



Test Management Tools Series

ApTest™ Manager Admin Guide

TEST MANAGEMENT TOOLS SERIES

APTEST MANAGER ADMIN GUIDE

VERSION 2.22
AUGUST 2010

Copyright © 2000-2010 - Applied Testing and Technology, Inc.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Applied Testing and Technology, Inc.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. APPLIED TESTING AND TECHNOLOGY, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

ApTest is a trademark of Applied Testing and Technology, Inc. All other product and brand names used herein are service marks, trademarks, or registered trademarks of their respective companies or trademark owners. Applied Testing and Technology disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Applied Testing and Technology, Inc.

9298 Central Avenue • Suite 309
Blaine, MN 55434 USA
Phone 763-786-8160 • Fax 763-786-8180

www.aptest.com

SIGNIFICANT REVISIONS

Revision	Date
Version 2.22	August 2010
Chapter 1 – Manage Test Suites	
Added information about how to import tests and requirements from other test suites.	
Added information about how to specify which result means ‘incomplete’.	
Chapter 2 – Administration	
Added information about an optional warning that can be issued when ApTest Manager needs to change the name of a test case or requirement because it is not portable.	
Added information about permitting embedded table editing in wysiwyg and wysiwyg2 style fields.	
Updated screen shots.	
Version 2.21	February 2010
Chapter 2 – Administration	
Added information about how to reference execution fields from the bug tracking URI.	
Added information about how to enable step-by-step results when executing tests.	
Added information about new wysiwyg2 style.	
Updated screen shots.	
Version 2.20	September 2009
Entire Document	
Migrated to new house style.	
Chapter 1 – Managing Test Suites	

Revision	Date
Added information about marking values in fields as being obsolete. Added information about making fields as being graphable. Chapter 2 – Administration Added a note about marking accounts as disabled.	
Version 2.19	April 2009
Chapter 1 – Managing Test Suites Added information about new owner-related email triggers. Added information about how to cleanup deleted files. Added information about problem-report related reserved execution fields.	
Version 2.18	September 2008
Chapter 1 – Managing Test Suites Test Case selector popup window for atm_tests field (1.13.3) Chapter 2 – Administration LDAP Account Creation and Modification (2.2.3) Manage LDAP Server Configuration (2.2.5) Normal user access controls for Run these tests/Run associated tests (2.3.4.13) Test email configuration (2.3.10)	
Version 2.17	February 2008

Revision	Date
Chapter 1 – Managing Test Suites	
Checkbox and radio button field types (1.13.4, 1.14.4, 1.18.3)	
Version 2.16	October 2007
Chapter 1 – Managing Test Suites	
Date and Date-Time types for Session Variables (1.18.3)	
Chapter 3 – Advanced Topics	
Added Section on moving to a new server (3.6)	
User defined logo for menu bar in reports (3.7.3)	
Version 2.15	June 2007
General	
Run Data Fields renamed to Execution Fields	
Chapter 1 – Managing Test Suites	
Enable Requirements (1.7)	
Create tests from requirements (1.9)	
Create requirements from tests (1.8)	
Requirements Fields (1.16)	
Special fields (1.16.3, 1.17.3)	
Autonumbering (1.16.3)	
atm_mcomments field (1.16.3.1)	

Revision	Date
Requirement ->Test Case linking Fields (1.16.3.1)	
Automatic entries for modification history table for copy, rename, and import (1.16.4)	
Name part Requirement and Test Case Field flag (1.16.6)	
audittrail Execution Field flag (1.17.5)	
Configurable work day length (1.14)	
Result code ordering implications (1.20)	
Chapter 2 – Administration	
Normal user access restriction for lock/unlock Test Cases	
Time style	
Chapter 4 – SQL Datastore	
Chapter added	
Version 2.14	March 2006
Split off from User's Guide	

Table of Contents

SIGNIFICANT REVISIONS.....	II
PREFACE.....	XI
MANUAL ORGANIZATION.....	XI
DOCUMENTATION SET	XI
STYLISTIC CONVENTIONS	XII
CUSTOMIZATION.....	XII
1 MANAGING TEST SUITES	1-1
1.1 CHANGE THE TEST SUITE'S DESCRIPTION.....	1-2
1.2 COPY THE TEST SUITE	1-2
1.3 RENAME THE TEST SUITE.....	1-2
1.4 DELETE THE TEST SUITE	1-2
1.5 ARCHIVE THE TEST SUITE	1-2
1.6 RENUMBER THE AUTO-NUMBERED TESTS/REQUIREMENTS	1-2
1.7 ENABLE REQUIREMENTS.....	1-3
1.8 CREATE REQUIREMENTS FROM TEST CASES.....	1-3
1.9 CREATE TEST CASES FROM REQUIREMENTS.....	1-4
1.10 MANAGE THE TEST SUITE'S SQL DATASTORE.....	1-5
1.11 SYNCHRONIZE THE TEST SUITE	1-5
1.12 CLEANUP HIDDEN DELETED FILES	1-5
1.13 IMPORT FROM ANOTHER SUITE	1-5
1.14 TEST SUITE CONFIGURATION.....	1-6
1.14.1 Configurable Elements	1-6
1.14.2 Configuration Editors.....	1-7
1.15 CONSIDERATIONS FOR TEST SUITE DESIGN.....	1-8
1.15.1 Defining Requirements and Test Case Fields.....	1-8
1.15.2 Defining Test Case Selectors	1-8
1.15.3 Defining Requirements and Test Cases	1-9
1.15.4 Defining Test Sets	1-9
1.15.5 Defining Execution Fields and Session Variables	1-9
1.15.6 Defining Test Results	1-10
1.15.7 Defining Templates	1-10
1.16 CONFIGURING REQUIREMENT AND TEST CASE FIELDS.....	1-10
1.16.1 Requirement/Test Case Field editor	1-11
1.16.2 Requirement and Test Case Field Attributes	1-11
1.16.3 Special Requirement and Test Case Fields	1-12
1.16.3.1 Reserved Field Names.....	1-15
1.16.4 Requirement/Test Case Field Types	1-15
1.16.5 Requirement and Test Case Field Styles	1-21
1.16.5.1 ID Field Styles.....	1-21

1.16.5.2	Menu/Text Field Styles	1-22
1.16.5.3	Date Field Styles	1-22
1.16.5.4	Userlist Field Styles	1-22
1.16.5.5	Textarea Field Styles	1-22
1.16.5.6	Check boxes/Radio buttons Field Styles	1-24
1.16.6	<i>Requirement and Test Case Field Flags</i>	1-24
1.17	CONFIGURING EXECUTION FIELDS.....	1-25
1.17.1	<i>Execution Field Editor</i>	1-26
1.17.2	<i>Execution Field Attributes</i>	1-26
1.17.3	<i>Special Execution Fields</i>	1-27
1.17.4	<i>Execution Field Types</i>	1-29
1.17.5	<i>Execution Field Styles</i>	1-32
1.17.5.1	Date Field Styles	1-32
1.17.5.2	Textarea Field Styles	1-32
1.17.6	<i>Execution Field Flags</i>	1-33
1.18	CONFIGURING TEMPLATES.....	1-35
1.18.1	<i>Missing Field Errors in Templates</i>	1-36
1.18.2	<i>Check Templates for Errors</i>	1-37
1.18.3	<i>Report Templates</i>	1-37
1.18.4	<i>Templated Test Case Reports</i>	1-38
1.18.5	<i>Templated Requirements Reports</i>	1-38
1.18.6	<i>Editing and Execution Templates</i>	1-39
1.18.7	<i>Template Editor</i>	1-39
1.18.8	<i>Table manipulation</i>	1-39
1.18.8.1	Tables with Header/Footer Elements	1-42
1.18.9	<i>Field Names in Templates</i>	1-44
1.19	TIME TRACKING FIELDS	1-44
1.20	CONFIGURING RESULT CODES	1-46
1.21	CONFIGURING SESSION VARIABLES.....	1-48
1.21.1	<i>Session Variable Editor</i>	1-48
1.21.2	<i>Session Variable Attributes</i>	1-49
1.21.3	<i>Session Variable Types</i>	1-50
1.21.4	<i>Session Variable Flags</i>	1-53
2	ADMINISTRATION.....	2-1
2.1	IMPORTANT ADMINISTRATIVE FEATURES	2-2
2.1.1	<i>Make sure you do not run out of disk space</i>	2-2
2.1.2	<i>Email Notifications</i>	2-2
2.1.3	<i>Timezones</i>	2-2
2.1.4	<i>Time styles</i>	2-2
2.2	LDAP ACCOUNTS.....	2-2
2.3	NORMAL USER ADMINISTRATIVE FEATURES.....	2-4
2.3.1	<i>Account Creation</i>	2-4
2.3.2	<i>Account Modification</i>	2-5
2.4	ADMINISTRATOR ADMINISTRATIVE FEATURES	2-6
2.4.1	<i>Account Management</i>	2-7

2.4.1.1	Restricting Test Suite Access.....	2-8
2.4.1.2	Email Notifications.....	2-10
2.4.2	<i>Check for Updates to ApTest Manager</i>	2-10
2.4.3	<i>Profile Catalog Management</i>	2-11
2.4.4	<i>System Configuration</i>	2-11
2.4.4.1	Should ApTest Manager be closed to non-administrative users?	2-11
2.4.4.2	What message should be displayed on the login page?	2-12
2.4.4.3	Bug Tracking Integration	2-12
2.4.4.4	What is the default test suite access mode?	2-13
2.4.4.5	Do you want to send runtime error notification to ApTest?	2-14
2.4.4.6	What is the URI of your HTTP proxy server?	2-14
2.4.4.7	How many days before inactive logins are removed?	2-14
2.4.4.8	What is the default time style for your users?	2-14
2.4.4.9	What is the default time zone for your users?	2-14
2.4.4.10	Should users have to relogin after they close their browser?	2-15
2.4.4.11	Should the test execution result, notes, and time fields be pre-populated with the last information entered?	2-15
2.4.4.12	What is that maximum number of Session Variables that will be displayed horizontally?	2-15
2.4.4.13	What is the maximum number of sessions that can be displayed on the Run Tests and Select Report pages?	2-15
2.4.4.14	What is the authentication token for users of the RPC interface?	2-15
2.4.4.15	Should the year be included when displaying dates that are in the current year?	2-15
2.4.4.16	How long in seconds should transactions wait for a lock before aborting (>60!) ?	2-16
2.4.4.17	Should a user be prompted if a folder, test case, or requirement name they enter would be automatically changed to avoid the use of non-portable characters?	2-16
2.4.4.18	Should tables be permitted in wysiwyg fields?	2-16
2.4.4.19	Normal User Capabilities	2-16
2.4.4.20	How many rows should be displayed per page in large reports?	2-17
2.4.4.21	Email Settings.....	2-18
2.4.5	<i>Manage LDAP Configuration</i>	2-18
2.4.6	<i>Manage SQL Configuration</i>	2-21
2.4.7	<i>View Login and License Information</i>	2-21
2.4.8	<i>View Test Suite Information</i>	2-21
2.4.9	<i>Archive System Data</i>	2-22
2.4.10	<i>Test Email Configuration</i>	2-22

