

Categories of Types

TrackRecord information types define the structure of the individual pieces of information stored in a project database. Information types, therefore, constitute the heart of TrackRecord. Manipulating types provides control over the structure of the database.

Although all types are equal from a technical point of view, administrators should think of types as falling into two categories:

- ◆ *Common* types
- ◆ *Restricted* types

Usually, common types have privileges set so that most users can create and modify items of these types (Defects, Tasks, etc.). Restricted types usually have privileges set so that only administrators can manipulate them.

Administrators can determine whether a type is common or restricted with the Type Editor property **Display to All Users**, as described in “Creating and Modifying Types” on page 161. Common types will be displayed to all users; restricted types will not.

Administrators can use the **Show All** check box on the **Item Browser** to view both common and restricted types. Unchecking the **Show All** check box changes the **Item Browser** display to common types only.

Conceptually, restricted types fall into two categories as well: *administrative* and *contained*. Administrators use administrative types for items that are used behind the scenes. For example, the Priority and Resolution types are used to create items that become the building blocks of other structures.

Contained types are created from within another item and would have no use outside the context of the containing item. For example, the Phone Number field within a Team Member item actually holds a Phone Number item. Phone Number items, however, have no meaning as stand-alone items.

Abbreviations

When an item is displayed in a report, TrackRecord must display at least one field from the item. The field that is always displayed is called the Default Abbreviation. Each type must identify one Default Abbreviation. In addition, each type can have one Short and one Long Default Abbreviation.

TrackRecord creates a Default Abbreviation based on the first field in a type. You can change this default, and you can specify a Short and

Long abbreviation, with the Type Properties dialog, as described in “Creating and Modifying Types” on page 161.

Identifying Duplicate Imported Items

When items are imported into the TrackRecord database, TrackRecord must determine if the items are duplicates of existing items. Each type must contain one field or combination of fields to be compared with imported items to determine duplicates. This field or fields is identified with the **Dupe** option on the Type Properties dialog, as described in “Creating and Modifying Types” on page 161. If you do not specify a Dupe abbreviation, the Default Abbreviation will be used.

Altering Access Rights for Types and Fields

Access rights can be set for either an entire type or the individual fields within a type.

Type permissions inherit the access control specifications that apply globally, and then let you restrict access further within each group. In other words, if a user is assigned to the Guest group only, and the global permissions for that group is Read, you cannot use type permissions to grant Write permission for an item of any type.

Type permissions use the same names as global permissions: Read, Write, and Add.

To delete an item, a user must have write access to the item type and to every field in the type.

Field permissions inherit the specifications of global and type restrictions and, optionally, can refine permissions to the field level to add another restriction called Hide. For example, if you want to prevent group members from QA and Documentation from knowing the name of a particular project’s client, you can restrict that field by hiding it while leaving the other fields visible.

As with type permissions, field permissions can be changed in the Type Editor. You can use the Read and Write restrictions at the field level, but not Add.

ActiveLink Tags

TrackRecord administrators can name types, and label the fields on types, to match their company’s development process. For example, if

the organization calls software defects “bugs,” they can name a defect type “Bug.”

Note Milestone Status views depend on certain ActiveLink tags. If these tags have been removed or altered, the Milestone Status option might not be available.

DevPartner Studio depends on certain types, and fields within types, to properly integrate its developer tools with the TrackRecord database to gather information automatically. To provide for this dependency, TrackRecord supplies ActiveLink *tags* that let administrators link the types and fields they create with the underlying structure that ActiveLink expects. If you use the types shipped with the sample TrackRecord database, you can change type and label names without concern for ActiveLink tags. Use ActiveLink tags only when you create new types, or delete fields from an existing type. The on-line help describes these tags.

Database Lock-out

Certain changes to the database, such as creating or modifying types, cannot be made while users are logged on using the database. If you attempt these changes while the database is in use, a dialog will be displayed listing current users and allowing you to advising them to log off.

To send a message requesting that users log off, click the **Send Message** button, edit the default message if desired and click **Send**. The message is displayed in the TrackRecord client of all logged on users, including those accessing the database through the WebServer. Note that the timing of the message display depends on each user’s polling frequency setting.

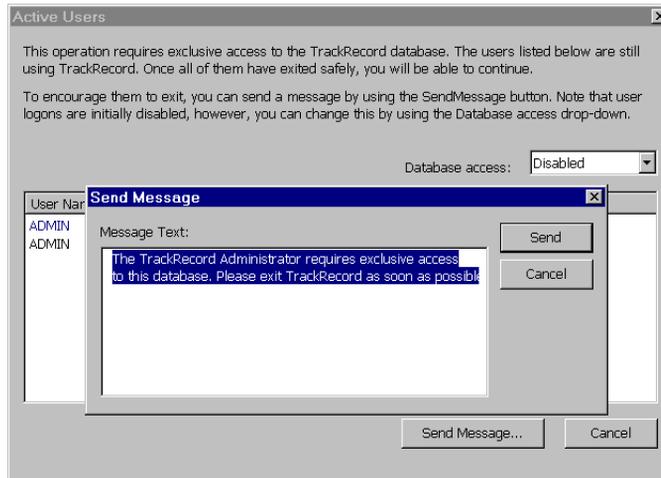


Figure 11-7 Database Lock-out

By default, new logins are disabled when this dialog is displayed. You can use the drop-down menu to enable logins; clicking **Cancel** will also re-enable logins.

If a user attempts to log in while logins are disabled, a message box will inform them that logins are disabled. They can choose to log in using a different database.

The user list is refreshed every few seconds. Once all users have logged off, this dialog is dismissed and you can make your desired changes.

If the database is being accessed without a logged on user, for example, by AutoAlert, WebServer, BoundsChecker, or Microsoft Project, the dialog will include an Automation entry. The administrator must ensure that these processes are shut down. BoundsChecker users must exit BoundsChecker.

Type Administration Tasks

The tasks of type administration include:

- ◆ Creating new types and modifying existing types
 - ◇ Adding and naming fields
 - ◇ Laying out, sizing, and positioning fields
 - ◇ Altering the access rights to types and individual fields
- ◆ Creating custom items for Choice types (optional)
- ◆ Creating rules for the type (optional)

Development teams can create any information types they need to implement any development process they choose. Until a team

understands the possibilities, however, defining information types might be a haphazard process.

To assist teams in their understanding of information types, TrackRecord ships with a sample database populated with information types that TrackRecord's designers—who develop software themselves—created to set up one kind of development process. By studying these types, administrators can develop an understanding of how types integrate to implement a process.

To study the types in the sample database, create new items of a given type to see what kinds of information that type requires. Also, users with administrative privileges can use the Type Editor to open a type and view its fields, including its properties and access rights.

Creating and Modifying Types

While companies can use the sample database shipped with TrackRecord as a production database, many organizations will see the need to add to or modify the information types supplied. The procedures that follow explain type creation and modification.

To Create or Modify a Type

- 1 Log on to TrackRecord as an Administrator.**
- 2 Close all TrackRecord items and views (defects, reports, etc.).**
- 3 From the TrackRecord Administrator menu, click Edit Types.**

A warning is displayed advising you to backup your database before modifying types. Refer to “Backup and Restore Database” on page 181 if you would like to backup your database at this time.

- 4 If users or other TrackRecord clients, such as the WebServer, are logged on using this database, a dialog will be displayed listing current users.**

Refer to “Database Lock-out” on page 159 for information about this dialog.

When this dialog is dismissed, the Choose a Type dialog is displayed.

- 5 If you are creating a new type, on the Choose a Type dialog click New and proceed to Step 7.**

If you are modifying a type, click on the Type and click the Edit button.

- 6 To modify type properties, click the **Type Properties** button and proceed with Step 7.
If changing field properties only, skip to “Using the Type Editor” on page 164.

7 On the **Type Properties General** tab (Figure 11-8):

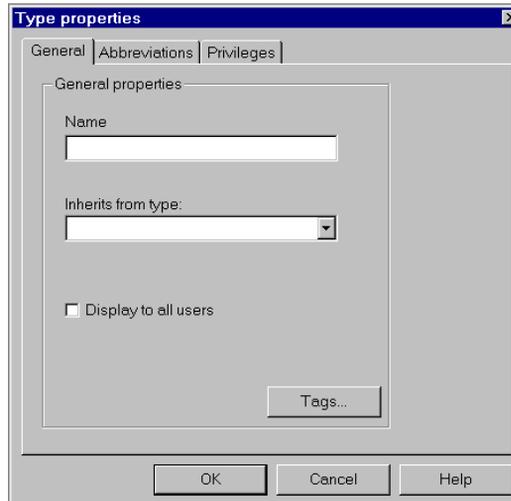


Figure 11-8 Type Properties General tab

- ◇ Enter a name for the type.
- ◇ Optionally, use the **Inherits from Type** menu to select a type from which your new type will inherit information. Refer to “Type Inheritance” on page 156 if you are unfamiliar with this concept.
- ◇ Check the **Display to All Users** check box to designate this type as a common type, or leave it blank to make this type a restricted type. Refer to “Categories of Types” on page 157 if you are unfamiliar with this concept.
- ◇ Use the **Tags** button to add tags to this type. See the online help for information on tags.

8 **On the Abbreviations tab, identify the field(s) to be used as Default Abbreviation(s).**

Setting Abbreviations is optional.

Refer to “Abbreviations” on page 157 for a description of how abbreviations are used.

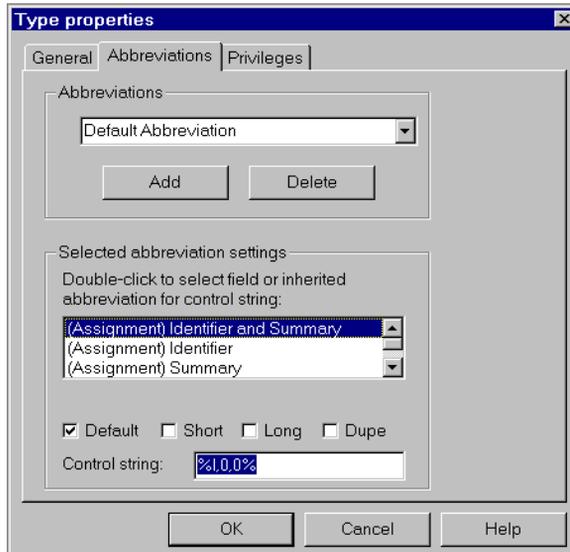


Figure 11-9 Type Properties Abbreviations Tab

If you are creating a type that does not inherit from another type, the list of fields will be empty until you create fields, as described in later steps. You can return to this tab after you add fields to the form. If this type inherits from another type, the inherited fields are displayed.

To change the Default abbreviation, select **Default** in the **Abbreviations** list. In the **Selected Abbreviation Settings** list, select a field (or combination of fields), and check the **Default** option.

To set the Short and Long abbreviations, click the **Add** button and add a name for this abbreviation, then select the field or combination of fields from the **Selected Abbreviation Settings** list and check the **Short** or **Long** option.

To specify a field to be used to identify duplicate items when importing data, select the field or combination of fields and check the **Dupe** option. If you do not specify a Dupe abbreviation, the Default Abbreviation will be used.

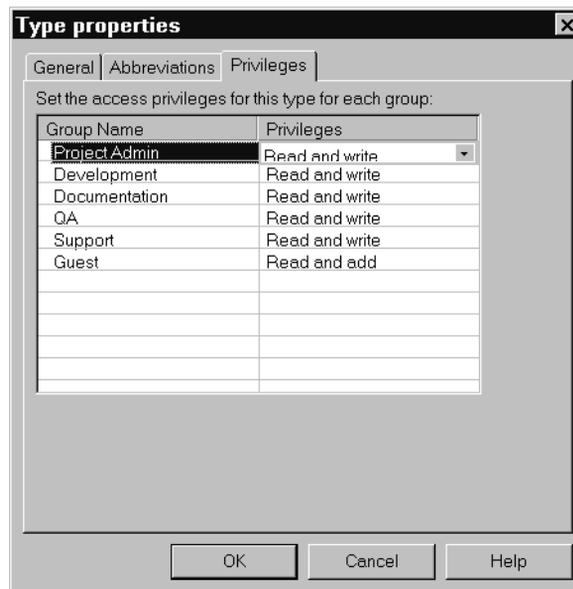
The control string is similar to a C-style control string. It can contain alphanumeric characters, which are duplicated in the abbreviation, and one or more control sequences, which are used

to put information from the item into the abbreviation.
TrackRecord automatically creates the control string.

- 9 **To restrict access to this type, on the Privileges tab click a group and use the drop-down menu to select the appropriate privilege for this group.**

Note While you can change the privilege setting for any group, you cannot raise the ceiling on that group’s privileges beyond the upper limit established through Group Administration.

Refer to “Altering Access Rights for Types and Fields” on page 158 for a general discussion of access privileges.



- 10 **When you have completed entering information on these three tabs, click OK to open the Type Editor.**

You can return to make changes on this dialog by using the **Type Properties** button in the Type Editor.

Using the Type Editor

You use the **Type Editor** dialog to insert fields into a form, modify fields on a form, or to modify settings on the **Type Properties** dialog.

To Use the Type Editor

- 1 Follow the steps in “To Create or Modify a Type” on page 161 to display the Type Editor.

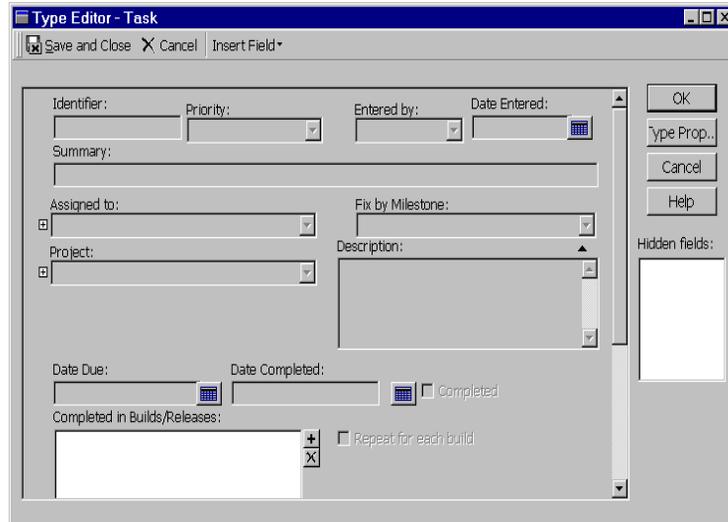


Figure 11-10 Type Editor

To modify Type properties, click the **Type Properties** button.

Refer to “To Create or Modify a Type” on page 161 for information about type properties.

- 2 To move a field, click and hold that field and drag it to another location.
- 3 To modify the properties of a field, right-click on the field and select Open.

The **Field Properties** dialog opens. Refer to the following sections for a description of the **Field Properties** dialog.

- 4 To insert a field into the type, drop down the Insert Field menu (Figure 11-11) and select a field category.