3 ADVANCED TOPICS..... 3-1

3.1	DATABASE.....	3-1
3.2	REVISION CONTROL INTEGRATION.....	3-1
3.3	REQUIREMENT AND TEST CASE FILES	3-1
3.3.1	<i>Copying Tests and Requirements between Test Suites</i>	3-4
3.3.2	<i>Registering Changes</i>	3-4
3.4	EXPORTING SUITES	3-4
3.5	BACKING UP APTEST MANAGER FILES	3-5
3.6	MOVING TO A NEW SERVER	3-5
3.7	USER EXTENSIONS	3-6

3.7.1	<i>Javascript Extensions</i>	3-6
3.7.2	<i>CSS Extensions</i>	3-7
3.7.3	<i>Adding a Logo to Reports</i>	3-7
3.8	SCRIPTS	3-7
4	SQL DATASTORE	4-1
4.1	SUPPORTED DATABASES.....	4-1
4.1.1	<i>Database installation</i>	4-1
4.2	DATASTORE OPERATION	4-2
4.3	SQL DATASTORE CONFIGURATION	4-2
4.3.1	<i>Global SQL Datastore Configuration</i>	4-3
4.3.2	<i>Per Test Suite SQL Datastore Configuration</i>	4-3
4.4	INITIALIZING SQL DATASTORES	4-4
4.4.1	<i>Initializing SQL Datastores for multiple Test Suites</i>	4-4
4.4.2	<i>Initializing a single Test Suite's SQL Datastore</i>	4-5
4.5	DATASTORE LAYOUT	4-5
4.5.1	<i>Tables related to Requirements</i>	4-5
4.5.2	<i>Tables related to Test Cases</i>	4-6
4.5.3	<i>Tables related to Test Sessions</i>	4-7
4.5.4	<i>Tables related to Test Session Results</i>	4-8
4.5.5	<i>Tables related to User IDs</i>	4-9
4.5.6	<i>Table related to Test Suite Information</i>	4-9
4.5.7	<i>Notes</i>	4-10

Table of Figures

Figure 1 – Manage Test Suite screen	1-1
Figure 2 - Edit Test Case Fields screen.....	1-11
Figure 3 - Edit Execution Fields screen.....	1-27
Figure 4 - Display of Template with Missing Field	1-37
Figure 5 - Edit Report Templates screen	1-38
Figure 6 - Edit Template screen	1-41
Figure 7 - Example Header/Footer table.....	1-43
Figure 8 - Edit Result Codes screen.....	1-48
Figure 9 - Edit Session Variables screen.....	1-50
Figure 10 - Click Username to Access Administration	2-1
Figure 11 – Normal User’s Create Account screen with LDAP enabled	2-4
Figure 12 - Normal User’s Edit Account screen	2-6
Figure 13 - Administrator’s Management screen.....	2-7
Figure 14 - Administrator’s Change Existing Account screen.....	2-9
Figure 15 - Administrator’s Change Suite Access Permissions screen.....	2-10
Figure 16 – Manage LDAP Server Configuration screen	2-20
Figure 17 - Test Case File.....	3-3

PREFACE

MANUAL ORGANIZATION

This manual is divided into four chapters that present advanced features of ApTest Manager.

Chapter 1 – Managing Test Suites

Configuring ApTest Manager Test Suites.

Chapter 2 – Administration

ApTest Manager Administrative features.

Chapter 3 – Advanced Topics

Additional ApTest Manager features and functions.

Chapter 4 – SQL Datastores

Mirroring data in relational databases.

DOCUMENTATION SET

The *ApTest Manager User Guide* presents additional information on ApTest Manager. Review of the User Guide is suggested before reading this document.

Chapter 1 – Introduction

An overview of the features and benefits of ApTest Manager.

Chapter 2 – Using ApTest Manager

Managing testing with ApTest Manager.

Chapter 3 – Defining Tests

Defining test requirements, specifications, and procedures with ApTest Manager.

Chapter 4 – Running Tests

Executing ApTest Manager Test Sessions.

Chapter 5 – Viewing Reports

Viewing ApTest Manager test reports.

Chapter 6 – Usage Scenarios

Examples of using ApTest Manager to solve some common problems

STYLISTIC CONVENTIONS

Italics indicate important references, placeholders, and command line variables.

Boldface indicates emphasis. Boldfaced text is used to draw attention to active menu selections or hypertext links.

`Courier` type represents examples of computer-generated output, code samples, or a typed command line entry.

The paired hyphen and ‘greater than’ characters (->) denote separate elements of a mouse command sequence when moving through a series of menus.

Brackets [] are used to enclose optional items in a typed entry. Enter only the information within the brackets, and not the brackets themselves. Alternately, brackets are used to identify a bracketed menu item or a key on the keyboard (e.g., the Escape key is expressed as [Esc]).

Braces { } are used to enclose required items in a typed entry. Enter only the information within the braces, and not the braces themselves.

Representations of graphical user interface elements, such as the browser’s “back” button, are displayed graphically (i.e.  Back).

CUSTOMIZATION

ApTest Manager is template driven, allowing it to be customized to match existing test processes and procedures. Thus, an organization gains the benefits of improved management of its testing process without having to modify that process or adopt a new methodology.

In this Guide examples are based on templates derived from the IEEE 829 standard for test documentation. When working with a Test Suite that is based on different templates some screens may be different from those in this Guide.

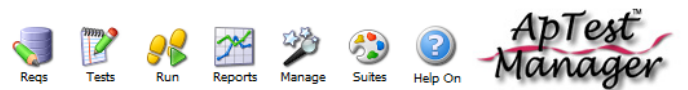
Also, ApTest Manager can be configured to limit access to some of its features to users with different levels of privileges. Thus the functionality shown in this Guide may not be available to all ApTest Manager users.

1 MANAGING TEST SUITES

Test Suite administrative functions are accessed from the Manage Test Suite screen. Click the Manage icon on the ApTest Manager Menu Bar to get to this area of ApTest Manager. A user must have Suite Manager access to a Test Suite in order to access these functions. Otherwise the icon will show a red X and the user will not be able to access this screen.

The Manage Test Suite screen allows the user to change the Test Suite's configuration and perform Test Suite administration operations.

Manage Test Suite



Administrator: [Bob Manager logout](#)

ApTest_Manager_Sample [Suite Manager]

Make changes to your test suite's configuration:

- | | |
|---|--|
| Edit Requirement Fields | Edit Requirement Editing Template |
| Edit Test Case Fields | Edit Test Case Editing Template |
| Edit Execution Fields | Edit Single Test Case Execution Template |
| Edit Session Variables | Edit Multiple Test Case Execution Template |
| Edit Result Codes | Edit Report Templates |
| | Check Templates for Errors |

Perform high level test suite management operations:

- | | |
|---|---|
| Change the Test Suite's Description | Create Requirements from Test Cases |
| Copy the Test Suite | Create Test Cases from Requirements |
| Rename the Test Suite | Manage the Test Suite's SQL Datastore |
| Delete the Test Suite | Synchronize the Test Suite |
| Archive the Test Suite | Cleanup hidden deleted files |

Copyright © 2000-2010 Applied Testing and Technology, Inc. All rights reserved.

Figure 1 – Manage Test Suite screen

Aspects of Test Suite administration may also be restricted to users with administrative privilege through the Manage System Configuration screen.

MANAGING TEST SUITES

1.1 CHANGE THE TEST SUITE'S DESCRIPTION

The description defined when the current Test Suite was created can be modified. Click **Make Changes** after altering the description.

1.2 COPY THE TEST SUITE

An existing Test Suite can be copied, as an alternative to creating a completely new Test Suite. The Test Cases which make up the Test Suite are copied along with users' access permissions. Test Sets, Test Sessions, Test Session results, and access permissions may optionally also be copied.

If SQL Support is enabled on a per suite basis the Suite created by the copy operation does not have SQL support enabled for it after the copy.

1.3 RENAME THE TEST SUITE

An existing Test Suite can be renamed. The contents of the Test Suite are unchanged.

1.4 DELETE THE TEST SUITE

The current Test Suite can be deleted. This should be done with careful consideration as all the information associated with the Test Suite is permanently lost.

This feature may be restricted to users with administrative privilege through the Manage System Configuration screen.

1.5 ARCHIVE THE TEST SUITE

Create a portable zip-format file of all the test cases, requirements, results, templates, and field definitions for the current test suite. This archive is suitable for use as a backup or for use when importing into another ApTest Manager server.

1.6 RENUMBER THE AUTO-NUMBERED TESTS/REQUIREMENTS

A link is displayed to allow renumbering to be performed if an ID style of auto-number by suite or by folder is configured for Requirements or Tests Cases. Renumbering may be desirable if Requirements or Test Cases have been reordered (see the *User's Guide* for details). The renumbering process numbers the Requirements/Test Cases sequentially based on their current order within their Folders.

These features can also be used to transform the Requirements//Test Cases in a suite from user defined ID strings to autonumbered IDs.

To save the existing ID strings during the renumbering process, first create a Requirement/Test Case Field of type Text with an appropriate size to hold the ID strings and the Name part flag set. This allows existing Test Suites with ID strings for Test Cases to be converted to auto-numbered Test Cases by:

- Changing the Test Case ID style to auto number (by suite or by folder)
- Selecting **Renumber the auto-numbered tests**, mapping the ID to a Field with the Name Part attribute set (see Section 1.16.6) (this option will only be available if there are Requirements/Test Cases with non-numeric IDs).

The resulting Requirement/Test Case tree contains autonumbered tests that also have the original string IDs along with the numeric IDs as part of the Requirement/Test Case names. After renumbering auto-outline style can be selected if desired.

1.7 ENABLE REQUIREMENTS

For Suites that do not have one, enables a separate Requirements tree. The user is asked to specify a Test Suite Profile that contains Requirements Field definitions; these are installed as the Requirements Field definitions for this Suite. To automatically populate the Requirements tree see the following Section. Note that to be able to link Test Cases to Requirements the *atm-reqlink* Field needs to be added to the Test Case Editing template.

1.8 CREATE REQUIREMENTS FROM TEST CASES

For Suites that have a separate Requirements tree, automatically creates Requirements for Test Case with no associated Requirement(s). This includes associated Requirements that have been deleted after being moved to the Trash. This can be used for example to populate the Requirements tree for a Suite that has not had one (after enabling Requirements as per Section 1.7).

The user can optionally specify additional criteria for the Test Cases for which to create Requirements.

Each new Requirement is linked to the Test Case for which it was created. To see this linkage for a Test Case the Test Case Field named *atm_reqlink* can be inserted into a template; to see it for a Requirement the Requirement Field *atm_tests* can be inserted into a template (see the Admin Guide for details). Values of Test Case Fields may also be optionally copied to Fields in the Requirements created.

The IDs of the created Requirements depend on the ID styles of Test Cases and Requirements:

- If the ID style for Requirements is numeric the IDs of the Requirements created are assigned sequentially. If the ID style for Test Cases is plain the Test Case ID is included in the Fields that can be mapped to Requirement Fields (as the ID value would otherwise be lost).
- If the ID style for Requirements and Test Cases is plain the IDs of the Requirements created are the IDs of the corresponding Test Cases.

- If the ID style for Test Cases is numeric and the ID style for Requirements is plain the IDs of the Requirements created are the names of the corresponding Test Cases (contents of the ID and name-part Fields, separated by underscores).

If necessary, new Folders are added to the Requirements tree to contain the new Requirements.

If a Modification history table is configured for Requirements in the Test Suite an entry is placed in the each Requirement created indicating the Requirement was created to cover the corresponding Test Case (identified by its name at the time).

Note that references to uploaded files are relative to a specific tree when selected from the file browser and thus mapping a Field of file links will likely not result in working links in the copy.

1.9 CREATE TEST CASES FROM REQUIREMENTS

For Suites that have a separate Requirements tree, automatically creates Test Cases for Requirements with no associated Test Case(s). This includes associated Test Cases that have been deleted after being moved to the Trash. The user can optionally specify additional criteria for the Requirements for which to create tests.

Each new Test Case is linked to the Requirement for which it was created. To see this linkage for a Test Case the Test Case Field named atm_reqlink can be inserted into a template; to see it for a Requirement the Requirement Field atm_tests can be inserted into a template (see the Admin Guide for details). Values of Requirement Fields may also be optionally copied to Fields in the Test Cases created.

The IDs of the created Test Cases depend on the ID Field styles of Test Cases and Requirements:

- If the ID style for Test Cases is numeric then the IDs of the Test Cases created are assigned sequentially. If the ID style for Requirements is plain the Requirement ID is included in the Fields that can be mapped to Test Case Fields (as the ID value would otherwise be lost).
- If the ID style for Requirements and Test Cases is plain then the IDs of the Test Cases created are the IDs of the corresponding Requirements.
- If the ID style for Requirements is numeric and the ID style for Test Cases is plain the IDs of the Test Cases created are the names of the corresponding Requirements (contents of the ID and name-part Fields, separated by underscores).

If necessary, new Folders are added to the Test Case tree to contain the new Test Cases. If 'A test case per requirement' is specified Folders are created so a new Test Case has the same folder path in the Test Case tree as the corresponding Requirement does in the Requirements tree. If 'A folder and test case per requirement' is specified an additional Folder is created with the same ID as the new Test Case. If the ID style is numeric than the Folder name will also include any Name part Fields.

MANAGING TEST SUITES

If a Modification history table is configured for Test Cases in the Test Suite an entry is placed in the each Test Case created indicating the Test Case was created to cover the corresponding Requirement (identified by its name at the time).

Note that references to uploaded files are relative to a specific tree when selected from the file browser and thus mapping a Field of file links will likely not result in working links in the copy.

1.10 MANAGE THE TEST SUITE'S SQL DATASTORE

If SQL Support is enabled for the Test Suite (see Section 4.3) this link is displayed to provide management operations for it. See Chapter 4 for descriptions of these operations.

1.11 SYNCHRONIZE THE TEST SUITE

This function updates ApTest Manager's internal database for the Test Suite to match the contents of its Requirement and Test Case files (see Section 3.3 for details on these files).

1.12 CLEANUP HIDDEN DELETED FILES

This function will examine the test suite and remove any Test Cases or Requirements that have been deleted and are not referenced. You may also choose to remove all hidden, deleted files even if they are referenced.

1.13 IMPORT FROM ANOTHER SUITE

Through this function, you can import test cases or requirements from another test suite on the same server. When doing so, you will be prompted to select what type of thing you are importing, where you are importing from, how the fields from the source suite should map into the current suite, what should be imported, and where in the current suite the new items should be placed.

Some things to keep in mind when importing data from another suite:

- The tool permits you to map any field to any other field. If fields are dissimilar (e.g., importing from a textarea field to a menu field) this will likely result in a warning because the data will not map in a sensible way.
- If you are attempting to duplicate the structure of one suite in another suite, it is best to set the import target as the top level of the current suite so that the system will automatically create the folders with the same structure.
- When importing tests or requirements that have dissimilar naming styles (e.g., plain in the source suite vs. auto-number by folder in the target suite) the system will do its best to make sensible names.

- When importing and merging data, the IDs of the objects being imported need to be exactly same in the source and the target in order for any merging to take place. If the names do not match, then a new object will be created in the target suite.

We recommend that you experiment with importing using a sample suite – create a copy of the suite you want to import into, and work in that copy until you determine the best model that works for migrating data among your test suites.

1.14 TEST SUITE CONFIGURATION

A key feature of ApTest Manager is the ability to customize the information that defines a Test Suite and how this information is displayed. Requirement and Test Case Fields can be added or deleted and Field characteristics such as title, format, style, and placement can be specified. As well, both reports and data entry screens may be customized to display information in unique styles, and the results that can be assigned to Test Cases can be changed.

This configurability allows ApTest Manager to be adapted to fit easily into an organization's QA process, and for different test specifications and procedures to be used for different products, all under the ApTest Manager umbrella.

When a new Test Suite is created its initial configuration is selected from a catalog of predefined Profiles. After the Test Suite has been created its configuration can be customized further.

Modifying a Test Suite's configuration affects only that Test Suite, not the Profile it was created from. The configuration of a Test Suite can however be added to the Profile catalog, allowing this configuration to be used when creating new Test Suites in the future. Similarly, changing a Profile does not impact existing Test Suites, only ones created in the future. See Chapter 2 for details on administration of the Profile catalog.

Test Suite configuration changes must be made very carefully as incorrect configuration can damage a Test Suite. A complete review of this Chapter and a gentle touch are recommended.

Ideally, a Test Suite's configuration should be defined before Requirements and Test Cases are entered. This ensures complete information is entered for each Requirement and Test Case and avoids the possibility of introducing incompatible configuration formats after the fact.

1.14.1 CONFIGURABLE ELEMENTS

The configurable aspects of a Test Suite are:

- Fields that make up Requirements and Test Cases. This includes both a description of the information and how it is displayed when edited, e.g. as a text field, menu, etc. Examples of the sort of information a Field may contain are the author, creation date, and type of a Test Case or Requirement and Problem Reports submitted for the execution of a Test Case.
 - Requirement Fields define the information which makes up a Test Suite's Requirements. These Fields are managed with the Requirement Field Editor (see Section 1.16).

MANAGING TEST SUITES

- Test Case Fields define the information which makes up a Test Suite's Test Cases. These Fields are managed with the Test Case Field Editor (see Section 1.16).
- Execution Fields are similar to the Fields for a Test Case but apply to Test Sessions rather than Test Cases. They allow custom information to be collected at execution time through Test Case Execution templates. This information is associated with a particular run of a Test Case rather than the Test Case itself. These Fields are managed with the Execution Field Editor (see Section 1.17).
- Templates define how Requirement, Test Case, and Execution Fields are formatted in ApTest Manager reports and screens: which Fields are displayed and how they are laid out. Fields must be included in templates in order to be displayed.

Separate Templates are provided for the screens used to edit requirements, edit and execute tests, as well as for an unlimited number of different reports. This allows the user to alter the display and formatting of Requirement, Test Case, and Session information separately for different areas of ApTest Manager. These Fields are managed with Template Editors (see Section 1.18).

- Test Case results define the display name and display color for the possible results of a Test Case. ApTest Manager does not attach any particular significance to a specific result and any number of possible results can be defined by the user. These Fields are managed with the Results Editor (see Section 1.20).
- Test Session Variables define information about the test environment for a Test Session. Their configuration includes both a description of the information and how it is displayed, e.g. as a text field, menu, etc. Examples of the sort of information Session Variables may contain are the platform, OS, software, and hardware used to execute a Session. These Fields are managed with the Session Variable Editor (see Section 1.21).

1.14.2 CONFIGURATION EDITORS

Field Editors utilize a variant of the table Field, controlling a table of items with its standard row controls. Fields can be added, copied, and deleted. Fields can be rearranged, so they are presented in a particular order. Attributes can be specified for each Field. The information entered is checked for consistency when changes are made as well as when they are saved.

To add a Field click a  or  icon to add a new row to the Editor, and then enter the attributes for the Field, as described below, into the new row. New Fields will be available immediately in new and existing Requirements/Test Cases (though will have no value in existing Requirements/Test Cases).

To delete a Field click the  icon for the Field's row in the editor.

The attributes of existing Fields can be modified as desired. Changes will immediately apply to new and existing Requirements/Test Cases.

MANAGING TEST SUITES

Click the  or  icon for a Field's row in the editor to change its location.

Click the  icon for a Field's row in the editor to create a copy of the Field.

This interface is also used by the Session Variable Editor.

Template Editors utilize a variant of the wysiwyg text field editor with additional controls for developing tables.

1.15 CONSIDERATIONS FOR TEST SUITE DESIGN

ApTest Manager is a very flexible product with a rich set of features. It is intended to be adaptable for use in many different test processes and is highly configurable. This configurability allows ApTest Manager to be used in many different ways. The following are some issues to consider when configuring ApTest Manager.

1.15.1 DEFINING REQUIREMENTS AND TEST CASE FIELDS

Test Suites may be configured to have any number and type of Fields in Requirements and Test Cases. ApTest Manager contains several sample sets of Requirement and Test Case definitions based on the IEEE 829 standard for software test documentation. These include a variety of menu, text, and table Fields.

Suites can use one of these sets of definitions or an organization can modify a sample to fit its needs. For example by changing the range of version numbers in the version number Field to match its products' versions, adding a Field that specifies the hardware configurations a test applies to, or replacing the sample with a unique set of Fields that match the organization's specific requirements.

Defining a Requirement or Test Case entails specifying values for each Field: the product versions it applies to, inputs, outputs, etc.

The specification of Requirements is optional.

Requirements may be defined and Test Cases can be linked to Requirements that they fulfill, in one-to-one, many-to-one, and one-to-many arrangements, chosen when a Test Case is defined. Alternatively Test Cases may simply be defined on their own, with requirements not defined within ApTest Manager, instead maintained externally or simply not used at all. The ApTest Manager Requirements area is available for Test Suites created with some Profiles and not others. Test Suites without Requirements support can be converted to use Requirements if desired.

1.15.2 DEFINING TEST CASE SELECTORS

An important element in Test Suite configuration is identifying which Test Case Fields are "selectors". Selector Fields have a special role in picking groups of tests that form Test Sets.

When creating a Test Set, selector Fields are used to specify a subset of the tests in a Test Suite. Field values may be specified for each selector and those values must be present in a Test Case in order for it to be included in the Test Set. For example Test Cases in a specific language or that apply to a particular release or type of test cycle. It is safe to change which Fields are configured as selectors at any time.