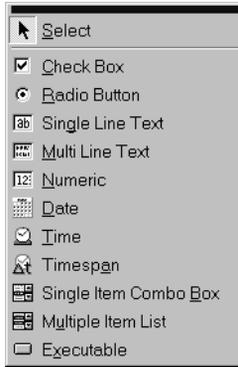


Figure 11-11 Insert Field Menu

A pointer is displayed in the form, indicating where the field will be added. Move the pointer around the form and click the mouse to position the field.

If the type inherits from a parent type and you place the new field in the inherited section, the new field will be added to all types inheriting from that parent.

When you drop the field into a position, a Field Properties dialog is displayed.

There are three tabs on the Field Properties dialog, each of which is described in the following sections.

Field Properties - General Tab

The Field Properties General tab (Figure 11-12) displays the following options:

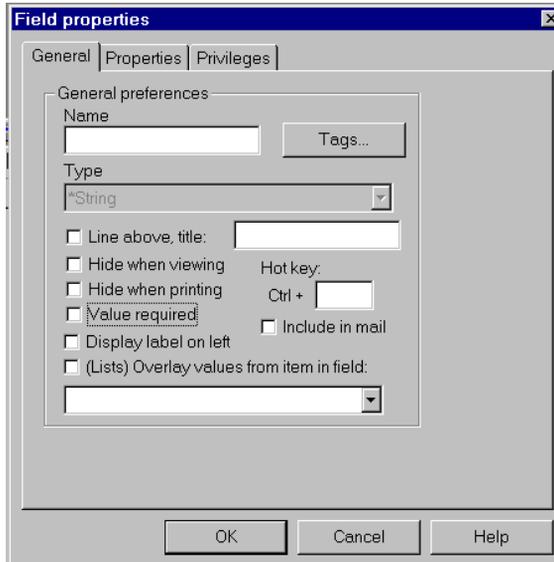


Figure 11-12 Field Properties - General Tab

- ◆ A name for the field (required).
- ◆ A field type specification. In general, you cannot change the field type of an existing field. If, however, you have no items in the database of the type you are editing, you can change its type. The field types you can select depend on the palette tool you used to create the field. For instance, a field created with the date tool cannot be specified as a Phone info field type.
- ◆ A **Line Above Title** checkbox for dividing your form into labeled sections. You can also put static text on the form by setting an optional title for the line.
- ◆ Two checkboxes for hiding the field and its contents onscreen and while printing.
- ◆ A **Hot key** text box (optional). The hot key is used to edit a field directly from within other views, without using the Item view.
- ◆ A **Value Required** checkbox to determine if this is a required field.
- ◆ A **Display Label on Left** checkbox to display the label on the left of the field rather than above it.
- ◆ An **Include in Mail** field to determine if this field should be included in the mail message if the AutoAlert utility is used.
- ◆ The **(Lists) Overlay Values from Item in Field** checkbox specifies that if this field is empty, check the item in the specified field for any fields named the same as this field. If an identically named field is found, the contents of that field are displayed in this field. For example, this property can be used to overlay the address and

phone fields from a Company into any Person items that contain that Company. The effect is similar to that of applying a template; the values from the selected item are used, but no implicit links are created. If the field contents are changed in any way, this action is ignored.

Field Properties - Properties Tab

The contents of the Properties tab will vary depending on the type of field you are creating or changing.

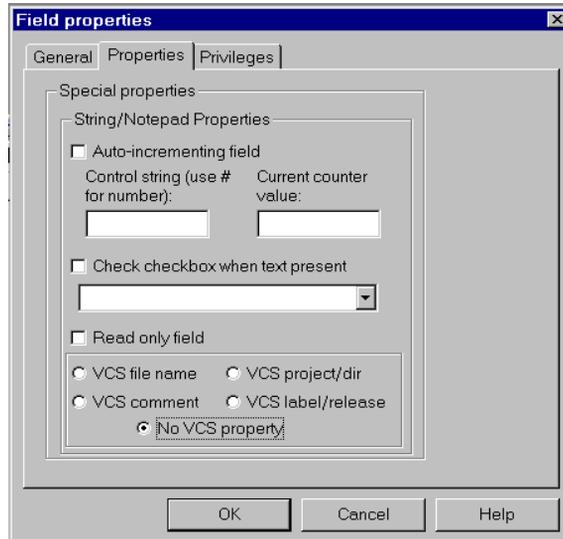


Figure 11-13 Field Properties - Properties Tab

◆ Compound Type Properties

Compound type fields contain other TrackRecord items, and are displayed in an Item as drop-down lists and regular list boxes.

- ◇ The **Is a List** checkbox indicates whether the field offers a list of items using a standard Windows list box. Once the list box property has been applied to a field, it cannot be removed.
- ◇ The **Constrain Choices by Values in Field** checkbox specifies that when choosing an item for this field, TrackRecord filters the items shown according to this item. For example, the **Fix by Milestone** field in a Defect Report could be constrained by the **Project** field. Then, a list of Fix by Milestone Defects would be limited by Project.
- ◇ The **Create New Items when Adding to List** checkbox is only applicable for list fields. If it is checked, a new item of the

labeled type is automatically created when the **Add** or **Ins** button is clicked, and deleted from the list when the item of that type is removed.

- ◇ The **Expand When Viewing** checkbox specifies that non-list fields should be expanded inline, showing all of the sub-fields, by default, when viewing an item in TrackRecord.
- ◇ The **Expand When Printing** checkbox specifies that non-list fields should be expanded inline, showing all of the sub-fields, by default, when printing an item.
- ◇ The **VCS Source Files** checkbox indicates that this field contains a list of VCS source files to be used and updated when checking files in and out from TrackRecord.

◆ **String/Notepad Properties**

The String Properties dialog appears for all string and Notepad fields.

- ◇ The **Autoincrementing Field** checkbox automatically assigns a value to the field when an item containing the field is initially created. This attribute is primarily useful for creating fields like task identifiers, when you want a unique number to identify each new task.
- ◇ The control string is a text string that is used to create the identifier. You should use a pound sign (#) in the string where you want the number to appear. For example, to get identifiers of the form “TrackRecord-00001”, you would use the control string “TrackRecord-#”.
- ◇ The **Current Counter Value** is an optional integer that specifies the current value for the counter number. You can change this value at any time.
- ◇ The **Check Checkbox When Text Present** checkbox specifies that the named field should be checked if any text is present in the field, and should be unchecked if no text is present.
- ◇ The **Read Only Field** checkbox indicates that the field cannot be changed in items. You would ordinarily use it only with auto-incrementing fields, or with fields that will automatically be filled from templates in all cases.
- ◇ The **VCS File Name** checkbox indicates that the field will contain a file name from the version control system. The only way to select the value of a VCS file name field is by choosing the file name from the version control dialog.

- ◇ The **VCS Project/Dir** checkbox indicates that the field will contain a project or directory name from the version control system. The only way to select the value of a **VCS Project/Dir** field is by choosing the project/directory from the version control dialog. You would normally place a field with this attribute both in the project item (to be used as a default for new source and data file items), and in the source and data file items themselves (to allow for the possibility that the VCS project may be different on a file-by-file basis).
- ◇ The **VCS Comment** checkbox indicates that the field is used to initialize the check-in comment field when checking files in from an item of this type. In the standard Defect type, the Description field is used as the VCS Comment.
- ◇ The **VCS Label/Release** checkbox indicates that the field contains a VCS release string. The only way to select the value of a VCS release field is by choosing the release from the version control dialog.
- ◇ The **No VCS Property** checkbox indicates that a VCS is not being used.

◆ **Checkbox/Radio Button Properties**

The Checkbox/Radio Button Properties tab appears for all check box and radio button fields.

- ◇ The **Enter Date When Checked** checkbox specifies that the named field, which must be a Date or a Time/Date field, should be filled with the current time and date when the checkbox is checked.
- ◇ The **Launch Executable When Checked** checkbox launches a specified executable file when checked. The list of executable files you can specify appears under Updated Field.
- ◇ The **Update Recurring Times** checkbox actually creates a button instead of a Check Box that, when clicked, advances the starting date of the specified time-related field. This button field is only used for recurring times; for example, for an Appointment that occurs every day, this button could be clicked to indicate that today's occurrence of the Appointment has passed. The first time that the button is clicked, the date of the field advances to "today," if it was previously earlier than "today." On subsequent clicks, or if the date was already set to "today", the date advances to "tomorrow."

- ◇ The **None** checkbox allows you to clear previously checked options.
- ◇ The **Required checkbox for parent lists** checkbox to indicate whether this field is required.
- ◇ An **Updated Fields** drop down menu to indicate which fields will be updated.
- ◆ **Date Properties**

The Date Properties dialog appears for all date- and time-related fields: dates, times, and time spans.

 - ◇ The **Initialize to Current Date/Time** checkbox initializes the value in the field to the current date and time when a new item with this field is created.

Field Properties - Privileges Tab

The Privileges tab allows you to specify which groups will have access to this field. Refer to “Altering Access Rights for Types and Fields” on page 158 for a discussion of access privileges.

Note While you can change the privilege setting for any group, you cannot raise the ceiling on that group’s privileges beyond the upper limit established through Group Administration.

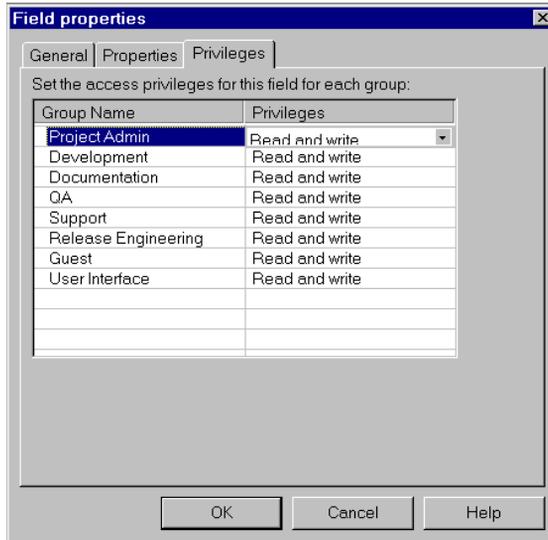


Figure 11-14 Field Properties - Privileges Tab

- 1 Click in the Privileges column of the group whose access rights you want to modify.
- 2 Click the new privileges setting and click OK.

You can use the Field Properties General tab to Hide a field to further control access rights to a field. Making a field invisible does not disable the field and a template could supply a value to that field, but the template supplied field would not be visible.

When all Field Properties have been set, click **OK** to see the field on the type form. When all fields have been added, you can click the **Type Properties** button to set Abbreviations, if desired, as described in Step 8 on page 163.

When all type properties and fields have been set, click **Save and Close**. The **Choose a Type** dialog is displayed, showing the newly created or modified type.

Restrictions

TrackRecord places some restrictions on the modifications administrators can make to information types.

- ◆ You can delete only the last field added to a type if items of this type already exist in the database.

- ◆ You cannot move an inherited field past the boundary of the parent fields when a type inherits from another type. (Note that you can, however, add a new field to the inherited section.)

Modifications you can always make include:

- ◆ Adding new fields
- ◆ Deleting newly added fields
- ◆ Changing the label of a field
- ◆ Hiding a field
- ◆ Moving fields to change the layout of the type
- ◆ Changing the access rights to a type or to a field
- ◆ Changing properties, such as requiring input for a field

Custom Items

The TrackRecord sample database ships with ten Choice types, including Department, Priority, Severity, and others, and several choice items for each of these types. For example, the Priority type includes the choices Critical, High, Normal, Low, and Suggestion. You can use these Priority items, or you could, for example, delete the Suggestion choice item and create a new choice item called Wish List.

Note that Status and Action choice items should be set through the Workflow Editor rather than through this procedure. Refer to “Workflow Administration” on page 147.

To Create or Delete a Custom Item

- 1 Click the Tools menu and click Item Browser.**
- 2 Click the Show All check box.**
- 3 Under the Choice Fields type, select the type for which you want to create a custom item.**

For example, under the Choice Fields type you could select the Priority type to create a custom item of the Priority type.

A list appears under Items.

- 4 To delete an item, right-click on the item and select Delete.**
For example, you could right-click on the “5 - Suggestion” item of the Priority type, click **Delete** and click **Yes** in the confirmation dialog.
- 5 To create a new item, click the New icon in the upper left section of the Item Browser toolbar.**

An item form opens.

6 Enter the information for the new item.

For example, for a Priority item you could enter “5” in the Name field, and “Wish List” in the Description field. Other item types, such as Department, only contain a name field.

7 Click Save and Close.

To Modify a Custom Item

1 Click the Tools menu and click Item Browser.

The Item Browser dialog opens.

2 Click the Show All check box.

3 Under Choice Fields, select a type to modify.

For example, select Department in the Types list to modify a Department item.

4 Double-click the item to be modified.

For example, double-click the Documentation item.

The item form opens.

5 Make the modification.

For example, you could replace the Name “Documentation” with “Tech Pubs.”

6 Click Save and Close.

Adding Rules That Control the Behavior of Types

TrackRecord provides a Rules engine that permits creating policies that govern the behavior of information types. Administrators can use rules to require that certain fields of information must be supplied when certain other fields have a specified value. For example, an administrator might create a rule that states, “If Priority is equal to Critical, then Fix by Date is required.”

Administrators create rules for individual types, and can create multiple rules for any given type. Rules apply either globally, or to users who belong to specified groups. Rules use the same operators as TrackRecord queries, except for the “is between” operator.

A child type inherits any rules created for its parent, and inherited rules are applied before a type’s own rules. A type’s own rules are applied in the order they appear within the Rules editor, and administrators can rearrange this order. Multiple conditions within a single rule provide an AND logical operation and all the conditions must be true for the rule to be applied. Separate rules provide the logical OR operation.

Rules are applied upon:

- ◆ A Save action
- ◆ A loss of focus on a field
- ◆ A change to a radio button or check box
- ◆ The execution of a workflow Action based on the contents of the item before the Action

Rules take the form:

```
IF {condition}, THEN {action}
```

To Add a Rule

- 1 **Click the Administrator menu and click Edit Rules.**

The **Choose a Type** dialog is displayed.

- 2 **Double-click a type.**

The **Rules** dialog opens.

- 3 **Click New to add a new rule.**

The **Rules Wizard** dialog box opens.

- 4 **Use the Rules Wizard to build the rule.**

On the first screen, create the IF condition and click **Add** and **Next**.

On the second screen, create the THEN condition and click **Add** and **Next**. On the third screen, enter a name for the rule, select groups to which the rule will apply, and click **Finish** to close the Rules Wizard. The **Rules** dialog opens.

Use the **Rules Wizard** on-line help for additional assistance.

- 5 **When multiple rules exist, use the Move Up and Move Down buttons on the Rules dialog to rearrange the order in which rules are applied.**
- 6 **Click OK to exit and save your work.**

Registering One or More Application Software Builds

A Project item includes a field for **Current Build Number**. This field designates a number for the build registration utility to use as a starting value when constructing build names. Build Registration creates a build name that usually includes an incrementing build number to identify a specific software build, along with the executable components that define that build.

Software developers cannot submit debug tool sessions from DevPartner tools, such as BoundsChecker and TrueCoverage, to the TrackRecord database without a registered build. The Build Registration

utility registers builds. A configuration administrator must create at least an initial build against which developers can submit debugging information.

Software development groups with multiple developers usually create an automated build process to construct their applications. This automated process could use, for example, MS-DOS batch files to call Make utilities that automate compiler operation and place files in specified locations.

Administrators can make TrackRecord build registration a part of this automated build process. Automating build registration ensures that a build gets registered at the time the build completes.

To Register a Single Build

- 1 From the Windows Start menu, select Programs/Compuware/TrackRecord/Build Registration.**
The Build Registration Utility opens.
- 2 Select a project from the Project drop-down list.**
Click **Edit** at the bottom of the window if you want to modify the default formula for constructing build names.
- 3 Click New.**
The **Submit Build Information for <Projectname>** dialog opens.
- 4 Click Add.**
The **Add Executable Files to <Buildname>** dialog opens and displays directory and file information.
- 5 Choose the EXE, DLL, and OCX files that compose this build.**
These components include the executable components that your developers are building for this project.
- 6 Click Add.**
- 7 Click Submit and then Exit.**

To Automate Build Registration

- 1 Open your development group's build automation script in a file editor.**
- 2 Enter a line to run the command-line version of the build registration utility, BUILDNEW.EXE (use the /? switch to view the options and syntax of this command).**

For example, enter the lines:

```
buildnew /P"Wildfire" /U"Joe Developer,mypasswd"  
c:\Wildfire\Release\*.exe
```

```
buildnew /P"Wildfire" /U"Joe Developer,mypasswd" /A  
c:\Wildfire\Release\*.dll
```

A variation of this syntax uses a file name in place of a list of components:

```
buildnew /P"Wildfire" /U"Joe Developer,mypasswd"  
@wildfire.bld
```

Here, the file `WILDFIRE.BLD` contains the paths, individual files names, or wildcard references to the executable components you want to include in the registered build.

3 Save the file.

Chapter 12

TrackRecord Database Administration



This chapter describes the tasks required to set up and maintain the TrackRecord database.

Note If you are using an existing TrackRecord Version 4 database, contact Compuware Technical Support for migration information.

Overview

During installation, Setup prompts for a database name. If the database does not exist, the TrackRecord installation program creates a database using the supplied name. This database contains sample types, queries and views that can be used to familiarize users with TrackRecord.

Once TrackRecord is installed (as described in the *TrackRecord Installation Guide*) and configured (as described in Chapter 11, “TrackRecord Configuration”), users can begin to generate and accumulate project data. Administrators will then need to protect that data by managing the database.

Management tasks include:

- ◆ Database administration, including backing up, checking and creating new databases
- ◆ Setting up TrackRecord WebServer to allow access to the database through a browser
- ◆ Setting up TrackRecord AutoAlert to allow automatic e-mail notification of database changes

Each of these tasks is described in the following sections.

Database Administration

TrackRecord database administration involves a variety of duties including:

- ◆ Creating new databases
- ◆ Maintaining database reliability
 - ◇ Making backups
 - ◇ Copying a database
 - ◇ Checking the database
 - ◇ Repairing database indices

TrackRecord's Database Administration utility provides the primary tool set for database administration.

New Database

The TrackRecord sample database contains several default types (Defect, Task, Person, and so on) and several default queries and reports. You can use this database as your production database, making modifications as appropriate for your environment, or you may choose to create a new, blank database rather than use the sample. You may also wish to create new databases to support multiple projects.

You can create new databases with either:

- ◆ A copy of the sample database shipped with TrackRecord that provides the sample types, queries, Outline Reports, sample data, and a default workflow
- ◆ A database containing the types from the sample database, but no data, queries, Outline Reports, or default workflow

To Create a New Database

- 1 **From the Start/Programs/Compuware/TrackRecord menu, select the Database Administration Utility.**
- 2 **Log in as a user with administrator privilege.**
The Database Administration Utility dialog opens.
- 3 **Click the New button.**
A New Database dialog opens.
- 4 **Enter a name for the new database.**
- 5 **Choose a database type, either Blank Database or Sample Database.**
- 6 **Click Create.**

Backup and Restore Database

Protection of data requires frequent and regular backups. Database Administration allows you to schedule daily backups at any time you specify, usually during the overnight period when few if any development team members will be working with TrackRecord.

To Schedule Backups of Your Database

- 1 From the Start/Programs/Compuware/TrackRecord menu start the Database Administration Utility.
- 2 Log in as a user with administrator privilege.
The Database Administration Utility dialog opens.
- 3 From the displayed list, select the database to be backed up.
- 4 Click the Backup button.
A Backup Settings for <database> dialog opens.
- 5 Check the box labeled Backup Daily At to schedule a regular daily backup.
- 6 Click the drop-down arrow to select a time from the displayed list.
- 7 Click OK.

To Restore a Backed-up Database

- 1 Halt the TrackRecord (Faircom) server.
- 2 From the Start/Programs/Compuware/TrackRecord menu start the Database Administration Utility.
- 3 Log in as a user with administrator privilege.
The Database Administration Utility dialog opens.
- 4 From the displayed list, select the database to be restored.
- 5 Click the Restore button.
- 6 Restart the TrackRecord (Faircom) server.

Copy Database

You can copy an existing database to a new location and change its name. This operation lets you make minor modifications to an existing database without having to start over from scratch.

To Make a Copy of a Database

- 1 Halt the TrackRecord server.**

See the *TrackRecord Installation Guide* for instructions for manually stopping the server. For Windows NT machines running TrackRecord as a service, use the Control Panel/Services option to stop the Faircom Server.
- 2 From the Start/Programs/Compuware/TrackRecord menu, open the Database Administration Utility.**
- 3 Log in as a user with administrator privilege.**

The Database Administration Utility dialog opens.
- 4 Select a database from the displayed list.**
- 5 Click the Copy button.**

The Copy button will not be active if the TrackRecord server is running.
- 6 Enter a name for the database to be created.**
- 7 Click Copy.**

Check Database

Software developers sometimes experience computer crashes or network problems, such as the failure of a server acting as a router. These failures may occur while data is being submitted to TrackRecord, which may leave the database with incomplete entries. Check Database will recover as much data as possible and place items in a workable state.

On a routine basis, check a database monthly.

To Check Your Database for Errors

If you are running TrackRecord and are logged in to the database to be checked, you can use the Administrator menu's **Check Database** option to perform this task. To check other databases:

- 1 With the server running, from the Start/Programs/Compuware/TrackRecord menu start the Database Administration Utility.**
- 2 Log in as a user with administrator privilege.**

The Database Administration Utility dialog opens.
- 3 Select a database from the displayed list.**
- 4 Click the Check button.**

- 5 **If users are logged on using this database, a dialog will be displayed listing current users.**

You cannot check a database while users are using the database. Refer to “Database Lock-out” on page 159 for information about this dialog. When all users have logged off, this dialog is dismissed and the **Check <dbname> Database** dialog opens.

- 6 **Click Check to fix broken internal links or Cancel to exit without checking the database.**

If you are not currently logged in to the database, you will be prompted to log in.

Rebuild Database

Only a catastrophic failure, such as a power failure, could require a database index rebuild.

To Rebuild Your Database Index

- 1 **Halt the TrackRecord server.**

See the *TrackRecord Installation Guide* for instructions for manually stopping the server. For Windows NT machines running TrackRecord as a service, use the Control Panel/Services option to stop the Faircom Server.

- 2 **From the Start/Programs/Compuware/TrackRecord menu, start the Database Administration Utility.**

- 3 **Log in as a user with administrator privilege.**

The Database Administration Utility dialog opens.

- 4 **From the displayed list, select the database to be rebuilt.**

- 5 **Click the Rebuild button.**

The Rebuild button will be inactive if the TrackRecord server is running.

- 6 **If you want to defer reclaiming file space, click No Compact Files.**

Compacting files reclaims file space created when you delete database data. Over time, this unclaimed space might degrade the performance of a database slightly.

- 7 **Click Rebuild.**

Administering WebServer

WebServer allows users to access a TrackRecord database using a Web browser such as Microsoft Internet Explorer or Netscape Navigator/Communicator. The WebServer user interface is described in Chapter 8, “Using TrackRecord WebServer.” The following sections describe features available only to users with Administrator privilege.

All users authorized to use the Windows TrackRecord client are also authorized to use the WebServer client. See “Creating Team Members” on page 138 to create a new Web user who is not already set up in TrackRecord.

The following sections contain information that will assist you in interpreting the statistics available through the WebServer Administrator’s menu. A thorough understanding of this information is *not* required to successfully administer WebServer. This information can help you assess system performance and requirements.

WebServer Processes

When a user logs in to WebServer, a session manager process – TRWebSessionsHost – is automatically created. TRWebSessionsHost assigns a session identifier to each logged in client and assigns a license to that session.

When the user makes a request to the database, WebServer creates a Database Host (TRWebDBHost) process or adds the user’s session to an existing DBHost process. A DBHost process caches values for a specific database. When a DBHost reaches its threshold for the number of sessions it can handle, the WebServer creates another DBHost.

- ◆ Each DBHost services requests for only one database.
- ◆ There can be multiple DBHosts for each database.
- ◆ Multiple sessions can be served by each DBHost.

As users log off, their sessions are terminated and their licenses become available for use by other users. When there are no more user requests, the TRWebSessionsHost and TRWebDBHost processes will be shut down automatically.

Permanent and Transient Sessions

The session identifier assigned by the TRWebSessionsHost process is kept for the entire life of the client session. This is the user’s permanent

session. It identifies this user's requests until the user logs off to terminate the session.

When the user makes a first request, WebServer creates a transient session. Transient sessions are data structures that cache data from users for a short period of time. They are created for efficiency. If the transient session is unused for ten minutes (i.e., there have been no requests from this client), the transient session times out and is removed. If the client subsequently makes a request, a transient session is recreated from the information in the permanent session.

Monitoring permanent sessions and transient sessions separately allows you to accurately assess the requirements of your server and users.

WebServer Log Files

The Web server (IIS, Netscape, O'Reilly Website) on which you are running TrackRecord WebServer creates a log file. Analysis of this log file can help you ensure that your system is sized properly.

Since all WebServer requests are logged as CGI requests, be sure to enable CGI request logging in your log file. (Logging of HTML page and GIF file requests are generally enabled; CGI requests may not be enabled by default.)

Ending a Session

There are several ways to end a WebServer session:

- ◆ The user clicks the Logout link
- ◆ The user closes the browser
- ◆ The user accesses a different URL or clicks the Home button

In these cases, the session is terminated and eventually all system processes are stopped.

To ensure that sessions are terminated and processes stopped in situations such as a loss of power in the client or server machine, WebServer maintains a license beat between the WebServer application client and the WebServer. If the license beat is interrupted, sessions are terminated and system processes are stopped.

- ◆ If the client's attempt to send a license beat fails four times in a row, the client will stop sending license beats.
- ◆ If a license beat response contains an invalid session ID error, the client will stop sending license beats. (This error could be

generated, for example, if the NT server running the TrackRecord WebServer has crashed.)

- ◆ If the server has not received a request (minimally, a license beat) from a client for an extended time, the server will terminate the session.

If a user attempts to access the database after their session is terminated, they will receive a message that the application has terminated and be prompted to log in again.

WebServer Administration Menu

Using the WebServer interface, the WebServer administrator can monitor server performance and perform other administrative functions.

To Perform Administrative Functions

- 1 Open a Web browser, such as Microsoft Internet Explorer or Netscape Navigator.**

- 2 Enter the following address:**

`http://www.servername/TrackRecord/Welcome.html`

where *servername* is the name of the server on which WebServer is installed.

The TrackRecord WebServer Login page opens.

- 3 Enter the name of your TrackRecord database, your user name and your password.**

Be sure to log in as a user with Administrator privilege.

- 4 Click OK.**

The TrackRecord WebServer main window opens.

Refer to Chapter 8, “Using TrackRecord WebServer,” for information about using this interface.

- 5 To perform administrative functions, click the **Manage** link in the top frame of the window.

The Administration Menu is displayed, as shown in Figure 12-1.

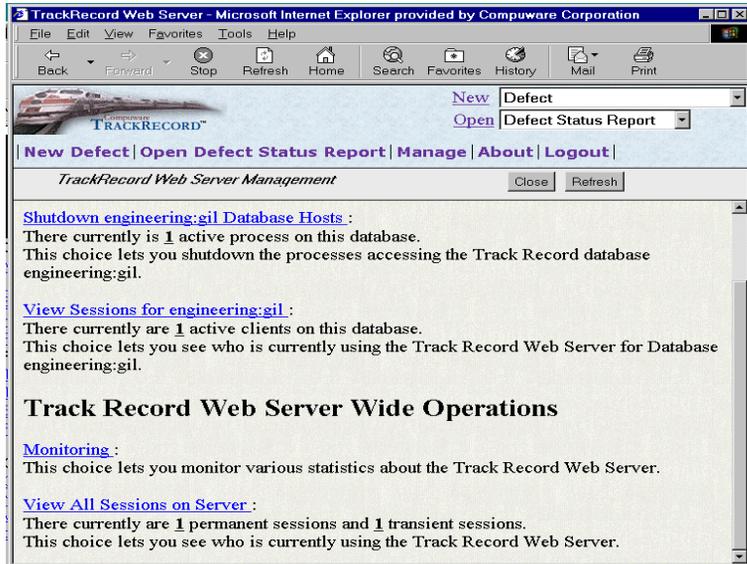


Figure 12-1 WebServer Administrative Options

The first two options allow you to shut down or view session information for the database named in your login screen. The second two options allow you to view information for all WebServer users, regardless of the database being used.

Note In addition to the options described in this section, the Web Monitor utility can be used to shut down all WebServer processes in the event of a catastrophic failure. Refer to “WebMonitor” on page 189 for information about the Web Monitor utility.

The options on this screen are:

- ◆ **Shutdown *dbname* Database Hosts** - This option allows you to shut down the Database Hosts for the named database. (Refer to “WebServer Processes” on page 184 for a description of database hosts.) Clicking this option prompts you to confirm that you want to shut down all hosts. Database Hosts will be recreated when a request, such as a license beat, is received. User sessions will not be terminated.

- ◆ **View Sessions for *dbname*** - This option allows you to view a list of the users currently accessing this database through the TrackRecord WebServer (Figure 12-2).

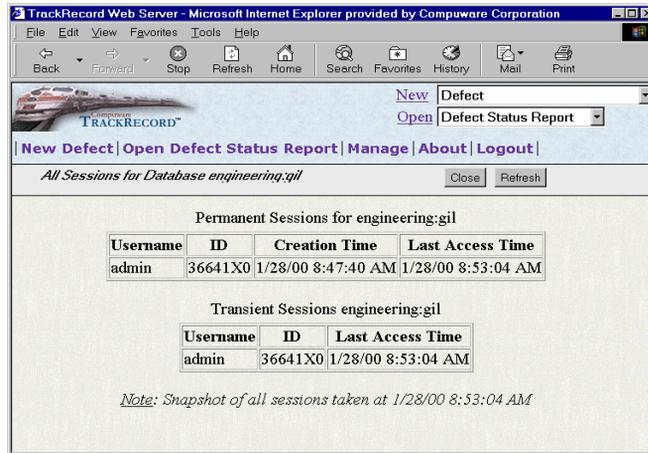


Figure 12-2 View Sessions

All permanent and transient sessions that are accessing this database are listed, as well as the time of the last request from each session to the WebServer. Click the **Refresh** button to update the information displayed. Click **Close** to return to the WebServer Administrator menu.