1.15.3 DEFINING REQUIREMENTS AND TEST CASES

When a Requirement or Test Case is created it may be given a name, as are the Folders into which Requirements and Test Cases are grouped.

Picking the names for these Folders and Requirements/Test Cases should be done with some thought. Names should be long enough to identify what a Folder/Requirement/Test Case is testing. However, if names are too long they make reports difficult to read. Thus, it is advisable to design in advance what the Requirement/Test Case tree will look like and use Folder names which are not overly long. It is also a good idea not to have a tree too wide to display on users' screens.

By default, Requirements and Test Cases are displayed in alphabetic or numeric order and Folders are displayed in alphabetic order. A user defined order may be specified for the contents of each Folder if desired.

1.15.4 DEFINING TEST SETS

Test Sets identify subsets of the Test Cases in a Test Suite, possibly all the Test Cases but more often just some. This allows a large collection of tests to be contained in a Test Suite with different projects composed of different groups of tests used in different test runs.

The selectors for a Test Suite determine what Test Sets can be created. For example, to create Test Sets that contain just the tests from specific Folders, the ID Field needs to be a selector so Folders can be selected from it when creating a Test Set.

It is thus important to consider what Test Sets will be created when defining a Test Suite configuration. For example, to be able to have Test Sets for different product versions require a Test Case Field with which the product versions each Test Case applies to can be specified. This Field needs to be a selector.

Test Sets can be created at any time based on any combination of selector Field values: at the start of a test campaign or at any point during the campaign, for example to focus more testing on a specific product feature.

1.15.5 DEFINING EXECUTION FIELDS AND SESSION VARIABLES

In general, in formal testing systems each unique environment relevant to an implementation under test should be tested separately, and the aggregate of the results of such testing analyzed to give a view as to whether the implementation "passes" or not.

MANAGING TEST SUITES

In ApTest Manager, this is accomplished by creating a "Test Session" with Session Variables that describe each unique test environment, and executing all of the tests from a Test Set for each Session in its test environment.

Session Variables and Execution Fields are intended to capture information associated with the execution of a Test Set in a specific test environment. Example Session Variables are the OS, browser, hardware, and network on which a Session is run. Example Execution Fields are the Problem Reports created based on executing a Test Case.

Session Variables and Execution Fields are defined on a per Test Suite basis. Each Test Session has a set of Session Variable values for that Session, such as the OS, browser, and hardware it was run on. Variables and Execution Fields may be defined as menus with a list of predefined values to choose from or Fields into which text can be entered, and may have a variety of other attributes.

Session Variables are reported in ApTest Manager reports and can also be used to search for Sessions with specific Variable values, such as a list of all the Sessions run on a particular product version on a particular piece of hardware. Execution Fields are also reported in ApTest Manager reports and can be used to limit reports to Test Cases with specific Execution Field values, such as those which have had a PR filed for them.

It is thus important to create a meaningful collection of Session Variables for a Test Suite. Similarly, create custom Execution Fields for any special information testers will record when executing tests.

1.15.6 DEFINING TEST RESULTS

The possible result values a Test Case may be given when it is run can be configured. Different results can be used to associate a Test Case execution with different process states. For example: On Hold, Needs to be ReRun, Needs Bug Report Filed, etc.

ApTest Manager reports present charts and table showing the number of tests with each of the results defined. Information that does not need to be reported as a result, such as remarks by the user running the test, can be placed in the note field or an Execution Field associated with a run of a Test Case in a Session.

1.15.7 DEFINING TEMPLATES

Once the Fields and Variables for the Requirements, Test Cases, and Test Sessions in a Test Suite are defined, specify how they are displayed by defining templates for editing requirements, editing and running tests, as well as for reports. **Fields must be included in templates in order to be displayed.**

1.16 CONFIGURING REQUIREMENT AND TEST CASE FIELDS

Requirement and Test Case Fields are configured separately. Similar Field definitions and tools for editing them are used.

1.16.1 REQUIREMENT/TEST CASE FIELD EDITOR

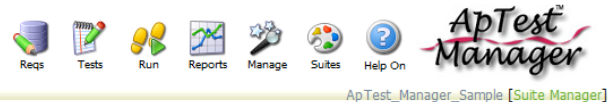
The editors for Requirement and Test Case Fields specify the elements making up Requirement and Test Case Fields, respectively, for a Test Suite.

The Fields shown in Figure 2 are examples from one of the Profiles in the catalog shipped with ApTest Manager.

Once defined, Requirement and Test Case Fields need to be included in one or more Test Suite templates in order to be displayed (see Section 1.18). If a Field is deleted it needs to be removed from all templates it is used in.

Except where a field is display-only, when editing a Requirement/Test Case a form element is presented for a Field in which to enter a value. Elsewhere, i.e. when displaying a Requirement/Test Case, the Field's value for the Requirement/Test Case is shown.

Edit Test Case Fields



Through this screen you can review and make changes to the fields contained in the Test Suite's Test Cases.

Fields must be included in a Template, defined separately, in order to be displayed.

Please review Chapter 1 of the ApTest Manager Admin Guide carefully before modifying this information. Press "Save Changes" when you are done.

[Save Changes](#) [Cancel](#)

	Name	Label	Type	Size	Style	Values	Flags
	id	ID	id	20x1	plain	n/a	selectable graphable
	cdate	Creation Date	creation date	n/a	n/a	n/a	selectable
	mdate	Last Modification Time	last modification date	n/a	n/a	n/a	selectable
	muser	Last Modified By	last modifying user	n/a	n/a	n/a	selectable graphable
	atm_owner	Assigned Test Developer	userlist	10	n/a	n/a	selectable graphable
	author	Created By	creator	n/a	n/a	n/a	selectable graphable

Figure 2 - Edit Test Case Fields screen

1.16.2 REQUIREMENT AND TEST CASE FIELD ATTRIBUTES

Each Test Case Field has seven attributes:

MANAGING TEST SUITES

NAME	The name of the Field. Field names may only contain alphanumeric characters as well as the period (.), hyphen (-) and underscore (_) characters, and are case-insensitive. Field names beginning with the string "atm_" are reserved for use by ApTest Manager and are not available for use in Fields defined by users. Also, names that begin with RQT, FIELD, or RESULT may not be used.
LABEL	Defines the text label that is used when presenting this Field.
TYPE	Defines the type of data that can appear in this Field.
SIZE	Describes the size of the form element used for this Field when editing.
STYLE	Defines how the contents of the Field are formatted when displayed.
VALUES	Defines default Field values.
FLAGS	Indicates properties of the Field. Multiple flags may be set per Field.

1.16.3 SPECIAL REQUIREMENT AND TEST CASE FIELDS

Some Fields for Requirements and Test Cases are required and must be present. The delete icon for a row in a Field editor is grayed if the row's contents are required.

There may be no more than one instance for Requirements and/or for Test Cases of some Fields. The copy icon for a row in a Field editor is grayed out if only one instance of its Field is allowed.

The following Fields are restricted (these Fields also have restrictions on which of their attributes may be edited).

atm_average_result	Requirements Only. Used in report templates to indicate the best result for associated Test Cases executed in multiple Sessions. This Field is required, is a display only field, and there may not be more than one such Field.
atm_best_result	Requirements Only. Used in report templates to indicate the average result for associated Test Cases executed in multiple Sessions. This Field is required, is a display only field, and there may not be more than one such Field.
atm_locked	Indicates in a Requirement/Test Case has been locked. This Field is required and there may be not be more than one such Field for Requirements and more than one for Test Cases. This is a display only Field – the user must lock/unlock the Requirement/Test Case via the GUI to modify this information.
atm_reqcount	Test Cases only. The number of Requirements a Test Case is linked to. This Field may not be edited, is required, and there may be not be more than one such Field. This is a display only

MANAGING TEST SUITES

Field - the user must change the Requirements that a Test Cases is linked to in order to modify this information.

atm_reqlink	Test Cases only. The Requirements a Test Case is linked to. When specifying a value for this field a tree of defined Requirements is displayed from which one or more Requirements can be selected. This Field is required and there may be not be more than one such Field. This Field may be marked as selectable and thus have a selector summary in reports, however it is not available as a selector in Customize Report or Define Test Set.
atm_testcount	Requirements only. The number of Test Cases linked to a Requirement. This Field may not be edited, is required, and there may be not be more than one such Field. This is a display only Field - the user must change the Test Cases linking to a Requirement to modify this information.
atm_tests	Requirements only. The Test Cases linked to a Requirement. When specifying a value for this field a tree of defined Test Cases is displayed from which one or more Test Cases can be selected. This Field is required and there may be not be more than one such Field. This Field may be marked as selectable and thus have a selector summary in Requirements reports, however it is not available as a selector in Customize Report or Define Test Set..
atm_worst_result	Requirements Only. Used in report templates to indicate the average result for associated Test Cases executed in multiple Sessions. This Field is required, is a display only field, and there may not be more than one such Field.
author	The author of the Requirement/Test Case. This Field is required and there may be not be more than one such Field for Requirements and more than one for Test Cases. This is a display only Field - this information may not be modified by the user.
cdate	The creation date of the Requirement/Test Case. This Field is required and there may be not be more than one such Field for Requirements and more than one for Test Cases. This is a display only Field - this information may not be modified by the user.
ID	The ID of the Requirement/Test Case. This Field is required and there may be not be more than one such Field for Requirements and more than one for Test Cases. This is a display only Field - this information may not be modified by the user (except for example by renaming the Requirement/Test Case). The ID may be a string entered by the user or a number automatically assigned by ApTest Manager, depending on the ID style configured (see Section 1.16.5.1). In many areas of ApTest Manager the ID is prefaced by the folders which contain the Requirement/Test Case, e.g. Billing/Data Entry/56 Invoice Entry, identifying a Test Case

contained within the Data entry Sub Folder in a Billing Folder that is the 56th Test Case and deals with testing the entry of invoices.

Optionally, other Fields for the Requirements/Test Cases may be configured with the 'Name part' attribute (see Section 1.16.6), meaning their value is added to the ID to form the Requirement/Test Case name.

For example, if a Name part Field called "summary" is configured for Requirements and auto numbering by suite is configured for the Requirements ID style, a requirement would have a name that was its auto-assigned number followed by the value of its summary field, e.g. "57 Invoice number must support 3 digits". Name part Fields are marked with a * (green asterisk) when editing a Requirement/Test Case.

ID and name part Field values in names are separated by spaces and are combined in the order the Fields are specified in the Requirement/Test Case Field editor.

SIZE specifies the height of a menu of Folders displayed for this Field on a query screen, e.g. on the Customize Report or Define Test Set screens (the *ApTest Manager User Guide* for details).

SIZE may also be of the form YYxZZ, in which case the depth of the Folders displayed is limited to ZZ. For example with a Folder tree AA/BB/CC setting SIZE to 5x2 would cause AA/BB to be included in the menu but AA/BB/CC would not be. This can be useful to limit the menu size for a deep tree of Folders. Normally selecting a Folder from this menu does NOT automatically select any child Folders (i.e. all the Folders in a hierarchy need to be selected individually). Folders below the depth limit however are selected if their parent Folder is.