- ◆ **Monitoring** - This option allows you to monitor permanent and transient sessions, database hosts, and the requests being issued to the WebServer. Clicking this option displays a monitor window, as shown in Figure 12-3.

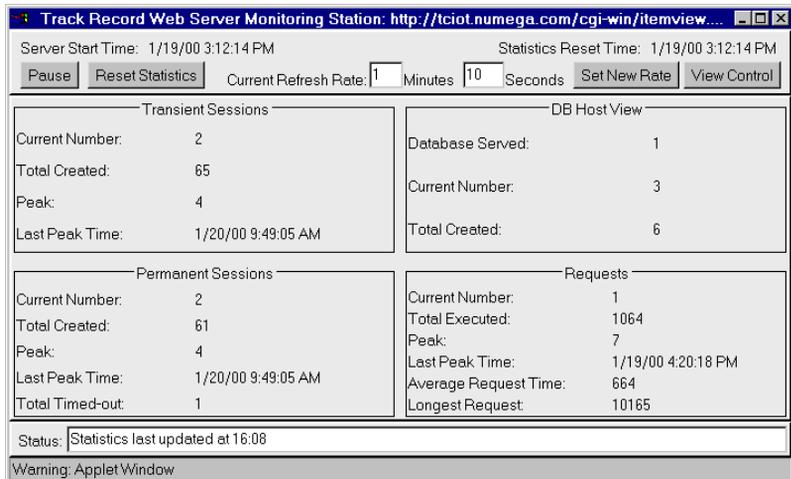


Figure 12-3 Monitor Window

- ◇ Click the **Pause** button to prevent the Monitor window from being refreshed. The button text changes to **Resume**. Click the **Resume** button to resume monitoring. Since every update to the monitor is a request, constantly updating the monitor can load the server unnecessarily.
- ◇ Click the **Reset Statistics** button to reset values to zero. This allows you to view an accurate representation of the activity occurring at the current time.
- ◇ To change the rate at which the monitor window is refreshed, change the time displayed and click the **Set New Rate** button.
- ◇ By default, statistics about Permanent Sessions, Transient Sessions, DBHosts and Requests are displayed. To remove one or more of these from the Monitor display, click the **View Control** button and uncheck the items to be removed.
- ◆ **View All Sessions on Server** - This option allows you to see who is currently using the TrackRecord Web Server, regardless of the database they are accessing. Permanent and transient sessions for all databases are listed, as well as the time of the last request from each session to the WebServer.

WebMonitor

WebServer contains a process (TRWebSessionsHost) that maintains all currently active user sessions. This process contains a monitoring component called the Web Monitor. Assuming that the installer has set the appropriate DCO options (as described in the *TrackRecord Installation Guide*), this monitoring component is visible on the system on which the WebServer is installed. You cannot display it on all Administrator systems.

The Web Monitor is equivalent to the View All Sessions option on the WebServer Administration dialog, but allows two additional areas of functionality: it allows the administrator to view the internals of the database, and it contains a **Shutdown Now** button. **Shutdown Now** shuts down the TRWebSessionsHost process, which will invalidate all user sessions.

Troubleshooting Common WebServer Issues

TrackRecord Web client diagnostic information is displayed in the Java Console. If you are encountering problems, enable the Java Console to assist in diagnosing the problem.

In the Netscape browser, the Java Console is automatically enabled. In the Internet Explorer browser, do the following to enable the Java Console.

To Enable the Java Console

- 1 From Internet Explorer's Tools menu, select Internet Options.
- 2 From the Advanced tab, scroll to display the Java Virtual Machine group and check the Java Console Enabled option.

To View the Java Console Diagnostics

- ◆ In Internet Explorer, from the View menu select Java Console.
- ◆ In Netscape, from the Communicator menu select Tools, then Java Console.

WebServer requires that Java and Javascript be enabled in the client's browser. For Netscape users, from the Edit menu select Preferences. Click the Advanced group and check the Enable Java and Enable Javascript options.

For the most current TrackRecord support information, please visit the Support section of NuMega's Web site at:

<http://www.compuware.com/numega>

From this location, you will have access to our online KnowledgeBase and the most current patches and updates for the product.

If the TrackRecord WebServer has been installed and you are getting an error message saying that the site "http://..." cannot be opened, or you are having difficulty accessing the TrackRecord WebServer login page or are unable to log in to the database, please check the following:

- 1 Be sure that you can run the TrackRecord Windows client on your local machine and access the database, without using the WebServer. If the database cannot be accessed from this machine using the TrackRecord Windows client (TR.EXE), then the TrackRecord WebServer will also not be able to access the database.
- 2 Make sure the World Wide Web Publishing services is set to interact with the desktop. This can be verified by going to the Start menu, then Settings, Control Panel, Services, World Wide Web Publishing Service, Properties, Log On As: Allow Service to Interact with Desktop.
- 3 Make sure the DCOMCNFG settings are correct:
 - a From the Start menu, click Run, and type in "DCOMCNFG".

- b From the Applications tab, select TRWebDB.TRWebDBHost.
 - c Click on the **Properties** button.
 - d Go to the **Identity** tab and be sure the user account to log in to the database is set to "The interactive user."
Also be sure that a user with Administrator privileges to the database is logged in to NT while the WebServer is running, since Administrator privilege might allow the WebServer to work where the Launching User's permissions will not.
 - e Repeat Steps b through d for TRWebSessionsHost.UserInfo.
- 4 Check that the URL is correct. Your URL has to be:
`http://<servername>/TrackRecord/Welcome.html`
 - 5 Make sure that you do not have an ampersand (&), a quote sign, a greater than sign (>), or a less than sign (<) in any of your TrackRecord fields that are going to be displayed on the Web. If you do, you will not be able to open up the Web page.

Other questions that might arise are:

- ◆ Calendar menu that is available when using the TrackRecord Windows client is missing
This option is not available in the Web server. If you click on the button, it will not do anything. The date must be entered manually in the following format: mm/dd/yy.
- ◆ Items are not displayed
Only items shown in the **Display to All Users** list will show up on the WebServer. The types that appear in the list can be controlled with the Display to All Users type property, which you can set in the Type Editor in TrackRecord.
- ◆ When using TrackRecord WebServer, always enter date information using the same date format as that being used by the computer running the WebServer. For example, if you configured the WebServer machine to use dates in the format dd/mm/yyyy, you should enter a date 28/10/1999, not 10/28/1999.
If you want to enter dates using different formats, you can run multiple instances of WebServer, each on a different computer configured for a different date format. Connect each of these WebServers to the same TrackRecord database.
Note that alphabetic characters are not supported in dates.

Configuring AutoAlert

The TrackRecord AutoAlert utility sends e-mail messages to users to notify them of changes in the TrackRecord database. AutoAlert lets you define flexible criteria for notifying each user of changes that might be of interest to them.

To receive automatic mail notification, the user sets up the notification criteria by creating TrackRecord queries with the **Email Me When New Items Match This Query** option enabled. AutoAlert monitors the TrackRecord database, periodically checking to see if any new items match these queries.

Whenever a new or changed item matches one of the queries, TrackRecord sends an e-mail message to the owner of the query. By using TrackRecord's flexible query engine, AutoAlert makes it possible for users to receive mail notification based on almost any criteria they require.

Configuring AutoAlert consists of the following tasks:

- ◆ Setting up AutoAlert users
- ◆ Setting up queries
- ◆ Using the AutoAlert Administrator to set up databases and options
- ◆ Optionally, changing field properties to specify which fields should be included in mail messages

Each of these tasks is described in the following sections.

Installation Notes

The *TrackRecord Installation Guide* contains instructions for installing the AutoAlert Server and the AutoAlert Administrator. The following notes are included in those instructions and are repeated here for your convenience.

After installation, you must create a mail profile for AutoAlert's use.

- 1 On the machine on which the AutoAlert Server is installed, log on to an account that has privileges in the domain in which the service will run.**
- 2 From Start/Settings/Control Panel select the Mail option.**
- 3 On the Services tab click the Show Profiles button.**
 - If a profile named AutoAlert is listed, skip to Step 4.
 - If a profile named AutoAlert is not listed:
 - a Click the **Add** button to start the Profile Wizard.

- b Check the appropriate **Information Service**, typically Microsoft Exchange Server.
 - c For the **Profile** name, enter the name AutoAlert.
 - d Enter the name of your **Exchange server** and a **mailbox** name. You should set up a mailbox for AutoAlert's exclusive use. AutoAlert e-mail notices are sent from this mailbox.
 - e Accept the defaults for the remaining Wizard prompts and click **OK**.
- 4 **Run Outlook using the profile you created in Step 3 and attempt to send a message. If you are not successful, contact your Exchange administrator for assistance before continuing.**
 - 5 **From the Start/Settings/Control Panel/Services option, select AutoAlert Service and click the Startup button.**
 - a In the Startup Type box, check **Automatic**. This step is optional but recommended.
 - b In the Log On As: box, check the **This Account** option and enter information for the account you logged in to in Step 1.
 - c Click **OK** to close the Startup dialog.
 - d With the AutoAlert Service selected, click the **Start** button.
The AutoAlert Service may take up to a minute to start when it is first started.

Setting Up AutoAlert Users

For users to receive e-mail notification, their e-mail address must be included in their TrackRecord login information. If an e-mail address was not entered when the user's Team Member item was created, do the following:

- 1 **Start TrackRecord and log on as a user with administrator privileges.**
- 2 **From the Toolbar's Item Browser button, select Team Member.**

If the user is listed in the item list, click on the user's name and proceed to Step 3.

If the user is not listed, click the **New** button in the **Item Browser's** toolbar. Refer to "Creating Team Members" on page 138 for information about creating Team Members.

3 Click on the TrackRecord User tab.

If the fields on this tab are active, skip to Step 4.

If the fields on this tab are inactive, the Team Member is not an authorized TrackRecord user. Use the **TrackRecord User** control in the toolbar to select this user's status.

Note If your Team Member Item window is not wide enough, the TrackRecord User control will not be visible on your toolbar. Resize the window until the control is displayed.

The user status options are:

- ◇ Normal – Active with no administrative authority
- ◇ Administrator – Active with administrative authority
- ◇ Inactive – Used only to retain user information in the database while locking out that username and password for TrackRecord logins

4 For TrackRecord users (Normal and Administrator), enter this user's login name in the name field.

Optionally, enter and confirm a password. To allow users to log in without a password or to change their password the first time they log in, you may leave the Password field blank.

5 Enter an e-mail address for this user.

An e-mail address is required if the user will be using the AutoAlert utility.

6 Enter a Version Control System user name and password.

This step is required if the username is different from the name just entered and if you intend to integrate with a version control system.

7 Click OK, then click Close.

Setting Up Mail Queries

A mail query is a TrackRecord query in which the **Email Me When New Items Match This Query** option has been checked. Each AutoAlert recipient must construct a set of queries that describe the criteria for e-mail notification. For example, most users will probably want a mail query that selects any items assigned to them. The query editor provides an Options dialog with a setting for sending out e-mail whenever a new or changed item matches the query.

Mail queries can be shared with other users to be used as a normal query, but AutoAlert will send e-mail notification only to the owner of

the query. The one exception is mail queries created by the Admin user. If a user with the username Admin creates a query, checks the **Email Me** option, and also checks the **Share to All** option, all TrackRecord users will receive e-mail when an item matches the query criteria. If the query includes the “current user” criterion, the e-mail will be specific to each TrackRecord user. (Note that this feature is available only to the user with the username “Admin”, not to any other user with administrator privilege.)

Users can set up their own mail queries or you can set up queries for users.

To Set Up Mail Queries for Individual Users

- 1 **Start TrackRecord and log in as the user you are configuring (look on the right side of the status line to see the name of the current user).**

To log on as a specific user, close all open windows, and from the **Tools** menu select **Change Login**.

- 2 **Create a query that selects items that this user would be interested in seeing.**
- 3 **On the Query creation dialog, click Options and check the Email Me When New Items Match This Query option.**
- 4 **Repeat Steps 2 and 3 for any other queries that you want to create.**

For example, most users will probably want a mail query that selects any items assigned to them.

- 5 **Repeat this process for each AutoAlert recipient.**

Using the AutoAlert Administrator

The AutoAlert Administrator allows you to add and remove databases to be polled by the AutoAlert service, show the users that are AutoAlert enabled, and set a variety of options.

To Run the AutoAlert Administrator

- 1 **On a machine on which the AutoAlert Administrator has been installed, from the Start/Programs/Compuware/TrackRecord menu select AutoAlert.**

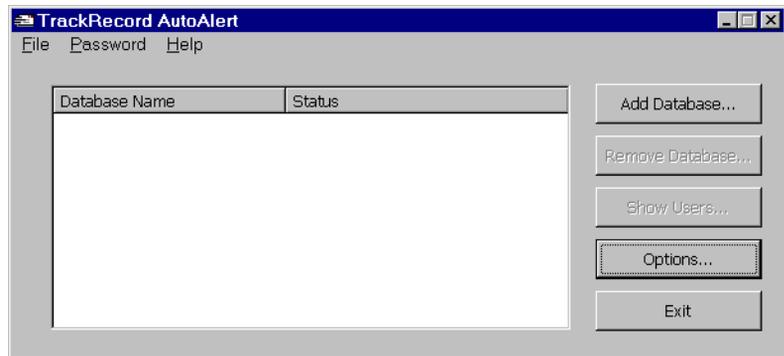
- 2 **If the AutoAlert service is not installed and running on this machine, you will be prompted for the name of the machine on which the service resides.**

If the service is on this machine and is not set to start automatically, start the service from the Start/Settings/Control Panel/Services option. Select **AutoAlert Service** and click the **Start** button.

- 3 **The first time the AutoAlert Administrator is run, you will be prompted to elect to use password protection.**

If you choose not to enter a password at this time, you can elect to do so at a later time through the AutoAlert Administrator's Password menu.

The **AutoAlert Administrator** dialog opens. The options on this dialog are described in the following sections.



Adding and Removing Databases

You must identify each database that AutoAlert is to poll for items matching users' queries.

To Add a Database

- 1 **Click the Add Database button.**
- 2 **Enter a username, password and database name and click OK.**
The database name is listed.
- 3 **To enable the database for AutoAlert polling, click its checkbox.**

To remove a database from AutoAlert's polling loop, uncheck it. To remove the database from the list, select the database name and click the **Remove Database** button.

Show Users

Clicking the **Show Users** button allows you to view a list of mail queries for e-mail enabled users on the selected database. To see the properties for a user, select a username from the **User name** drop-down list.

Note that any changes to the **User name** list, or to the properties of any mail enabled user, must be made in TrackRecord. No user information can be changed from the AutoAlert interface. Once a status change is made in TrackRecord, AutoAlert automatically updates the User list.

Options

Click the Options button to set the following:



- ◆ **Automatically save state information every** - Allows you to set the autosave frequency. Saving more frequently uses more system resources, but allows for better recovery in the event of a system crash. Note that AutoAlert automatically saves its state when it is shut down normally. The state information is saved in a file named **USERS.LOG** in the AutoAlert directory. Under normal circumstances, you should not delete this file.
- ◆ **Mail polling frequency** - Lets you specify how frequently AutoAlert checks for new or changed items. A lower setting will use more system resources, because AutoAlert will poll more frequently and therefore use more CPU time. A higher setting will use fewer resources, but will result in longer delays before mail notification is sent. For larger groups (20 or more mail users), use a polling frequency of 30 minutes or higher. Note that TrackRecord and your

mail package may also have polling frequencies that could result in additional delays before messages are sent by AutoAlert.

- ◆ **Send a daily test message to** - Sends a daily status message to a specified mail recipient (normally the administrator). If a message is not received, the administrator should investigate problems on the mail server. The message contains all information about current mail users and a copy of the mail log file (if one is being generated). You should enable this option until you are certain that AutoAlert is configured properly.
- ◆ **Keep a log file of all AutoAlert activity** - Keeps a detailed log file on all actions AutoAlert performs. The log is stored in a file named MAIL.LOG in the AutoAlert directory. Keep the mail log file when you start using AutoAlert, until it is clear that all users are receiving mail correctly. You can view the mail log by selecting the **View Mail log** option from the AutoAlert Administrator **File** menu.
- ◆ **Message consolidation threshold** - Lets you control how AutoAlert groups messages. When multiple notification items from one mail query need to be sent at one time, AutoAlert can send them separately or as a single mail message. A lower setting means you will receive fewer mail messages, but some messages will have multiple item notifications in them. A higher setting makes it more likely that each notification will result in a separate mail message.

Changing the Contents of Mail Messages

The message sent by AutoAlert consists of the default abbreviation for the type of item that triggered the notification and any fields from that item that are marked with the **Include in Mail** property. You can change the fields included in AutoAlert messages.

To Change the Fields in Your Mail Messages

- 1 **Shut down the AutoAlert service.**
- 2 **Log on to TrackRecord as an Administrator.**
- 3 **Close all TrackRecord items and views (Defects, reports, and so on).**
- 4 **From the TrackRecord Administrator menu, click Edit Types.**

A warning is displayed advising you to backup your database before modifying types. Refer to “Backup and Restore Database” on page 181 if you would like to backup your database at this time.

If users are logged on using this database, a dialog will be displayed listing current users. You cannot edit types while users are using the

database. Refer to “Database Lock-out” on page 159 for information about this dialog. Once all users have logged off, this dialog is dismissed and the **Choose a Type** dialog opens.

- 5 **Open the type you want to change by double-clicking its name.**
- 6 **Double click on the field you want included in mail messages to display the Field Properties dialog.**
- 7 **On the General tab, check the Include in Mail checkbox to have this field included in mail messages sent for this type.**
- 8 **Click OK to close the Field Properties dialog, then OK to close the Type Editor, then Close to close the Choose a Type dialog.**
- 9 **Restart the AutoAlert service.**

Importing and Exporting Data

TrackRecord allows you to transfer data between TrackRecord databases, import data from any delimited ASCII file, or export data to a delimited ASCII file or a Microsoft Access database. For example, you might import an address list to avoid retyping it, export a developer’s list of notes or defect reports, or import an appointment book’s contents.

TrackRecord provides two methods for importing and three methods for exporting data:

- ◆ The TrackRecord **Import-Export Wizard**. This wizard steps you through the process of importing, exporting, and creating a template.
- ◆ The **Import** and **Export** options on the Administrator menu. Note that you must create a template file before using these options.
- ◆ For exporting only, you can use the **Copy** option to transfer data to the clipboard. The data can then be pasted into another program.

Each of these options is described in the following sections.

Using the Import-Export Wizard to Export

You can use the Import-Export Wizard to export data from a TrackRecord database into a delimited ASCII file or into a Microsoft Access database. This data can then be imported into another TrackRecord database or into another program, such as Microsoft Excel or Microsoft Word.

To export data from a TrackRecord database:

- 1 Optionally, log in to TrackRecord, being sure to specify the database from which you intend to export data, and create a query that would find the information you wish to export.**

If you will be exporting all items of a particular type, you do not have to create a query and can proceed directly to step 2.

- 2 From the Start/Programs/Compuware/TrackRecord menu, select the Import-Export Wizard.**
- 3 Log in.**

Be sure to log in to the database from which you want to export data.

The Import-Export Wizard opens.

- 4 Select Export and click Next.**
- 5 Check an option to create either a file or an Access database and enter a name for the file to be created. Click Next.**
- 6 Choose whether to export all items of a certain type or the results of a Query.**

Use the drop-down lists to select the type or Query. Click Next.

- 7 If you have already created a template, enter its name.**

A template file specifies the type and fields to be exported. If you do not yet have a template file:

- ◇ Click the **New Template** button. You will be prompted through the process of creating a template.
- ◇ Select the type and fields to be exported. When selecting fields, note that the **Add** button is not enabled for Parent fields. You must add each child field individually.
- ◇ Choose a delimiter character, being careful to choose a character that is not contained in the data. The pipe (|) or tilde (~) characters are often good choices.
- ◇ If you want to save the template file, enter a file name.

- 8 A message will display the number of records exported. Click OK.**

You can choose to run the wizard again or close the wizard.

If you created a delimited ASCII file, it can be imported into another TrackRecord database or into any program capable of importing delimited ASCII data. If you created an Access database, it can be opened with Access.

Exporting Using the Command Line Interface

You can export TrackRecord data to an Access database through a command line interface rather than through the Import-Export wizard.

Using the command line interface allows you to automatically create an Access table on a periodic basis, for example, using Microsoft's Task Scheduler.

The command is `TRTOMDB.EXE`. The following parameters are supported:

-d TrackRecord database to use

-u Username

-p Password

-t Full path to the template file

-a Full path of the Access table to be created

-q Optional. Name of the query in this database that is to be used. If not specified, all items defined in the template will be exported.

-o Optional. When set to `TRUE`, it will suppress the MDB overwrite dialog. The default is `False`.

All parameters must be enclosed in double quotes. The following is an example of a properly formatted command line:

```
TRTOMDB.EXE -d"Default" -u"Admin" -p"Admin"  
-t"c:\template.tpl" -a"c:\access.mdb" -q"Bugs" -o"True"
```

Using the Import-Export Wizard to Import

To import data into a TrackRecord database, you must have previously created a delimited ASCII file. The ASCII file can be created by the Import-Export wizard, with the Export option on the Administrator's menu, or with any program capable of exporting to a delimited ASCII file, such as Microsoft Excel.

When importing data, you can potentially corrupt your database. It is advisable to make a copy of your database before attempting to import data.

If you are importing data from an ASCII file that was not created by the Import-Export wizard, you should verify that the data is in the correct format by exporting data from your existing database and comparing the file created with the file you intend to import. If the formats do not match, you can alter the data to be imported before attempting the

import. Importing data that was exported with the Import-Export wizard is less error-prone.

Note that the format of date fields must correspond to the currently configured windows short date format.

To Import Using the Import-Export Wizard

- 1 **From the Start/Programs/Compuware/TrackRecord menu, select the Import-Export Wizard.**

Be sure to log in to the database into which you want to import data.

- 2 **Select Import.**

- 3 **Enter the name of the file containing your exported data.**

- 4 **If you have already created a template file, enter its filename.**

A template file specifies the type and fields to be imported. If you do not yet have a template file, you will be prompted through the process of creating one. Select the type and fields to be imported and choose the delimiter.

- 5 **Indicate what action should be taken if duplicate data is found.**

Duplicate imported data can be discarded or existing data can be overwritten. Duplicates are determined by comparing the field whose abbreviation is Dupe. (This abbreviation was established when the type was created, as described in “Identifying Duplicate Imported Items” on page 158.)

The data is considered a duplicate if it matches existing data of the type being added or of a parent type. Note that the data is not considered a duplicate if it matches a child of the type being added.

A message will display the number of records imported and the wizard will be closed.

Using the Import and Export Menu Options

The Administrator’s menu contains **Import** and **Export** options. These options allow you to import and export data without using the Import-Export wizard.

A template file is required for both importing and exporting. You can use the Import-Export Wizard to create the template file by selecting the Template Only option. Refer to the previous sections for information about using the Import-Export Wizard to create a template file. (Alternatively, you can create a template file manually, as described in “Manually Creating Template Files and Importing” on page 203.)

You can choose to export data (the entire database or the results of a query) or type descriptions.

You can import data, type descriptions, or an outline created with a previous version of TrackRecord. When importing an Outline, it is saved as an Outline Report accessible through the current version of TrackRecord.

Using Copy and Paste to Export Data

If the information you want to export is displayed in an Outline view, the easiest way to transfer the information is to use Copy and Paste.

Set up the report to display the columns of information you wish to transfer, select the header you want to copy, and use the **Edit** menu's **Copy** option to copy the header. The displayed data will be transferred to the clipboard and can be pasted into Microsoft Excel, Word, Adobe FrameMaker or other applications.

Manually Creating Template Files and Importing

To import from or export to an ASCII file, you must first create a template file, which describes the mappings of the fields in your data files to the equivalent TrackRecord fields. You can use the Import-Export wizard to create a template, as described in previous sections, or you can manually create a template file.

To Set up a Template File and Import Information

- 1 Determine which TrackRecord types correspond to the type of information you want to import.**

Each ASCII file can only contain items of a single type, but any subfield of a particular item can also be included in the import file.

For example, you could import Tasks in a single ASCII file, and since Tasks can include an "Assigned to" field, you could also import People to fill that field in the Tasks. The People would be imported as an integral part of each Task in your ASCII file, but would be created as separate objects contained within the Tasks in TrackRecord (as all Tasks are normally created in TrackRecord).

- 2 Determine the delimiter character used to separate fields in your ASCII file.**

Double quote characters can be used in import files at the beginning and end of fields, but another delimiter character (often a comma is used) must also be used to delimit the fields.

- 3 Edit a file called <name>.tpl, where <name> is any name you choose to assign to the template.**
- 4 Insert the following text into the file:**

```
[Template]
TypeName=<TrackRecord type name>
DelimiterChar=<delimiter char>
```

where <TrackRecord type name> is the name of the TrackRecord type for the information you are importing (from step 1), and <delimiter char> is the delimiter character (from step 2). For example, you might enter:

```
[Template]
TypeName=Defect
DelimiterChar=$
```

To use a newline character as the delimiter, specify
DelimiterChar=\n.

- 5 List the names of the TrackRecord fields in the type specified that correspond to the types in your input file.**

There are two types of field types: Basic Types and Compound Types. Basic Types are simple fields for common types, like names, addresses, strings, numerics, etc. All other types are Compound Types.

For Basic Type fields, there may actually be more than one corresponding “field” in the import or export file. In the sample database, there are two fields that contain more than one field: Name and Address.

- ◇ For Names, there will be three fields (Last, First, Middle)
- ◇ For Addresses, there will be six fields (Line 1, Line 2, City, State, Zip, Country)

If a field type is a Compound Type, you must also specify how many subfields of the Compound Type are in the input file, by specifying the number after a comma on the FieldName line. For Basic Types consisting of multiple fields, each of these fields is considered one field, and the number of fields specified after the name of the Compound Field should reflect that, as shown in the Reported By field in the following example.

For example, to import a Defect containing a Description, a Project, and a Priority, you would use the following lines (without the comments):

```
FieldName=Description; The description of the defect
FieldName=Project,1 ; The Project, which has one
subfield.
; (there is no actual field in the
```

```

; input file corresponding to this
; line; it just affects the
; interpretation of the next line)
FieldName=Description ; The Description field from
; the Project (the one subfield)
FieldName=Priority ; The Priority of the Bug Report

```

To ignore a field that exists in the input file, use a `FieldName=` specifier, but don't specify a field name.

To import Check Box and Radio Button items, your ASCII (import) file should contain either 0 or a blank space for unchecked items, and 1 or an "X" for checked items.

So our sample would look like this:

```

[Template]
TypeName=Defect
DelimiterChar=$
FieldName=Description
FieldName=Project,1
FieldName=Description
FieldName=Priority
FieldName=Fixed

```

A more complex example, which would also export the people in the Reported by and Fixed by fields (and their names, companies, and addresses) would be:

```

[Template]
TypeName=Defect
DelimiterChar=$
FieldName=Description
FieldName=Priority
FieldName=Project,1
FieldName=Description
FieldName=Reported by,3; 3 fields are specified:
FieldName=Name; Name, Address, and Company
FieldName=Company,1; For the Company, only name is used.
FieldName=Name
FieldName=Address; This is the third field in Reported by
FieldName=Fixed by,3
FieldName=Name
FieldName=Company,1
FieldName=Name
FieldName=Address

```

- 6 Make sure your import file is formatted to correspond to the template file, with each field separated from the next field by the delimiter character (except for fields at the ends of records, which use the newline character as a delimiter).**

The following is a valid input file for the template described earlier:

```
Generic bug 1$TrackRecord$2  
Generic bug 2$TrackRecord$1
```

Each record should be on a line by itself, with the newline character representing the end of the record. The only exception is that newline characters are allowed within double-quoted fields.

See “Import File Specifications” on page 206 for information about specifying Name and Address fields.

Fields corresponding to TrackRecord lists must be terminated with an extra delimiter character after the last item in the list.

Note that the format of date fields must correspond to the currently configured windows short date format.

- 7 **Back up the contents of the TrackRecord object database subdirectory, in case you want to “undo” the import if there is a mistake in the template or data files.**
- 8 **To verify that your data is in the correct format, export data from your existing database and compare the file created with the file to be imported.**

If the formats do not match, alter the import file.

- 9 **Start TrackRecord, and choose Administrator, then select Import, and choose Database and the template and input files, then press OK.**

If there are any errors, you will see error messages.

Note Importing from ASCII files can be quite slow in many situations. The more items there are in your object database, the slower importing may be because TrackRecord must check for duplicate data. Importing into an empty object database will always be faster than importing into a large object database.

After a large import, clicking on a type in the Item Browser may result in a long pause as TrackRecord updates its indices. The more items you import, the longer this pause will be. This process can take several minutes to complete.

Import File Specifications

The following Basic Types must have all subfields appear in the following order in an import file, with each subfield separated by the delimiter character:

```
NameLast Name, First Name, Middle Name  
AddressAddress Line 1, Address Line 2, City, State,  
Zip, Country
```

For example, the following template could be used to import information about a person:

```
[Template]
TypeName=Person
DelimiterChar=,
FieldName=Name
FieldName=Address
FieldName=Phone Numbers,1
FieldName=Phone
```

Because Name and Address are Basic Type fields, you should only specify the name of the field (as above), not the name of all the subfields that make up the name or address. If in doubt about field names, look at an item in the Item view for the particular type that you are importing; the name to use is displayed near the field itself.