STYLE specifies the numbering style used for Requirement/Test Case IDs (see Section 1.16.5.1).

mdate

The date of the last modification of the Requirement/Test Case (through the GUI). When a Requirement/Test Case is created, this value is initialized to the creation date. This Field cannot be edited. This Field is required and there may be not be more than one such Field for Requirements and more than one for Test Cases. This is a display only Field – this information may not be modified by the user.

muser

The user who last modified the Requirement/Test Case (through the GUI). When a Requirement/Test Case is created, this value is initialized to the author of the Requirement/Test Case. This Field is required and there may be not be more than one such Field for Requirements and more than one for Test Cases.

1.16.3.1 RESERVED FIELD NAMES

Reserved Requirement/Test Case Fields trigger special ApTest Manager behavior. The names of these Fields begin with atm_, and thus are reserved.

atm_mcomments	This Field must be of type textarea. It may have any valid size or style for a textarea Field. This field is not required but there may not be more than one such Field for Requirements and more than one for Test Cases. This Field is intended to be included in a modification history table field. If it is present comments are entered into the Field automatically when a Requirement/Test Case is copied or renamed. A comment is also entered into atm_mcomments if a Requirement/Test Case is imported, (unless “Merge the imported data with the existing Requirement/Test Case” is selected”).
atm_owner	This Field must be of type userlist. It may have any valid size or style for a Field of this type. This field is not required but there may not be more than one such Field for Requirements and more than one for Test Cases. This Field is intended to be used to capture the user(s) assigned to develop the Requirement/Test Case. When specifying the value of this Field (e.g. on the Create Test Set screen) a list of usernames is presented, composed of all current users with access permission to the current Test Suite of Manage Execution and Edit or better.
atm_prid	This reserved field can be used to track problem report IDs that are associated with the test case. If the VALUE attribute of this field is populated, it represents a template for a URI that can be used to connect to the issue tracking system and retrieve the issue. The pattern ‘%PRID’ is replaced with each problem report ID in the field, and the resulting string used as the target of a hypertext ‘A’ element (e.g. http://www.aptest.com/bugzilla/show_bug.cgi?id=%PRID).
atm_prlink	This reserved field can be used to hold links to problem reports in an associated issue tracking system, similar to atm_prid above. Each element in this field will be turned the target of an HTML ‘A’ element when displayed. If both the atm_prid and atm_prlink fields are used, their values will be combined when displayed.

1.16.4 REQUIREMENT/TEST CASE FIELD TYPES

Values for the type of a Requirement/Test Case Field that may be selected by the user and their significance are:

check boxes	When editing a Requirement/Test Case the values in the VALUES column are displayed as a set of check boxes. The layout of the check boxes depends on the value of SIZE. SIZE can be either AUTO, WIDTHxAUTO (e.g. 4xAUTO), or AUTOxHEIGHT (e.g. AUTOx4).
--------------------	---

If SIZE is AUTO, the boxes are output in a single paragraph. Each input and its label are separated by a non-breaking space, so the inputs and labels will not be separated if the paragraph wraps.

If SIZE is WIDTHxAUTO, the inputs will be displayed in a grid WIDTH cells wide. If the number of inputs is not an integer multiple of WIDTH, there will be empty spaces in the last row, at the right.

If SIZE is AUTOxHEIGHT, the inputs will be displayed in a grid HEIGHT rows high. If the number of inputs is not an integer multiple of HEIGHT, there will be empty spaces in the last column, at the bottom.

Default values (which are assigned to new Requirements/Test Cases as they are created) may be specified by placing an asterisk before a value in the VALUES column. Elsewhere the value of a check boxes Field is the values selected for the Field in the Requirement/Test Case.

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will only appear when editing a Requirement/Test Case if the value(s) were already in use. Obsolete items remain selectable for searching and reporting.

date

When editing a Requirement/Test Case a date value may be selected from a calendar. A default value may also be selected from a calendar. The default value may be cancelled by clicking the adjacent 'x'. Elsewhere the value of a date Field is a string representing the date selected for the Field in the Requirement/Test Case.

file

When editing a Requirement/Test Case a series of file names may be entered, separated by commas. These are turned into links:

Simple file names (e.g. file.doc) are made into a link to the file relative to the folder of the current Requirement/Test Case.

File names starting with ~/ (e.g. ~/file.doc) are made into a link to the file relative to the root directory of the current Test Suite.

File names starting with a / (e.g. /file.doc) are made into a link to the file relative to the root directory of the web server.

Full URLs (e.g. http://www.aptest.com) may also be entered and are turned into links to the URL.

Files can also be selected by browsing a list of previously uploaded file for the Requirements or Test Cases in this Test Suite.

SIZE must be a single numeric value > 1.

MANAGING TEST SUITES

Elsewhere the value of a file Field is links to the files entered for the Field in the Requirement/Test Case.

modification history table A Field type that supports integrating a modification history into a Requirement/Test Case. This history is shown as a table in last-first order. Each row consists of a set of user-specifiable Fields that should, but are not required to, include Fields of type muser and mdate. For example:

	history	Change History	modification history table	AUTO	n/a	mdate, muser, atm_mcomments	n/a
	mdate	Last Modification Time	last modification date	n/a	n/a	n/a	selectable
	muser	Last Modified By	last modifying user	n/a	n/a	n/a	selectable
	atm_mcomments	Change Details	textarea	Autox5	formatted		readonly mandatory selectable

When editing a Requirement/Test Case this Field is a similar to a Field of type table but has special semantics as follows:

- A new table row is automatically shown at the top of the modification history table and is the only row that contains user-editable Fields. If the user enters data into one or more of the editable fields in the top row of the table during an edit, the row's information is saved when the edit is saved. Otherwise the content of the row is not saved.
- Rows cannot be reordered - no table controls are available.

Table entries are generated automatically for some operations:

- When a Requirement/Test Case is copied its modification history is replaced with an automatically created entry recording the copying.
- When a Requirement/Test Case is renamed an entry is automatically added to the modification history recording the renaming.
- When a Requirement/Test Case is imported its modification history is replaced with an automatically created entry recording the import, unless the "Merge the imported data with the existing Requirement/Test Case" option is selected.

A modification history table Field cannot contain another table or modification history table Field. SIZE specifies the width of the table, valid values are nnn (pixels), nnn% (percent of available space), or AUTO (as wide as possible in the available space).

The modification history table is not intended to provide the functions of a full revision control system. Utilize ApTest Manager's ability to integrate with third party revision control systems if such a capability is needed.

multi-select menu When editing a Requirement/Test Case the values in the VALUES column are displayed as a multi-selection menu. If there is enough space on the page the menu will be placed adjacent to its label. Otherwise it will move onto the line below. SIZE specifies the number of values displayed at once.

Default values (which are assigned to new Requirements/Test Cases as they are created) may be specified by placing an asterisk before a value in the VALUES column.

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will only appear when editing a Requirement/Test Case if the value(s) were already in use. Obsolete items remain selectable for searching and reporting.

Menu fields can depend on the values of checkboxes or radio buttons Fields. The semantics are the same as depending on multi- and single-select menus, respectively. Elsewhere the value of a multi-select menu Field is the values selected for the Field in the Requirement/Test Case.

radio buttons When editing a Requirement/Test Case the values in the VALUES column are displayed as a set of radio buttons. The layout of the radio buttons depends on the value of SIZE.

SIZE can be either AUTO, WIDTHxAUTO (e.g. 4xAUTO), or AUTOxHEIGHT (e.g. AUTOx4).

If SIZE is AUTO, the buttons are output in a single paragraph. Each button and its label are separated by a non-breaking space, so the buttons and labels will not be separated if the paragraph wraps.

If SIZE is WIDTHxAUTO, the buttons will be displayed in a grid WIDTH cells wide. If the number of buttons is not an integer multiple of WIDTH, there will be empty spaces in the last row, at the right.

If SIZE is AUTOxHEIGHT, the buttons will be displayed in a grid HEIGHT rows high. If the number of buttons is not an integer multiple of HEIGHT, there will be empty spaces in the last column, at the bottom.

A default value (which is assigned to new Requirements/Test Cases as they are created) may be specified by placing an asterisk before one of the values in the VALUES column. Setting a second default value is silently ignored.

MANAGING TEST SUITES

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will only appear when editing a Requirement/Test Case if the value(s) were already in use. Obsolete items remain selectable for searching and reporting.

Elsewhere the value of a radio buttons Field is the values selected for the Field in the Requirement/Test Case.

single-select menu

When editing a Requirement/Test Case the values in the VALUES column are displayed as a single-selection menu. If there is enough space on the page the menu will be placed adjacent to its label. Otherwise it will move onto the line below. SIZE specifies the number of values displayed at once.

A default value (which is assigned to new Requirements/Test Cases as they are created) may be specified by placing an asterisk before one of the values in the VALUES column. If no default value is specified, the first value becomes the default.

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will only appear when editing a Requirement/Test Case if the value(s) were already in use. Obsolete items remain selectable for searching and reporting.

Menu fields can depend on the values of checkboxes or radio buttons Fields. The semantics are the same as depending on multi- and single-select menus, respectively. Elsewhere the value of a single-select menu Field is the value selected for the Field in the Requirement/Test Case.

To utilize the mandatory flag with a single-select menu Field and require the user to make a selection, define its first value to be empty. For example:

	priority	Priority	menu (single-select)	1	plain	*High, Medium, Low	readonly mandatory selectable graphable Depends On Name part
--	----------	----------	----------------------	---	-------	--------------------------	---

Otherwise the user is able to leave the Field with the default value selected, but without necessarily making a selection explicitly.

table

The VALUES column of this Field contains a comma-separated list of other Field names that make up the columns of a table. For example:

MANAGING TEST SUITES

	files	Associated Files	table	Auto	n/a	fileName, fileDesc	Add Result Add Note
	filename	Link to File	file	30	n/a		readonly mandatory selectable
	filedesc	File Description	textarea	25x2	formatted		readonly mandatory selectable

A field may not be included in more than one Field of type table. When editing a Requirement/Test Case, in addition to the columns specified in the VALUES column, a column is included in the table that contains insert, delete, move, and copy controls, allowing the table to be expanded, contracted, and reordered. A table Field cannot contain another table or modification history table Field. SIZE specifies the width of the table, valid values are nnn (pixels), nnn% (percent of available space), or AUTO (as wide as possible in the available space). Table height is determined by the number of rows in the table, where the height of a row is determined by the height of the fields contained in the table.