For this template file, one valid input file might be:

```
Doe,John,J,123 Any
Street,,Anytown,DC,12345,USA,2024567890,,
Doe,Jane,,456 Any
Street,,Anytown,DC,12345,USA,2024567890,,
```

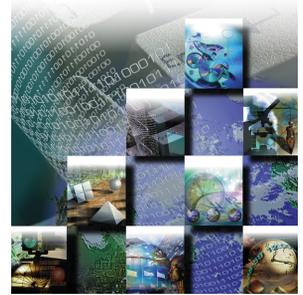
The extra comma after each of the phone numbers indicates the end of the list.

Problems with Input Files

If you are having problems with your input file and you are not sure if it matches your template, try using the same template to export some existing items to an ASCII file. After exporting the items, look at the file TrackRecord created. Your input file should follow much the same format.

Appendix A

ActiveX Programming Interface



This appendix provides reference material for programmers writing modules to access the TrackRecord databases.

ActiveX Interfaces

TrackRecord's ActiveX (formerly OLE) Automation interface allows any program with an ActiveX or OLE-compliant extension language to access a database, run queries, extract items, and perform other functions.

Using this information, you can write programs that can perform many useful functions, including:

- ◆ Automatic notification of new bugs
- ◆ Mail notification based on bug assignment
- ◆ Reports and calculations based on *live* TrackRecord data *without* exporting
- ◆ Automatic exporting of daily data, and transfer of the exported data to an external site
- ◆ Parsing and submission, by importing, of user-submitted bugs or feature requests

You have virtually unlimited access to TrackRecord's types and items from ActiveX; through some clever programming, TrackRecord can be customized to do almost *anything!*

Basic Design

Through Automation, TrackRecord exposes a number of objects that represent parts of its internals. Apart from the main Application, each object has a singular form, representing a single instance of the object, and a plural form, representing a collection of objects. Collections can be iterated over using standard Automation methods.

Simple Example

Here's a simple Basic program that retrieves the items that match a particular query, and displays the results. Since Automation controllers may use different extension languages, this example may not work exactly as presented here. Despite that, the general structure and usage of TrackRecord objects, properties, and methods remains the same.

```
Dim TR As New TRDatabase.application
Dim MyQuery As TRDatabase.Query
Dim Results As TRDatabase.Items
Dim Item As TRDatabase.Item
```

```
Set MyQuery = TR.GetQuery("Tasks Due Today")
Set Results = MyQuery.Run
```

'If your controller has iteration support, you can use:

```
For Each Item In Results
    Print Item
Next Item
```

'Otherwise, you can explicitly iterate using:

```
NumItems = Results.Count

For i% = 0 To NumItems - 1
    Print Results.Item(i%)
Next i%
```

Object Reference

The following sections provide object properties and methods.

Application Object

The Application object is the basis for most of the fundamental operations that you will perform. Before doing anything, you must create an instance of the application object, using a line like:

```
Set TR = CreateObject ("TrackRecord.Application")
```

Once created, you can call any of the methods in the object to perform operations on your TrackRecord data.

Properties

CurrentUser	CurrentUser As Object
--------------------	-----------------------

The user name of the currently logged in user.

DatabaseDirectory	DatabaseDirectory As String
--------------------------	-----------------------------

Current database directory.

ServerName	ServerName As Boolean
-------------------	-----------------------

Name of the server machine.

TypesModifiedDate and Time	TypesModifiedDateandTime As String
-----------------------------------	------------------------------------

Read only. When the type definitions were last changed.

Methods

AutoLogin	Function AutoLogin([UserName As String],[Password As String],UserID As Long) As Boolean
------------------	---

Automatically log in to TrackRecord using previously captured user name and password.

CreateItems	Function CreateItems() As Items
--------------------	---------------------------------

Returns an empty Item collection.

Export	Function Export (OutputFile As String, TemplateFile As String) As Items
---------------	---

Exports all items from the database. TemplateFile is optional; if not specified, all items are exported to a TrackRecord exchange file. If specified, all items matching the template specification are exported to OutputFile in delimited ASCII format. Returns an Items collection containing all the items that were exported.

GetItem	Function GetItem (ItemIdentifier As String) As Item
----------------	---

Returns a single Item from the database, based on the Item's unique identifier. This identifier can be passed to an external program by an Executable button.

GetMaxUsers	Function GetMaxUsers() As Integer
--------------------	-----------------------------------

Get the maximum number of users allowed.

GetQueries	Function GetQueries (LocalQueriesOnly As Boolean) As Queries
-------------------	--

Returns all of the Queries in the database, as a Query collection. LocalQueriesOnly is optional, and defaults to FALSE. If TRUE, all private queries for the currently logged in user are returned. These queries can also be retrieved from the User object; see "User Object" on page 227 for more information.

GetQuery	Function GetQuery (QueryNameOrIdentifier As String) As Query
	Returns a single Query from the database, based on its name or unique identifier. If a name is specified, and more than one query has the same name, the first one created is returned. See "Query Object" on page 224 for more information on Query objects.
GetType	Function GetType (TypeName As String) As Type
	Returns a single Type object, based on its name.
GetUser	Function GetUser(Username As String) As User
	Retrieve a single user by name.
GetUserByID	Function GetUserByID (UserID As Long) As User
	Retrieve a single user by UserID.
GetTypes	Function GetTypes () As Types
	Retrieves all of the Types in the database, as a Types collection.
GetUsers	Function GetUsers () As Users
	Returns a collection of User objects corresponding to the list of Users defined for this database. The User collection returned is "active", and will update as users are added and deleted by the database Administrator.
GetUserSetting	Function GetUserSetting(sectionName As String, variableName As String) As String
	Get a preferences setting for the currently logged in user.
Import	Import (InputFile As String, FavorDatabase As Boolean, TemplateFile As String) As Items
	Imports items into the database from the InputFile. FavorDatabase is optional, and chooses how items are handled when they duplicate items already in the database. If TRUE (the default), the items in the database are retained if duplicates are imported from InputFile. TemplateFile is also optional; if specified, InputFile is assumed to be an ASCII delimited file. All the items that have been imported are returned, as an Item collection, when the import is complete. If the set is empty on return, there was an error.
IsDBOpen	Function IsDBOpen () As Boolean
	Is a database currently open?
LogIn	Function LogIn([UserName As String],[Password As String],UserID As Long) As Boolean
	Log in as a specific user.

LogOut	Function LogOut() As Boolean
---------------	------------------------------

Logs out of TrackRecord, or rather, Logs in as the Admin user.

ProcessItem-Changes	Function ProcessItemChanges(ScriptFile As Filename, [UserID As Long]) As Items
----------------------------	--

Process item change script. Item change scripts can be used to make changes to existing items, and to create new items. Returns the items that were changed.

The format for the script is as follows:

The first line must always be either [*<identifier>*] or [*“<type name>”*] where *<identifier>* is an identifier string for an existing item, or *<type name>* is the name of the item type that should be used for creating a new item.

Each of the following lines is of the form:

```
<field-name>\t<field-value>
```

where *<field-name>* is the name of the field, and *<field-value>* is the value to set for the field. The \t represents a hard tab character in the file (ASCII 9), which can be entered with most text editors. Certain types of fields have additional modifiers other than the name of the field:

- **Name**
For a field called “Name”, the options are *Name.First*, *Name.Last*, and *Name.Middle*.
- **Address**
For a fields called “Address”, the options are *Address.Address1*, *Address.Address2*, *Address.City*, *Address.State*, *Address.Country*, and *Address.Zip*
- **Timespan**
For a field called “Timespan”, the options are *Timespan.Start Time* and *Timespan.End Time*, which contain the combined date and time separated by a null character, and *StartDate*, *StartTime*, *EndDate*, and *EndTime* for date or time individually.
- **Date/Time Options**
For a field called “Date/Time”, the options are *Date/Time.Date* and *Date/Time.Time*
- **Checkboxes**
For checkboxes and radiobuttons, the *<field-value>* should start with a space if the checkbox/radio button should be unchecked, and an ‘X’ if it should be checked.
- **Lists**
For a field called Reported By, the options are *Reported By.Add* and *Reported By.Delete*. For either one, the *<field-value>* should be the identifier of the item to add or delete.
- **Compound type lists**
For compound type fields, the *<field-value>* is the identifier of the item to select.

After all the fields are listed, the item is ended with a line:
[enditem]

Normally, items with an empty default abbreviation are not saved by ProcessItemChanges. However, if you want to force all items to be saved, whether they are complete or not, use the following line to end the item:
[enditem save]

To ensure that items are unique, end the item with:
[enditem unique]

Duplicates are determined by comparison of the Dupe field. The new or matching item will be returned.

Each item can be followed by the end of the file, or by another item.

A sample script might be:

```
[“person”]  
Name.Last Strange  
Name.First Mickey  
Company a000000  
[enditem]
```

Notes:

1. Notepad field data requires double quotes.
2. Any quote character (“) within a string field requires a preceding backslash (\) character.

PurgeCache	Sub PurgeCache()
-------------------	------------------

Purge the cache.

Quit	Sub Quit()
-------------	------------

Obsolete. Provided for backward compatibility.

Abbreviation Object

An Abbreviation represents a specific way of formatting the output of an Item. The Abbreviations in ActiveX are the same as the abbreviations used in *regular* TrackRecord.

Properties

Application	Application (Read only)
--------------------	-------------------------

Application object.

IsDefault	Boolean (Read only)
------------------	---------------------

Is this the default abbreviation?

IsDup	Boolean (Read only)
--------------	---------------------

TRUE if this is the abbreviation used for duplicate elimination.

IsLong	Boolean (Read only)
TRUE if this is the long default abbreviation.	
IsShort	Boolean (Read only)
TRUE if this is the short default abbreviation.	
Name	String (Read only)
The name of the abbreviation.	
Number	Integer (Read only)
The identifying abbreviation number: typically, you pass in the abbreviation number to other TrackRecord methods.	

Abbreviations Object (Collection)

A collection of Abbreviation objects, stored in a standard, iterable form.

Methods

Count	Function Count() As Integer
Returns the number of Abbreviation objects in the collection.	
Item	Function Item (Index As Integer) As Abbreviation
Returns the Abbreviation at Index in the collection.	

Database Object

Permits the opening of a specified database.

Methods

AutoLogin	Function AutoLogin(Username As String, Password As String, databaseDir As String, serverName As String, forceLogin As Boolean, appObject as Application) As Long
Automatically log in to TrackRecord using previously captured user information.	
GetAppFromHandle	Function GetAppFromHandle(sessionHandle As Long) As Application
Gets an application object from its session handle.	
OpenDatabase	Function OpenDatabase(Database As String, (Servername As String)) As Application
Open a database and retrieve an application object.	

Field Object

Represents a single field of an item.

Properties

AccessLevel	AccessLevel As Long (Read only)
Access control level.	
Application	Application (Read only)
Application object.	
DefaultValue	String (Read only)
Returns the name of the field.	
FieldValid	Boolean (Read only)
Returns TRUE if the current field object is valid.	
GetAbbreviations	Abbreviations
Retrieve the field abbreviation.	
IsGroupStart	Boolean
Is this the start of a radio button group?	
IsHidden	Boolean
Is this field hidden?	
IsList	Boolean (Read only)
Returns TRUE if the Field is a list, FALSE otherwise.	
IsMail	Boolean
Is this field marked for mail?	
IsReadOnly	Boolean
Is this field read-only?	
IsRequired	Boolean
Is this field required?	
ItemValid	Boolean (Read only)
Returns TRUE if there is an item associated with the current field, and that item is valid.	
Modified	Boolean (Read only)
Returns TRUE if the field contents could have been modified since they were last examined. This property automatically resets if you retrieve Value or DefaultValue.	
Type	Type (Read only)
Returns a Type object that contains the type of the object stored in the field.	

Value	Items
--------------	-------

Returns an Items collection containing the contents of the Field. If `IsList` is `FALSE`, there can only be one object in the collection. Otherwise, the collection will contain the contents of the list.

Note that if the field is empty, the collection will be empty as well. Check the `Count` property of the Items collection before examining the collection's contents.

Valid	Boolean (Read only)
--------------	---------------------

Returns `TRUE` if the object is still valid. The object could become invalid if the item containing it was deleted while you still had an instance of the field.

FieldLayout Object

The FieldLayout object is used to get back information about the layout of an item (fields and their ordering).

Properties

Application	Application (Read only)
--------------------	-------------------------

Application object.

Methods

Count	Function Count() as Long
--------------	--------------------------

Get the number of fields in the collection.

Item	Function Item(Index As Long) As Field
-------------	---------------------------------------

Get specified field.

GetFieldInfo	Function GetFieldInfo(Field As Field, Row As Long, HorizontalPosition As Long, Width As Long) As String
---------------------	---

Returns information about screen layout for a field. The return value is the actual contents of the field.

ValuesFromItem	Function ValuesFromItem(Item As Items) As Boolean
-----------------------	---

Sets source item(s) for field values. If this function is called, subsequent calls to GetFieldInfo will return field values for the specified item.

ValuesFromTemplates	Function ValuesFromTemplates(Templates As Items) As Boolean
----------------------------	---

Sets source item(s) for field values. If this function is called, subsequent calls to GetFieldInfo will return values based on the set of templates passed in (combined with the default templates).

Fields Object (Collection)

A collection of Field objects, stored in a standard, iterable form.

Properties

Application	Application (Read only)
--------------------	-------------------------

Application object.

Methods

Count	Function Count() As Integer
--------------	-----------------------------

Returns the number of Fields in the collection.

Item	Function Item (Index As Integer) As Field
-------------	---

Returns the Field at Index in the collection.

ItemByName	Function ItemByName (Name As String) As Field
-------------------	---

Returns the Field in the collection named Name.

ItemByTag	Function ItemByTag (TagToFind As String) As Field
------------------	---

Get specified field by tag.

History Object

Represents a single entry in a Change History list.

Properties

Action	String (Read only)
Contains a description of the what was done to the item. Possible values are:	
<ul style="list-style-type: none">• Changed• Checked• ItemCreated• ItemDeleted• ListItemAdded• ListItemDeleted• Unchecked	
Application	Application (Read only)
Application object.	
Field	Field (Read only)
Returns the Field that was modified.	
NewState	Item (Read only)
What is the new state?	
OldCheck	Boolean (Read only)
If the changed Field is a checkbox or radio button, and <code>OldValueAvailable</code> returns <code>Check</code> , <code>OldCheck</code> returns the previous state of the checkbox or radio button.	
OldDate	String (Read only)
If the changed Field is a Date field, and <code>OldValueAvailable</code> returns <code>Date</code> or <code>DateAndTime</code> , <code>OldDate</code> returns the date in the field before the user changed it.	
OldDateAndTime	String (Read only)
If the changed Field is a Time field, and <code>OldValueAvailable</code> returns <code>DateAndTime</code> , <code>OldDateAndTime</code> returns the date and time in the field before the user changed it.	
OldItem	Item (Read only)
If the changed Field is a reference to another item, and <code>OldValueAvailable</code> returns <code>Item</code> , <code>OldItem</code> returns the previous item in the field. For list actions, it returns the item added or deleted from the list.	
OldNumber	Long (Read only)
If the changed Field is a numeric field, and <code>OldValueAvailable</code> returns <code>Number</code> , <code>OldDateAndTime</code> returns the date and time in the field before the user changed it.	
OldState	Item (Read only)
What was the old state?	
Type	Type (Read only)
Returns the type of the Field that was changed.	

User String (Read only)

Returns the name of the user who modified the item.

Value String

Returns the value of the current history record.

Methods

DateAndTime Function DateAndTime() As String

Returns the date and time the item was modified.

OldValueAvailable Function OldValuesAvailable() As String

Returns the type of "old" value available in this History object. Possible return values are:

- Check
 - Date
 - DateAndTime
 - Item
 - Note
 - Number
-

Histories Object (Collection)

A collection of History objects, stored in a standard, iterable form.

Properties

Application Application (Read only)

Application object.

Value string

Returns the value of all of the History information for the item, each entry on its own line, separated by "\n", as it would be displayed in the Change History column of the Outline view.

Methods

Count Function Count() As Integer

Returns the number of History objects in the collection.

Item	Function Item (Index As Integer) As History Returns the History object at Index in the collection.
Since	Function Since (DateAndTime As String) As Histories Returns the History objects entered since the passed date and time.

Item Object

An Item is a single object from the TrackRecord database.

Properties

Application	Application (Read only) Application object.
Default	Boolean Contains TRUE if the item is a default template.
DefaultValue	String (Read only) Contains the default abbreviation for the Item.
Fields	Fields (Read only) Contains the collection of Fields available in this object.
Identifier	String (Read only) Contains the uniquely identifying string that represents this Item. This identifier can be saved and used later to retrieve the same object.
IsReadOnly	Long 0=read/write, 1=read only, 2=read/add Is this field read-only?
Modified	Boolean (Read only) Contains TRUE if the Item may have been modified since the last time you retrieved one of its properties.
Private	Boolean Contains whether or not the item is local to the current user. Can only be set if Template is TRUE.
Template	Boolean Contains TRUE if the item is a template.
Type	Type (Read only) Contains the Type object corresponding to the type of the current Item.
Valid	Boolean (Read only) Contains TRUE if the object is still valid. The object could become invalid if the item was deleted while you still had an instance of it.

Methods

Cast	Function Cast (Type As Type) As Item
Attempts to cast this item to the passed Type. (For example, if there is an identity for this item of the specified type, it will return that identity.)	
Contents	Function Contents (Field As Field) As Items
Returns the contents of the specified Field.	
DeleteItem	Function DeleteItem() As Void
Deletes the specified item from the database. Does not apply access controls to the operation, so the item is always deleted, regardless of the privileges of the currently logged in user. This function can be very destructive, so be sure to carefully test any code that uses it.	
Downcast	Function Downcast() As Item
Casts the current Item to its “most derived” identity. For example, if you have retrieved a Task from the database, and the object also has a Bug Report identity, Downcast will return the Bug Report Item.	
Export	Function Export (OutputFile As String, TemplateFile As String) As Items
Exports the current Item to the specified OutputFile. TemplateFile is optional; if specified, the Item is written to a delimited ASCII file as specified in the template.	
ExtractAttachment	Function ExtractAttachment(ExtractToFile As String) As String
Extract attachment to a file.	
History	Function History (DateAndTime As String) As Histories
Returns the History objects associated with this Item. DateAndTime is optional: if not specified, all associated History items are retrieved. Otherwise, all History items since DateAndTime are retrieved.	
Identities	Function Identities () As Types
Returns the types of the other identities associated with this item, if any.	
Value	Function Value (Field As String, Abbreviation As Integer) As String
Returns the Abbreviation of the specified Field. Abbreviation is optional, and defaults to the default abbreviation (abbreviation 0).	
ValueByFieldAnd Abbrev	Function ValueByFieldAndAbbrev (Field As Field, Abbreviation As Abbreviation) As String
Performs the same function as the previous command, using Field and Abbreviation objects instead of Strings.	

Items Object (Collection)

A collection of Item objects, stored in a standard, iterable form. In addition to the typical methods, an Items collection can also have various set operations performed on it.

Properties

Application	Application (Read only)
--------------------	-------------------------

Application object.

Modified	Boolean (Read only)
-----------------	---------------------

Are there any modified items in the collection?

Methods

Add	Function Add (ItemToAdd As Item) As Void
------------	--

Adds ItemToAdd to the collection.

AddIdentifier	Sub AddIdentifier (IdentifierOfItemToAdd As String) (
----------------------	---

Add an item to the collection by identifier.

Clear	Function Clear () As Void
--------------	---------------------------

Empties the collection.

Count	Function Count () As Integer
--------------	------------------------------

Returns the number of Items in the collection.

Export	Function Export (OutputFile As String, TemplateFile As String) As Items
---------------	--

Exports all the Items in the collection to OutputFile. TemplateFile is optional; if specified, the Items are exported in delimited ASCII form, as specified in the template.

GetModifiedItems	Function GetModifiedItems () As Items
-------------------------	---------------------------------------

Returns a collection containing all of the Items that may have been modified since the last time the function was called.

Identifier	Function Identifier(Index As Long) As String
-------------------	--

Get specified item identifier.

Intersect	Function Intersect (ItemCollection As Items) As Items
------------------	--

Performs an intersection with the passed ItemCollection, returning the Items in common as the result.

Item	Function Item (Index As Integer) As Item
-------------	--

Returns the Item at Index in the collection.

Remainder	Function Remainder (ItemCollection As Items) As Items
	Returns the collection Items that aren't in both the current Items collection and the passed ItemCollection.
Remove	Function Remove (ItemToRemove As Item) As Void
	Removes ItemToRemove from the collection, if there.
Run	Function Run (QueryToRun As Query, DateSpecification As String, [Sort As Boolean]) As Item
	Runs the passed Query on the Items in the collection, returning the result. DateSpecification is optional; if specified, the Query is run as a Calendar query, substituting the passed date as it would in the Calendar view.
Union	Function Union (ItemCollection As Items) As Items
	Combines the items in the current collection with ItemCollection, returning the result.

Query Object

A Query is a single query object from the database. It can be used to selectively extract Items based on their content.

Properties

Application	Application (Read only)
	Application object.
CalendarQuery	Boolean (Read only)
	Returns TRUE if this Query can be used as a Calendar query.
Identifier	String (Read only)
	Contains the uniquely identifying string that represents this Query. This can be saved and used later to retrieve the same object.
MailQuery	Boolean
	Returns TRUE if the Query is a Mail query.
Modified	Boolean (Read only)
	Contains TRUE if the Query may have been modified since the last time you ran it.
NeedsToRun	Boolean (Read only)
	Contains TRUE if the database has changed in such a way that the results of the query might be different.
Private	Boolean
	Contains whether or not the item is local to the current user.

Valid	Boolean (Read only)
--------------	---------------------

Contains TRUE if the object is still valid. The object could become invalid if the item was deleted while you still had an instance of it.

Value	String (Read only)
--------------	--------------------

Contains the name of the Query.

Methods

Export	Function Export (OutputFile As String, TemplateFile As String, DateSpecification As String) As Items
---------------	--

Exports the results of the Query to OutputFile. TemplateFile is optional; if specified, the Items are exported in delimited ASCII form, as specified in the template. If a DateSpecification is provided, the Query is run as a Calendar query, substituting the passed date as it would in the Calendar view.

Run	Function Run (DateSpecification As String,[Sort As Boolean]) As Items
------------	---

Runs the Query on the database, returning the result. DateSpecification is optional; if specified, the Query is run as a Calendar query, substituting the passed date as it would in the Calendar view.

Queries Object (Collections)

A collection of Queries, stored in a standard, iterable form.

Properties

Application	Application (Read only)
--------------------	-------------------------

Application object.

Methods

Add	Function Add (QueryToAdd As Query) As Void
------------	--

Adds QueryToAdd to the collection.

Clear	Function Clear () As Void
--------------	---------------------------

Empties the collection.

Count	Function Count () As Integer
--------------	------------------------------

Returns the number of Queries in the collection.

Item	Function Item (Index As Integer) As Query
Returns the Query at Index in the collection.	
Remove	Function Remove (QueryToRemove As Query) As Void
Removes QueryToRemove from the collection, if there.	

Type Object

Represents an item Type from the database. Every Item object has a Type, and a given Type can be derived from other Types, inheriting their properties, or have Types derived from it.

Properties

Application	Application (Read only)
Application object.	
Fields	Fields (Read only)
Retrieves the collection of Fields associated with this Type.	
GetAbbreviations	Abbreviations (Read only)
Returns the Abbreviations that can be used with this Type.	
GetInheritedTypes	Types (Read only)
Contains the Types that this Type is derived from, if any.	
IsInShortList	Boolean (Read only)
Is this type in the short list?	
NewItems	Items (Read only)
Returns the collection of Items of the current Type that have been created since the Type object was retrieved, or since NewItems was last checked.	
Valid	Boolean (Read only)
Is this type a valid type?	
Value	String (Read only)
Returns the name of the Type.	

Methods

Field Layout	Function FieldLayout(LayoutType As Long) As FieldLayout
---------------------	---

Retrieve FieldLayout objects for viewing this type. The LayoutType is reserved for future use; pass 0 for this value.

GetDuplicates	Function GetDuplicates(dupeString As String) As Items
----------------------	---

Returns the set of Items that have the specified Dupe abbreviation.

GetItemInfo	Function GetItemInfo(LastID As String, NumItems As Long, StartWith As String) As String
--------------------	---

Retrieve next NumItems items of this type. The strings returned are in the form of the default abbreviation for the first item, followed by a newline character, followed by the identifier of the item, followed by another newline character, followed by the next abbreviation, and so on.

Items	Function Items() As Items
--------------	---------------------------

Returns all the Items in the database of the current Type.

Types Object (Collection)

A collection of Types, stored in a standard, iterable form.

Properties

Application	Application (Read only)
--------------------	-------------------------

Application object.

Methods

Count	Function Count() As Integer
--------------	-----------------------------

Returns the number of Types in the collection.

Item	Function Item (Index As Integer) As Type
-------------	--

Returns the Type at Index in the collection.

User Object

A User is an individual, created by the database administrator, who can log in to the database and perform operations. Each user has a number of properties associated with it that can be examined through this ActiveX object. In addition, the user's private Query and Template objects can be retrieved.

Properties

Active	Boolean (Read only)
Returns TRUE if the administrator has activated this user, FALSE otherwise.	
Admin	Boolean (Read only)
Returns TRUE if the user is an administrative user and FALSE otherwise.	
Application	Application (Read only)
Application object.	
DefaultValue	String (Read only)
Returns the name of the User.	
MailAddress	String (Read only)
Contains the mail address defined for this User.	
MailEnabled	Boolean (Read only)
Returns TRUE if mail processing has been enabled for this user.	
Modified	Boolean (Read only)
The property is TRUE if this User may have been modified since it was retrieved.	
UserID	Long (Read only)
Returns an integer that can be passed to ProcessItemChanges to specify that the current user be referenced in any Change History entries that result from script processing.	
UserName	String (Read only)
Retrieves the name of the User.	
Valid	Boolean (Read only)
Returns TRUE if the User is still valid, FALSE otherwise. A User can become invalid if the administrator deletes it while an automation client still has an active reference to it.	

Methods

GetQueries Function GetQueries (IncludeNonMail As Boolean, IncludeMail As Boolean) As Queries

Retrieves all the private queries associated with this user. By providing different values for IncludeNonMail and IncludeMail, various combinations of mail and private queries can be returned.

Note that the collection retrieved from GetQueries remains active. As private and/or mail queries are added/removed for this user, they're added to or removed from the collection.

GetTemplates Function GetTemplates () As Items

Retrieves all of the private templates associated with this user.

Note that the collection retrieved from GetTemplates remains active. As templates are added/removed for this user, they're added to or removed from the collection.

Users Object (Collection)

A collection of Users, stored in standard, iterable form.

Properties

Added Boolean (Read only)

Returns TRUE if there are new users in the collection. These new User objects can be retrieved with GetAddedUsers, documented below.

Application Application (Read only)

Application object.

Modified Boolean (Read only)

Returns TRUE if any users in the collection may have been modified. The potentially modified User objects can be retrieved with GetModifiedUsers, documented below.

Methods

Count Function Count () As Integer

Retrieves the number of users in the collection.

GetAddedUsers Function GetAddedUsers () As Users

Gets the collection of users that were added since the original Users collection was instantiated, or since this method was last called.

GetModifiedUsers Function GetModifiedUsers () As Users

Gets the collection of users that may have been modified since the original Users collection was instantiated, or since this method was last called.

Item Function Item (Index As Integer) As User

Returns the User at Index in the collection.

Glossary



- abbreviation** Text used to identify an item in the TrackRecord database.
- absolute date** Entry in a date field that identifies a specific date; for example, *May 19, 1999* or *2/14/87*. *Compare* relative date.
- access control** Function of an application that restricts user access to data according to permissions derived from the user's identity or membership in groups.
- ActiveX control** An interface object on a form that enables or enhances a user's interaction with an application. ActiveX controls have an `.ocx` file name extension.
- administrator** The person responsible for installing, configuring, and maintaining TrackRecord user accounts. The TrackRecord administrator may or may not be responsible for maintaining the network and database for a site.
- ASCII file** American Standard Code for Information Interchange (pronounced ASK-ee). Code in which numbers from 0 to 255 represent individual characters, such as letters, numbers, and punctuation marks. Standard for text representation.
- audit trail** Record identifying each change made to an item or report and the identity of the person making the change. *See* change history.
- bug report** *See* defect.
- change history** Record identifying each change made to an item or report and the identity of the person making the change. *See* audit trail.
- child header** A header that is subordinate to another header within an Outline Report. In the following example, Open Bugs is a child header of Bugs, and Priority 1 is a child header of Open Bugs. *Compare* parent header, sibling header.

```
Bugs
  Open Bugs
    Priority 1
```

coverage	Percentage of a project that has been tested, expressed as a percentage of lines tested or a percentage of the functions tested.
defect	Report describing a problem found with software or hardware undergoing testing. A defect report is updated over time as a problem is reported, reproduced, fixed, and tested.
Defect	Item in the TrackRecord database describing a specific problem identified during project testing.
derived type	A data type that inherits all of the fields of its parent's type. For example, a Software User object inherits all the fields from the Person data type.
export	To generate a formatted file containing data from the TrackRecord database for use by another application.
filter	Process that selects information from the database according to user-specified criteria.
function coverage	A measure of how much of the code for a project has been tested, expressed as a percentage of functions tested across all TrueCoverage sessions relative to the total number of functions in a project.
header	Entry in an outline report that corresponds to a database query.
Header Wizard	TrackRecord utility that simplifies the process of defining and formatting outline reports. The Header Wizard was called the Header Engineer in previous TrackRecord releases.
import	To bring data generated by another application into the TrackRecord database.
inheritance	Property of data types that allow them to derive their structures from parent types.
item	Object or entity of a specific type in the TrackRecord database. Equivalent to a record in a relational database.
label	User-defined name you can attach to a specific software release or build in version control systems.
line coverage	A measure of how much of the code for a project has been tested, expressed as a percentage of code lines tested across all TrueCoverage sessions relative to the total number of code lines in a project.
link	Relationship between items in the TrackRecord database. A link is created whenever one item references another. For example, attaching a note to a Person object creates a link between the two items.