Elsewhere a table is displayed with the values entered into each column of the table in the Requirement/Test Case.

text

When editing a Requirement/Test Case an input field of type="text" is displayed and text may be entered. For this type of Field, SIZE specifies the width of the Field in characters when editing, and VALUES specifies the default value for the Field, set when a new Requirement/Test Case is created. Elsewhere the value of a text Field is the value for the Field in the Requirement/Test Case.

textarea

When editing a Requirement/Test Case an input field of type="textarea" is displayed and text may be entered. SIZE specifies the width and height of the Field when editing, using the form "WxH". The width may be a numeric value (in characters) or AUTO (e.g. AUTOx2) in which case the width of the Field automatically adjusts to 95% of the size of the containing element (e.g. a table a cell). VALUES specifies the default value for the Field, set when a new Requirement/Test Case is created. Elsewhere the value of a textarea Field is the value for the Field in the Requirement/Test Case.

userlist

When editing a Requirement/Test Case a menu of usernames is presented, composed of all current users with access permission to the current Test Suite of Run Assigned Only or greater. SIZE specifies the number of users displayed at once. Elsewhere the value of a userlist Field is the value(s) selected for the Field in the Requirement/Test Case. Note that user information is

stored in as the user's account name. Thus that is what needs to be supplied when doing a search and replace on a Field of this type.

1.16.5 REQUIREMENT AND TEST CASE FIELD STYLES

The style column value is not currently significant for the following Field types: author, cdate, file, mdate, modification history table, muser, table, and user.

1.16.5.1 ID FIELD STYLES

Depending on the Style selected the ID may be a numeric value assigned by ApTest Manager or a text string entered by the user. The style of ID is separately configured for Requirements and Test Cases in a Test Suite.

By default the plain style causes Requirements/Test Cases within a folder to be put in alphabetical order. A numeric style causes the default order for Requirements/Test Cases within a folder to be numeric. Requirements and Test Case can be reordered by the user (see the *ApTest Manager User Guide*).

The ID style configuration can be changed at any time. Existing IDs are not changed but IDs for newly created Requirements/Test Cases are created with the newly selected style.

Four ID styles are provided: a text style and 3 numeric styles.

Plain	The ID is a text string specified by the user when the Requirement/Test Case is created (it may be renamed later).
Auto number by suite	The ID is a number automatically assigned by ApTest Manager. The number is incremented for each Requirement/Test Case created in the Suite, starting at 1. For instance ID 133 is assigned to the 133rd Requirement/Test Case in a Test Suite.
Auto number by folder	The ID is a number automatically assigned by ApTest Manager. The number is incremented for each Requirement/Test Case created in each Folder, starting at 1. For instance ID 133 is assigned to the 133rd Requirement/Test Case in a Folder.
Auto outline number	The ID is a number dynamically assigned automatically by ApTest Manager. The number that is displayed is based on the location of the Requirement/Test Case and the Folder containing it. For instance ID 1.3.0.3 is displayed for the third Requirement/Test Case in the first sub-Folder in a Test Suite, contained in its third sub-Folder. An outline number is also associated with each Folder in an outline numbered tree. Note that Auto-numbered Requirements/Test Cases can be changed to Auto-outlined Requirements/Test Cases and vice versa by simply changing the ID style. This can be useful with features which are available only when Auto-numbering is used, such as renumbering.

1.16.5.2 MENU/TEXT FIELD STYLES

For menu and text Fields the style is generally set to “plain” though the `minutes1-minutes8`, `minutes24`, and `number` styles are also valid.

minutes1-minutes8	The value of the Field is expected to be an ordinal number and is rendered as hdm using a work day from 1 to 8 hours long, depending on the style selected. Values may also be specified as hours (e.g. 3h) or days (e.g. 5d). Fractional values are supported (e.g. 1.5h or 1.5d). Combinations of units (e.g. 1d3h) are not supported.
minutes24	The value of the Field is expected to be an ordinal number and is rendered as hdm using 24 hour days. Values may also be specified as hours (e.g. 3h) or days (e.g. 5d). Fractional values are supported (e.g. 1.5h or 1.5d). Combinations of units (e.g. 1d3h) are not supported.
number	If the Field is used in specifying the sort order for a report in Customize Report, sorting is in numeric order. Otherwise sorting is in alphabetical order.

1.16.5.3 DATE FIELD STYLES

For date Fields valid styles are:

date only	The value of the Field is displayed as a month, day, and year.
date and time	The value of the Field is displayed as the month, day, year, hour, minute, and time zone.

1.16.5.4 USERLIST FIELD STYLES

For userlist Fields valid styles are:

multi select	Multiple selections may be made from the list of users.
single select	A single user may be selected from the list of users.

1.16.5.5 TEXTAREA FIELD STYLES

ApTest Manager offers four styles of textarea Fields.

The contents of textarea Fields are stored as HTML, but many textarea Field styles allow this information to be entered by the user without using HTML directly. These styles format information in different ways when the Field is edited, viewed, or shown in a report.

A textarea Field's style is shown in parentheses after the name the Field in the Edit Requirement or Edit Test Case screen.

Supported textarea Field styles and the way they handle information are shown below. Some additional deprecated styles are supported for backward compatibility with earlier versions, but are not shown here.

code	HTML special characters are transformed into their general entity equivalents, newlines are placed at the end of every line. This retains exactly the line formatting the user enters and this style should be used when that is important. A downside of this style is that lines are not wrapped to take advantage of a larger window. Text is displayed in a monospace font. This style is used primarily for entering fixed format data such as source code into a Field.
formatted	This style allows text to be entered much like typing on a typewriter. Newlines can be used to end lines and paragraphs. Spaces or tabs can be used at the start of a line to indent text. Special characters are automatically transformed into HTML so they display correctly in the browser (e.g. '<' is transformed into "&lt;"). Plus, lists can be created automatically. Lines within the Field that begin with "#. " (the pound sign followed by a period and a space) are automatically numbered when the Requirement/Test Case is viewed or executed. This is a convenient mechanism for making a Requirement/Test Case easy to understand and maintain. Lines beginning with "nnn. " (a number followed by a period and a space) are included in the numbered list, with the specified number.
html	Contents are expected to include HTML. Special characters need to be specified in HTML and all formatting is specified with HTML directives. For example, newlines do not have any effect, the HTML
 or <P> directives need to be used instead or lines run together when the Field is displayed. Knowledge of HTML is required in order to use a Field of this style.
wysiwyg	A What You See Is What You Get HTML editor provides a Word Processor style interface for formatting information. This is the most commonly used textarea Field style. See the <i>ApTest Manager User Guide</i> for details. Contents are stored in HTML, transparently to the user. Note the height value in the SIZE attribute for this style of Field needs to allow for the display of several rows of icons.
wysiwyg2	Similar to wysiwyg above, but the field only becomes editable when the 'Open Editor' button is clicked. Suitable for use when there are many wysiwyg fields on the form and you want to conserve screen space or reduce page load time.

1.16.5.6 CHECK BOXES/RADIO BUTTONS FIELD STYLES

For check boxes and radio buttons Fields the style is generally set to “plain”, though number is also supported.

number If the Field is used in specifying the sort order for a report in Customize Report, sorting is in numeric order. Otherwise sorting is in alphabetical order.

1.16.6 REQUIREMENT AND TEST CASE FIELD FLAGS

In absence of a flag, the indicated functionality does not apply to the Field.

Add Result If selected, during execution a column will be appended to the table where the tester can specify a result for each table row. The final result for the test is automatically calculated, but can be overridden by the tester. (only valid on table fields)

Add Note If selected, during execution a column will be appended to the table where the tester can capture notes about that test step. Note that the characteristics of this note field are defined by the Execution Field ‘atm_step_notes’.

dependson The value of the flag should be another menu Field (single- or multi-select). Note that a Field that has other Fields dependent on it cannot have its name changed and must be selectable.

When editing a Requirement/Test Case, the values displayed for this Field depend on the value(s) of the other menu. Entries for values are separated by newlines and optional commas and have dependencies specified in braces after them. Please note these are braces and NOT parentheses. The contents of the braces are a comma separated list of dependencies that this value is displayed for. If two or more entries are included in the list for the same value, e.g.

1{A},
1{B}

the last value is used and previous entries are discarded.

Note that a multi-select menu may have no value selected (e.g. if no default value is specified), in which case any Field that depends on it will display only the values that have no dependencies specified. The behavior of circular dependencies (m1 depends on m2, m2 depends on m1) is not defined, but is almost certainly bad.

The dependencies are OR-ed. There is no AND and no NOT.

MANAGING TEST SUITES

	Name	Label	Type	Size	Style	Values	Flags
	m1	Menu1	menu (multi-select)	5	plain	*A, B, C, D	readonly mandatory selectable graphable dependson
	m2	Menu2	menu (multi-select)	5	plain	*0, 1(A), 2(B), 3(A,B), 4(B)	readonly mandatory selectable graphable dependson=m1

Multiple menu Fields may depend on a single menu Field. Dependent fields can be cascaded (i.e. a Field may both be dependent on another Field and have another Field dependent on it).

Menu Fields within a Table Field may be dependent only on another menu Field in that table Field.

If m1 == A, m2 will contain 0, 1, 2, and 3.

If m1 == B, m2 will contain 0, 3, and 4.

If m1 == C, m2 will contain 0.

If m1 == [A,B], m2 will contain 0, 1, 2, 3, and 4.

graphable

The values in the field may be used to graph “Results by Selector” in various reports.

mandatory

The Field must be given a value before saving edits to a Requirement/Test Case. Mandatory Fields are indicated with a red asterisk when editing a Requirement/Test Case.

Name part

The Field is combined with the ID Field to form the name of the Test Case/Requirement. This is particularly useful where the ID is numeric. Specifying Name part Fields allows mnemonic information to be combined with this numeric information. This flag may be set for single-select menu and text type Fields.

readonly

The value of the Field may not be changed. When a readonly Field is displayed for edit a form element is not displayed, rather the Field’s value is shown.

selectable

Indicates a) the Field is available for use in defining a Test Set b) reports can be sorted by the Field, and c) in reports that provide them a Results by Selector table/graph can be created for the Field (available only in specific reports for specific Field types).

1.17 CONFIGURING EXECUTION FIELDS

The Execution Field editor specifies Fields containing information about the execution of a Test Case in a Test Session.

1.17.1 EXECUTION FIELD EDITOR

The Fields shown in Figure 3 are part of one of the Profiles in the catalog shipped with ApTest Manager.

Once defined Execution Fields need to be included in one or more Test Suite templates in order to be displayed (see Section 1.18). If a Field is deleted it needs to be removed from all templates it is used in.

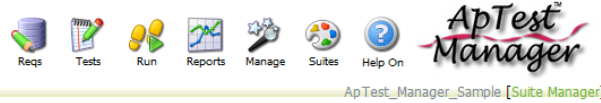
Except for display-only Fields, when a Field is included in an execution template a form element is presented when a test is executed that allows a value for the Field to be entered. When included in report templates the Field's value for a Test Case in a Test Session is shown.