merging	The process of compiling coverage information across multiple test sessions.
milestone	An event or point in a schedule that marks the transition of a project from one phase to another.
Milestone Status	Report that identifies the current status of a project with respect to a specified milestone, the volatility of a project from one build to another, data submitted from DevCenter or QACenter tools, and custom queries that summarize project information.
outline report	Structured view of information in the TrackRecord database that is updated dynamically.
parent header	A header that owns one or more subordinate headers within an outline report. In the following example, Bugs is the parent header of Open Bugs, and Open Bugs is the parent header of Priority 1 and Priority 2. <i>Compare</i> sibling header, child header. <pre> Bugs Open Bugs Priority 1 Priority 2 Closed Bugs </pre>
password	Text string used to verify the identity of a user before granting the user read or write access to TrackRecord information.
priority	Arbitrary integer value assigned to a defect to indicate its significance relative to other defects. Depending on the standards of your site, you can use a higher or lower number to indicate that one defect is more important than another.
recurring events	Event scheduled to occur at regular intervals, such as <i>Every Thursday</i> .
relative date	Entry in a date field that identifies a date relative to the current system date; for example, <i>Yesterday</i> or <i>Tomorrow</i> . <i>Compare</i> absolute date.
right-click	Pressing and releasing the right mouse button while positioning the cursor over an object or control on the screen. Typically opens a shortcut menu, which displays options applicable to the control or item over which the cursor is positioned.
shared	Property of a report that allows TrackRecord users other than the report's owner to view it. Only the owner of a report can modify a shared report, though other users can copy it (by saving it under a new name) and then modify their copy.
shortcut menu	Menu that appears when you right-click a control or item in a window.

sibling header

A header that is at the same level as another header within an outline report. In the following example, Open Bugs is a sibling header of Closed Bugs, and Priority 1 is a sibling header of Priority 2. *Compare* parent header, child header.

```
Bugs
  Open Bugs
    Priority 1
    Priority 2
  Closed Bugs
```

subproject

A project that is one component of a larger or more complex project.

task

Any activity in a project that has a beginning date and an end date.

Task

An item in the TrackRecord database that identifies a task in a project.

version control system (VCS)

An application that maintains multiple versions of source files, controls who can read and update source files, and records information about changes to source files.

volatility

A measure of the stability of code from one build to another, expressed as a percentage of the functions that changed between two builds.

C

- Calendar button, 22
- Calendar view, 62
 - opening, 65
 - setting up, 64
- Change History, 9
- Change History tab, 21
- Changing
 - queries, 66
 - the database, lock-out during, 159
 - the status of an item, 28
 - views, 66
 - your password, 16
- Check database, 182
- Checkbox, displaying in reports, 41
- Checking files in and out of a VCS system, 100
- Checking the database for changes, 82
- Child headers, 44
- Child types, 156
- Choice items, 173
- Client-Server architecture, described, 10
- Cloning projects, 143
- Code volatility information, 57
- CodeReview, 113
- Columns
 - in Outline Reports, 40
 - labeling with Show As, 41
- Command line interface for exporting, 201
- Common types, 157
- Configuration tasks, 134
- Controlling access to database, 135
- Copy database, 182
- Creating
 - a project, 142
 - a query, 33
 - a rule for types, 174
 - a shared query, 144
 - a workflow, 148
 - graphs, 51
 - groups, 136
 - items in Outline Reports, 44

- new databases, 180
- templates, 75
- types, 161

- Custom items, 173
- Customizing TrackRecord, 81

D

- Data
 - entering, 19
 - finding, 68
 - finding with queries, 32
 - import and export, 199
 - update frequency, 82
- Data collected by DevPartner, 118
- Data hierarchy defined, 8
- Data types. See Types.
- Database
 - administration, 180
 - check, 182
 - copy, 182
 - create new, 180
 - described, 7
 - limiting access to, 135
 - links, 7
 - lock-out, 159
 - logging in to, 12
 - migration note, 179
 - overview of, 7
 - polling, 82
 - rebuild, 183
 - restoring, 181
 - schedule backups for, 181
- Date format, for WebServer, 191
- Dates, entering on forms, 22
- DCOMCNFG settings, 190
- Debug session output, 119
- Default abbreviation
 - defining, 163
 - described, 157
 - using for column labeling, 41
- Default behavior, setting, 81

- Default font for Outline Reports, 84
- Default status
 - for defects, 155
 - for other types, 155
- Defect
 - creating, 26
 - submitting from QADirector, 128
- Deleting
 - groups, 137
 - items, 23
 - queries, 66
 - views, 66
- DevPartner tools
 - automated session submission, 117
 - build registration, 117
 - coverage information, 122
 - data collected by, 118
 - debug session output, 119
 - duplicates, 121
 - entering a defect from, 118
 - IDE integration, 115
 - instrumenting code, 117
 - integration with, 113
 - logging in from, 114
 - menu, 116
 - multiple matching builds, 122
 - open task list option, 116
 - remote error capture, 125
 - submitting CodeReview problems, 118
 - submitting debug sessions, 120
 - submitting sessions, 117
 - submitting tuning sessions, 121
 - toolbar buttons, 116
 - viewing task lists, 122
- Disabling the workflow, 147
- Dragging headers, 44
- Dupe option on the Type Properties dialog, 158
- Duplicate items, merging, 29
- Duplicate types when importing, 158

E

- Edit menu, 13
- Edit types, 161
- E-mail messages, enabling, 192
- Executable commands, 7
- Executable files, versions of, 124
- Expand Item button, 22
- Expand Text Field button, 24
- Export
 - command line interface, 201
 - data, 199
 - menu option, 202
 - using copy and paste, 203

F

- FailSafe, 113
- Fault Manager, integration with, 128
- Favorite reports, creation of global, 145
- Favorite reports, identifying, 67
- Favorite types, 83
 - creation of global, 145
- Favorites menu, 14
- Field properties
 - general tab, 166
 - privileges tab, 171
 - properties tab, 168
- Fields
 - access rights, 158
 - including in mail, 167
 - inserted into inherited section of form, 166
 - inserting into a form, 164
 - limiting access to, 135
 - overview of, 7
 - rearranging on a form, 164
- File menu, 13
- Files, checking into a VCS system, 100
- Filtering data in Milestone Status views, 58
- Finding data, 68
- Finding data using queries, 32
- Font, for Outline Reports, 84

Forms

- controls on, 22
- described, 19
- example of using, 26
- how to open, 26

G

General concepts, 1

Global

- access, 135
- Outline Report, 144
- preferences, 143
- query, 144
- shared views, 137
- templates, creating, 146

Graphics applications, integration with, 88

Graphs, 51

- copying to other applications, 88
- formatting, 53

Groups

- creating, 136
- deleting, 137
- described, 135
- impact of no assignment, 135

H

Header wizard, 47

Headers

- creating with header wizard, 47
- levels in Outline Reports, 44
- rearranging, 44
- remainder, 45

Help, how to use, 16

Hierarchy, data, 8

Historical information, value of, 143

Home page

- creation of global, 145
- displaying at start of session, 81
- identifying, 67

I

IDE integration, 115

Import

- data, 199
- duplicate types, 158
- file specifications, 206
- manually creating templates for, 203
- troubleshooting, 207

Import and export menu options, 202

Import/Export wizard, 199

Importing resources into Project, 92

Inactive user, 140

Inheritance, 8

- of rules, 174

Inserting fields into a form, 164

Integration with version control systems, 97

Item browser button, on Toolbar, 15

Item Browser window, 25

Item tab, on a form, 20

Items

- changing status of, 28
- creating new in Outline Reports, 44
- merging, 29
- opening in Outline Reports, 44
- overview of, 7

J

Java console

- diagnostics, 190
- enabling, 190

JCheck, 113

L

Licensing, 133

Limiting access to database, 135

Links tab, on a form, 20

Links, overview of, 7

Locking the database, 159

Logging in, 12

- as someone else, 85
- automatically, 82

- from DevPartner tools, 114
- Login ID, for database access, 139
- Login window, 12
- Logins disabled, 160

M

- Macros, Microsoft Project, installing, 90
- Main window, described, 13
- Mapping priorities with Microsoft Project, 93
- Membership in a team, 140
- Menu bar, 13
- Merging items, 29
- Microsoft Outlook, copying to, 87
- Microsoft Project
 - importing resources, 92
 - integration with, 89
 - logging into TrackRecord, 91
 - priorities, 93
 - synchronizing project information, 94
 - table of synchronization, 96
- Microsoft Word, copying to, 87
- Migration, from V4, 179
- Milestone Status view, 54
 - code volatility in, 57
 - creating, 56
 - filtering data, 58
 - not available due to missing tags, 159
 - queries in, 58
 - session information, 57
 - sharing, 57
 - use of tags in, 55
 - volatility graph, 62
 - volatility statistics, 59
- Modifying
 - a workflow, 148
 - queries, 66
 - types, 161
- Multiple
 - matching builds, 119
 - projects, 143
 - workflow states, 153

N

- New item button, on Toolbar, 15
- Numeric reports, 38

O

- Object database, 7
- Office Assistant, messages in, 90
- Online help, 16
- Open Home Page button, on toolbar, 15
- Open item button, on Toolbar, 15
- Opening
 - existing queries, 66
 - existing views, 66
 - items in Outline Reports, 44
- Options window
 - favorite types tab, 83
 - General tab, 81
 - Outline Reports tab, 84
- Order of rules inheritance, 174
- Outline Reports, 37
 - adding headers, 38
 - checkboxes in, 41
 - copying to text editors, 87
 - creating, 38
 - creating a global, 144
 - creating from inside a query, 36
 - creating graphs from, 51
 - creating items in, 44
 - default font, 84
 - double-clicking headers, 85
 - global, 144
 - header wizard, 47
 - multiple headers in, 44
 - opening items in, 44
 - printing, 42
 - rearranging headers in, 44
 - remainder header, 45
 - setting up columns, 40
 - setting up defaults, 84
 - templates for creating items in, 74
- Output of debug sessions, 119

Overview, 1
Overview of database concepts, 7

P

Parent types, 156
Parent/child relationships, 8
Password, 12
 changing, 16
Platform configuration information, with Dev-
 Partner tools, 123
Popup frames in online help, 17
Potential Duplicate Items Found, in DevPart-
 ner tools, 121
Preferences
 global, 143
 individual, 81
Preserving window layout, 81
Printing Outline Reports, 42
Priorities, in Microsoft Project, 93
Privileges
 assigning to Team Members, 141
 group, 135
Project
 administration tasks, 141
 cloning, 143
 creating, 142
 defined, 141
 multiple, 143

Q

QABatch, 127
QACenter, integration with, 127
QADirector, 127
 submitting a defect, 128
QAHiperstation, 127
QAHiperstation+, 127
QALoad, 127
QAPlayback, 127
QAPlayback+, 127
QARun, 127

Query

 composition strategies, 36
 creating, 33
 defined, 31
 deleting, 66
 described, 32
 for DevPartner tools, 119
 for e-mail notification, 35
 global, 144
 in Milestone Status views, 58
 modifying, 66
 naming, 34
 shared, 144
 sharing, 35
 sorting, 35
 testing, 36
 with Outline Reports, 38

R

Rearranging headers, 44
Rebuild database, 183
Registering builds, 175
Remainder header, 45
Remote error capture, 125
Resources (Project), 92
Restoring a database, 181
Restricted types, 157
Restricting access to a type, 164
Rules Wizard, 175
Rules, creating for types, 174
Run Query Now button, 36

S

Save and Close, 28
Save and Duplicate, 28
Save and New, 28
Save options, 28
Search tool, hiding, 15
Search, on Toolbar, 15
Searching for data, 68
 using queries, 32

- Sending a message for database lock-out, 159
- Sessions
 - automatic submission, 82
 - developer tool, 117
 - information in a Milestone Status, 57
- Share groups, 137
- Sharing
 - queries, views, templates, 137
 - templates, 146
- Shortcuts, on the Toolbar, 14
- Show All check box, for types, 157
- Show As, in column set up, 41
- Sibling headers, 44
- Simple search, 68
- SmartCheck, 113
- Starting TrackRecord, 12
- States and actions, described, 147
- States, in workflow, 150
- Status
 - changing, 28
 - default, 155
- Status bar, 15
- Status field
 - adding to a type, 153
 - used with workflow, 147
- Submitting a defect from QACenter, 128
- Submitting tool sessions, 117
- Summary reports, 38
- Synchronizing information with Microsoft Project, 94

T

- Tags, ActiveLink, 158
- Task, updating from DevPartner tools, 120
- Team member
 - change history tab, 140
 - creating, 139
 - in a share group, 138
 - item tab, 140
 - links tab, 140
 - membership tab, 140
 - user tab, 140
 - username for database access, 139
 - without database access, 139
- Templates, 73
 - associating with a header, 77
 - creating, 75
 - creating global, 146
 - default status, 155
 - for importing, creating manually, 203
 - naming, 76
 - nested order, 73
 - sharing, 146
 - within Outline Reports, 74
- Testing a query, 36
- Text editors, integration with, 87
- Toolbar, described, 14
- Tools menu, 14
- Tracking states in workflow, 147
- TrackRecord menu, in Project toolbar, 90
- Transitions in workflow, 151
- TRTOMDB.EXE, 201
- TrueCoverage, 113
- TrueTime, 113
- Type editor, using, 164
- Type properties
 - button, 162
 - general tab, 162
- Types
 - access rights, 158
 - adding a status field to, 153
 - categories of, 157
 - child, 156
 - creating, 161
 - creating rules for, 174
 - favorites, 83
 - inheritance, 156
 - limiting access to, 135
 - modifying, 161
 - overview of, 7
 - parent, 156
 - restricting access to, 164

- restrictions on modifications, 172
- show all, 157

Types of access rights, 136

U

Updates to tasks from DevPartner tools, 120

User

- authorizing, 140
- status options, 140
- types of, 10

Usernames, for database access, 139

V

Version control systems, 97

- checking files in and out, 100
- synchronizing with, 101
- troubleshooting, 101

Version resource strings, 124

Viewing data, 37

Views

- changing, 66
- defined, 31
- deleting, 66

Volatility graph, 62

Volatility statistics, 59

W

WebServer

- administering, 184
- administrator menu, 186
- date format, 191
- ending a session, 185
- log files, 185
- logging in, 105
- logging out, 111
- opening an Outline Report, 110
- processes, 184
- sessions, 184
- troubleshooting, 189
- web monitor, 189

Window menu, 14

Workflow

- actions, 151
- adding a status field to a type, 153
- based on type, 153
- building blocks of, 148
- creating, 148
- described, 147
- disabling, 147
- modifying, 148
- states in, 150
- transitions, 151

Workflow Editor, described, 148

X, Y, Z

YourProject, sample project, 142