1.17.2 EXECUTION FIELD ATTRIBUTES

Each Execution Field has seven attributes:

NAME	The name of the Field. Field names may only contain alphanumeric characters as well as the period (.), hyphen (-) and underscore (_) characters, and are case-insensitive. Note that names beginning with the string "atm_" are reserved for use by ApTest Manager and are not available for use in Fields defined by users. Also, names that begin with RQT, FIELD, or RESULT may not be used.
LABEL	Defines the text label that is used when presenting this Field.
SIZE	Describes the size of the form element used for this Field when executing a Test Case.
TYPE	Defines the type of data that can appear in this Field.
STYLE	Defines how the contents of the Field are formatted when displayed.
VALUES	Defines legal or default values for some Fields.
FLAGS	Indicates properties of the Field. Multiple flags may be set per Field.

Edit Execution Fields



User: [Bob Manager](#) [logout](#)
 Back to [Manage Suite](#)

ApTest_Manager_Sample [Suite Manager]

Through this screen you can review and make changes to this Test Suite's Execution Fields.

You may include Execution Fields in the Test Case Execution templates to gather custom information at Test Case execution time.

Please review Chapter 1 of the ApTest Manager Admin Guide carefully before modifying this information. Press "Save Changes" when you are done.

[Save Changes](#) [Cancel](#)

Name	Label	Type	Size	Style	Values	Flags
atm_prid	Problem Report IDs	text	38	n/a		mandatory selectable hidden reset on rerun readonly audit trail
atm_prlink	Problem Report Links	textarea	38x5	formatted		mandatory selectable hidden reset on rerun readonly audit trail
user	Run By	user list	5	n/a	n/a	selectable graphable
assignedto	Assigned To	user list	5	n/a	n/a	selectable graphable

Figure 3 - Edit Execution Fields screen

1.17.3 SPECIAL EXECUTION FIELDS

Some Execution Fields are required and must be present. The delete icon for a row in the Execution Field editor is grayed out if the row's contents are required

There may be no more than one instance of some Fields. The copy icon for a row in the Execution Field editor is grayed out if only one instance of its Field is allowed

The following Fields are restricted (these Fields also have restrictions on which of their attributes may be edited).

assignedto User(s) to whom a Test Case is assigned in a Test Session. This Field is mandatory, there may be not be more than one such Field. This is a display-only Field – this information is modified with the Set test case assignment screen for a Session.

atm_prid This reserved field can be used to track problem report IDs that are associated with the test case. When ApTest Manager is associated with an issue tracking system, the interface to that system can be configured to automatically update this field with problem report IDs if they are assigned while a test case is being executed. If the **VALUE** attribute of this field is populated,

it represents a template for a URI that can be used to connect to the issue tracking system and retrieve the issue. The pattern '%PRID' is replaced with each problem report ID in the field, and the resulting string used as the target of a hypertext 'A' element (e.g. http://www.aptest.com/bugzilla/show_bug.cgi?id=%PRID).

atm_prlink This reserved field can be used to hold links to problem reports in an associated issue tracking system, similar to **atm_prid** above. Each element in this field will be turned the target of an HTML 'A' element when displayed. If both the **atm_prid** and **atm_prlink** fields are used, their values will be combined when displayed.

atm_step_notes This optional reserved field is used in step-by-step testing to capture notes about each step in a test case.

exectimeclock, exectimestaff

Clock (24 hour days) and staff (1-8 hour work days) time to execute a Test Case in a Test Session. In an execution template these Fields display a single select menu of the possible time values configured for the corresponding planned time Test Case Field. In a report they display the execution time supplied by the user the last time the Test Case was executed in a Test Session. These Fields are mandatory if the corresponding plannedtime Test Case Field is defined, there may be not be more than one of each such Field.

note When used in an execution template this Field provides a textarea field into which the user can enter notes relating to the execution of a Test Case in a Test Session.

When used in a report template, information on the last time the Test Case was executed in the Session is shown: the executing user, execution date/time, notes and execution time the user entered, and the values of any Fields that have the audittrail flag set.

This Field is mandatory, there may be not be more than one such Field.

notes When used in an execution template this Field provides a textarea field into which the user can enter notes relating to the execution of a Test Case in a Test Session, followed by a history of its previous executions. The executing user, execution date/time, notes and execution time the user entered, and the values of any Fields that have the audittrail flag set are shown for each prior execution.

When used in a report template the complete execution history is shown.

This Field is mandatory, there may be not be more than one such Field.

notetext This Field may not be included in execution templates.

When used in a report template the execution note entered by the user from the last time the Test Case was run in a Test Session is shown.

	This Field is mandatory, there may be not be more than one such Field.
results	When used in an execution template this Field displays the configured result codes as a series of radio buttons. When used in a report shows the result from the last time a Test Case was run in a Test Session This Field is mandatory, there may be not be more than one such Field.
resultsdropdown	When used in an execution template this Field displays the configured result codes as a single select menu. In a report shows the result from the last time a Test Case was run in a Test Session. This Field is mandatory, there may be not be more than one such Field.
user	The user who last ran a Test Case in a Test Session. This Field is mandatory, there may be not be more than one such Field. This is a display-only Field – this information may not be modified by the user.
when	When a Test Case was last run in a Test Session. This Field is mandatory, there may be not be more than one such Field. This is a display-only Field – this information may not be modified by the user.

1.17.4 EXECUTION FIELD TYPES

Values for the Type of an Execution Field and their significance are:

check boxes	In an execution template the values in the VALUES column are displayed as a set of check boxes. The layout of the check boxes depends on the value of SIZE. SIZE can be either AUTO, WIDTHxAUTO (e.g. 4xAUTO), or AUTOxHEIGHT (e.g. AUTOx4). If SIZE is AUTO, the boxes are output in a single paragraph. The each input and its label is separated by a non-breaking space, so the inputs and labels will not be separated if the paragraph wraps. If SIZE is WIDTHxAUTO, the inputs will be displayed in a grid WIDTH cells wide. If the number of inputs is not an integer multiple of WIDTH, there will be empty spaces in the last row, at the right. If SIZE is AUTOxHEIGHT, the inputs will be displayed in a grid HEIGHT rows high. If the number of inputs is not an integer multiple of HEIGHT, there will be empty spaces in the last column, at the bottom. Default values (which are assigned to Test Cases in new Sessions as they are created) may be specified by placing an asterisk before a value in the VALUES column.
--------------------	--

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will remain available for searching and reporting, and will appear in reports or when reviewing session results, but will not be available when executing tests.

date

A Field which contains a date/time value. In an execution template a value may be selected from a calendar. A default value may be selected for this Field. The default value may be cancelled by clicking the adjacent 'x'.

When importing results, values for a Field of this type should be represented as seconds since the epoch (the time 00:00:00 UTC on January 1, 1970)

In the report the value of a date Field is a string representing the value of the Field.

multi-select menu

In an execution template the values in the VALUES column are displayed as a multi-selection menu. If there is enough space on the page the menu will be placed adjacent to its label. Otherwise it will move onto the line below. SIZE specifies the number of values displayed at once.

Default values (which are assigned to Test Cases in new Sessions as they are created) may be specified by placing an asterisk before a value in the VALUES column.

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will remain available for searching and reporting, and will appear in reports or when reviewing session results, but will not be available when executing tests.

Menu fields can depend on the values of checkboxes or radio buttons Fields. The semantics are the same as depending on multi- and single-select menus, respectively. In reports the value of the Field is the values selected from the menu.

radio buttons

In an execution template the values in the VALUES column are displayed as a set of radio buttons. The layout of the radio buttons depends on the value of SIZE.

SIZE can be either AUTO, WIDTHxAUTO (e.g. 4xAUTO), or AUTOxHEIGHT (e.g. AUTOx4).

If SIZE is AUTO, the buttons are output in a single paragraph. The each button and its label is separated by a non-breaking space, so the buttons and labels will not be separated if the paragraph wraps.

If SIZE is WIDTHxAUTO, the buttons will be displayed in a grid WIDTH cells wide. If the number of buttons is not an integer multiple of WIDTH, there will be empty spaces in the last row, at the right.

If SIZE is AUTOxHEIGHT, the buttons will be displayed in a grid HEIGHT rows high. If the number of buttons is not an integer multiple of HEIGHT, there will be empty spaces in the last column, at the bottom.

A default value (which is assigned to Test Cases in new Sessions as they are created) may be specified by placing an asterisk before one of the values in the VALUES column. Setting a second default value is silently ignored.

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will remain available for searching and reporting, and will appear in reports or when reviewing session results, but will not be available when executing tests.

single-select menu

In an execution template the values in the VALUES column are displayed as a single-selection menu. If there is enough space on the page the menu will be placed adjacent to its label. Otherwise it will move onto the line below. SIZE specifies the number of values displayed at once.

A default value (which is assigned to Test Cases in new Sessions as they are created) may be specified by placing an asterisk before one of the values in the VALUES column. If no default value is specified, the first value becomes the default. Menu fields can depend on the values of checkboxes or radio buttons Fields.

Obsolete values may be indicated by placing a carat (^) before a value in the VALUES column. Obsolete values will remain available for searching and reporting, and will appear in reports or when reviewing session results, but will not be available when executing tests.

The semantics are the same as depending on multi- and single-select menus, respectively. In reports the value of the Field is the value selected from the menu.

To utilize the mandatory flag with a single-select menu Field and require the user to make a selection, define its first value to be empty. For example:

	priority	Priority	menu (single-select)	1	plain	High, Medium, Low	readonly mandatory selectable graphable Depends On Name part
--	----------	----------	----------------------	---	-------	-------------------------	---

Otherwise the user is able to leave the Field with the default value selected, but without necessarily making a selection explicitly.

text

In an execution template an input field of type="text" is displayed and text may be entered. For this type of Field, SIZE specifies the width of the Field in characters when editing, and

VALUES specifies the default value for the Field. In reports the value of a text Field is the text entered.

textarea

In an execution template an input field of type="textarea" is displayed and text may be entered. SIZE specifies the width and height of the Field when editing, using the form "WxH". The width may be a numeric value (in characters) or AUTO (e.g. AUTOx2) in which case the size of the Field automatically adjusts to 95% of the size of the containing element (e.g. a table a cell). VALUES specifies the default value for the Field. In reports the value of a textarea Field is the text entered.

1.17.5 EXECUTION FIELD STYLES

The style column value is significant for date and textarea Fields.

1.17.5.1 DATE FIELD STYLES

For date Fields valid styles are:

date only The value of the Field is displayed as a month, day, and year.

date and time The value of the Field is displayed as the month, day, year, hour, minute, and time zone.

1.17.5.2 TEXTAREA FIELD STYLES

ApTest Manager offers four styles of Execution textarea Fields.

The contents of textarea Fields are stored as HTML, but many textarea Field styles allow this information to be entered by the user without using HTML directly. These styles format information in different ways when the Field is edited, viewed, or shown in a report.

A textarea Field's style is shown in parentheses after the name the Field in a Test Case execution template.

Supported textarea Field styles and the way they handle information are shown below. Some additional deprecated styles are supported for backward compatibility with earlier versions, but are not shown here.

asis HTML special characters are transformed into their general entity equivalents, newlines are placed at the end of every line. This retains exactly the line formatting the user enters and this style should be used when that is important. A downside of this style is that lines are not wrapped to take advantage of a larger window. Text is displayed in a monospace font. This style is used primarily for entering fixed format data such as source code into a Field.

formatted This style allows text to be entered much like typing on a typewriter.

Newlines can be used to end lines and paragraphs.

Spaces or tabs can be used at the start of a line to indent text.

Special characters are automatically transformed into HTML so they display correctly in the browser (e.g. '<' is transformed into "<").

Plus, lists can be created automatically. Lines within the Field that begin with "#. " (the pound sign followed by a period and a space) are automatically numbered when the Test Case is viewed or executed. This is a convenient mechanism for making a Test Case easy to understand and maintain. Lines beginning with "nnn. " (a number followed by a period and a space) are included in the numbered list, with the specified number.

html

Contents are expected to include HTML.

Special characters need to be specified in HTML and all formatting is specified with HTML directives. For example, newlines do not have any effect, the HTML
 or <P> directives need to be used instead or lines run together when the Field is displayed. Knowledge of HTML is required in order to use a Field of this style.

wysiwyg

A What You See Is What You Get HTML editor provides a Word Processor style interface for formatting information. This is the most commonly used textarea Field style. See the *ApTest Manager User Guide* for details. Contents are stored in HTML, transparently to the user. Note the height value in the SIZE attribute for this style of Field needs to allow for the display of several rows of icons.

wysiwyg2

Similar to **wysiwyg** above, but the field only becomes editable when the 'Open Editor' button is clicked. Suitable for use when there are many wysiwyg fields on the form and you want to conserve screen space or reduce page load time.

1.17.6 EXECUTION FIELD FLAGS

In absence of a flag the indicated functionality does not apply to the Field. Valid flags are:

audit trail

The value of the Field is included in the execution notes, as part of a table of Fields with this flag set and their values. These Field/values are thus recorded in the notes Field's audit trail for each execution and can be included in reports, the Session Summary, etc. The current value of a Field for a Test Case depends on the value of the reset on rerun flag, regardless of any audit trail.

dependson

The value of the flag should be another menu Field (single- or multi-select). Note that a Field that has other Fields dependent on it cannot have its name changed and must be selectable.

MANAGING TEST SUITES

When editing a Test Case, the values displayed for this Field depend on the value(s) of the other menu. Entries for values are separated by newlines and optional commas and have dependencies specified in braces after them. Please note these are braces and NOT parentheses. The contents of the braces are a comma separated list of dependencies that this value is displayed for.

If two or more entries are included in the list for the same value, e.g.

1{A},
1 {B}

the last value is used and previous entries are discarded.

The dependencies are ORed. There is no AND and no NOT.

Multiple menu Fields may depend on a single menu Field. Dependent fields can be cascaded (i.e. a Field may both be dependent on another Field and have another Field dependent on it).

Edit Execution Fields



User: Bob Manager [logout](#)
[Back to Manage Suite](#)

ApTest_Manager_Sample [Suite Manager]

Through this screen you can review and make changes to this Test Suite's Execution Fields.

You may include Execution Fields in the Test Case Execution templates to gather custom information at Test Case execution time.

Please review Chapter 1 of the ApTest Manager Admin Guide carefully before modifying this information. Press "Save Changes" when you are done.

[Save Changes](#) [Cancel](#)

	Name	Label	Type	Size	Style	Values	Flags
  	m1	Menu1	menu (multi-select)	5	n/a	*A, B, C, D	mandatory selectable graphable settable dependson reset on rerun readonly audit trail
  	m2	Menu2	menu (multi-select)	5	n/a	*0, 1{A}, 2{B}, 3{A,B}, 4{B}	mandatory selectable graphable settable dependson=m1 reset on rerun readonly audit trail

If m1 == A, m2 will contain 0, 1, 2, and 3.

If m1 == B, m2 will contain 0, 3, and 4.

If m1 == C, m2 will contain 0.

If m1 == [A, B], m2 will contain 0, 1, 2, 3, and 4.

Note that a multi-select menu may have no value selected (e.g. if no default value is specified), in which case any Field that depends on it will display only the values that have no dependencies specified. The behavior of circular dependencies (m1 depends on m2, m2 depends on m1) is not defined, but is almost certainly bad.

graphable	The values in the field may be used to graph “Results by Selector” in various reports.
hidden	The Field is not presented to the user if referenced in a template. It may only be used when referenced in a script.
mandatory	This Field must be given a value in a Test Session. Mandatory Fields are indicated with a red asterisk in execution templates.
readonly	The value of the Field may not be changed when executing the test. When a readonly Field is displayed during test execution a form element is not displayed, rather the Field’s value is shown. Fields marked readonly may be assigned if they are marked as settable
reset on rerun	If this flag is set the Field value is initialized to its default when a Test Case is rerun. If this flag is not set then the Field is initialized to its value from the last time the Test Case was executed in the Test Session.
selectable	Indicates a) reports can be sorted by a Field, and b) in reports that provide them a Results by Selector table/graph can be created for a Field (available only for specific Field types).
settable	This Field is available to have its values set for a Test Set or Test Session with the Assign test cases in Set/Session screens. Setting this flag and ‘reset on rerun’ is not recommended, as any value assigned is reset when a Test Case is run.

1.18 CONFIGURING TEMPLATES

Templates define the ways in which Test Case information is presented in ApTest Manager screens and reports. Separate templates are provided for editing Test Cases and Requirements, executing Test Cases, and for templated reports. **Requirement, Test Case, and Execution Fields must be included in templates in order to be displayed.**

The content of a template is HTML markup and Requirement, Test Case, and/or Execution Field names. Though information can be presented in any way desired, typically the template data is structured using HTML tables. Note however that anything can be put into a template that is legal in an HTML page, including text, HTML, javascript, VBscript, etc.

To incorporate a Requirement, Test Case, or Execution Field into a template, the name of the Field is enclosed within a pair of delimiters, preceded by the string "RQT" (e.g. <% RQT_domain %>) for Requirement Fields, the string "FIELD" (e.g. <% FIELD_domain %>) for Test Case Fields, and the string “RESULT” for (e.g. <% RESULT_version %>) for Execution Fields.

The Field is displayed based on how it is specified in the Requirement, Test Case, or Execution Field definitions, as text, a table, etc.

In Report templates the data for a Field is displayed. In Editing and Execution templates (for editing or executing a Requirement/Test Case) a form element is displayed for editable Fields allowing data to be entered or edited. For non-editable Fields the Field's value is displayed. In all cases the label for the Field is displayed along with the Field itself.

A template can be changed in any number of ways to alter the information displayed and its format.

- Change what Fields are shown in a report. Generally, different Fields may be included in different templates.
- Modify the format used to display information, to display it in some other way or just to alter the layout of a table, e.g.:
 - More or fewer table rows or columns can be employed.
 - The order of rows or columns in a table can be changed.
 - Some or all of the information can be formatted other than as a table, as an HTML list for example.
- Reflect Fields being added or removed from the Test Suite. If Fields are removed they must be removed from all templates where they were referenced. If Fields are added they must be added to any Templates in which they are to be displayed.

1.18.1 MISSING FIELD ERRORS IN TEMPLATES

If a Template contains a reference to a Field which is not defined an error message is displayed when a screen containing the template is shown, at the location of the Field reference in the template. An example is shown in Figure 4.

View Requirement



User: Bob Manager

ApTest_Manager_Sample [Suite Manager]

Close Window

Edit Requirement

Requirement: 1.0.1 User can point ATM to a bug tracking system															
Summary: User can point ATM to a bug tracking system			Creation Date: Nov 13, 2006 11:12AM CST												
Description: ApTest Manager can be configured to use a third party tool to log/manage bugs.															
Type: System	Unknown requirement field reference: <% RQT_xpriority %>	Difficulty: Medium	Source: Functional spec												
Object: User interface															
Owner Name: Joe Rice		Associated Files: None													
State: Ready Phase: n/a															
Number of Associated Tests: 1															
Associated Tests: ATM_Administration/bug_tracking_URI															
<table border="1"> <thead> <tr> <th colspan="4">Change History</th> </tr> <tr> <th>#</th> <th>Last Modification Time</th> <th>Last Modified By</th> <th>Change Details</th> </tr> </thead> <tbody> <tr> <td>1.</td> <td>Nov 13, 2006 11:12AM CST</td> <td>Sue Lake</td> <td>Created</td> </tr> </tbody> </table>				Change History				#	Last Modification Time	Last Modified By	Change Details	1.	Nov 13, 2006 11:12AM CST	Sue Lake	Created
Change History															
#	Last Modification Time	Last Modified By	Change Details												
1.	Nov 13, 2006 11:12AM CST	Sue Lake	Created												

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 4 - Display of Template with Missing Field

This is an error condition which should be remedied by removing the Field reference from the template or adding the Field to the Requirement, Test Case, or Execution Field definitions, as applicable.

1.18.2 CHECK TEMPLATES FOR ERRORS

Click this link from the Manage Test Suite screen to validate the correctness of the Test Suite's templates - flagging references to Fields that are not valid. Individual templates are also checked for errors when they are saved.

1.18.3 REPORT TEMPLATES

Templated reports can be configured to display different Fields, for purposes such as a Test Requirements report or a Test Specification report, and information can be formatted in different ways, as a spreadsheet or a series of tables for example.

A report template defines the information displayed for a single Requirement or Test Case. When information about several Requirements or Test Cases is displayed in a report, the template is applied to each Requirement/Test Case. So, for example, a report covering ten Requirements/Test Cases would contain ten tables, each formatted per that report's template.

Templates do not contain other report information, such as a Table of Contents.

Figure 5 shows the screen for editing report templates. New templates can be created or an existing template modified, including editing, renaming, or copying it. There can be an unlimited number of report templates, configured to produce reports on different Requirement/Test Case Fields.

Report names are limited to 50 characters, though this limit can be changed during installation.

In the Test Suite Profiles shipped with ApTest Manager a number of example report templates are included.

Two templates are provided for some reports, allowing two different versions to be generated. One version is defined with a Header/Footer element table, and thus displays like a spreadsheet (see Section 1.18.8.1). A second version presents more complete data for each Requirement/Test Case in a more complex format.

There are two types of report templates, based on the type of fields they contain: Test Case report templates and Requirements report templates. Each templated report is included in the list of reports available on the Select Report Screen, where they apply to Test Sessions. Some templated reports can also be generated from the Edit Requirements/Test Cases screens, where they apply to the repository managed with the screen.

Edit Report Templates



User: [Bob Manager logout](#)
 Back to [Manage Suite](#)

ApTest_Manager_Sample [\[Suite Manager\]](#)

You may [create a new report template](#), or work with an existing one:

	Name	Description	Report Type	Last Modified
	Traceability by Test Case	Requirements grouped by associated tests	Requirements	Sep 18 3:40:53PM CDT
	Traceability by Requirement	Tests grouped by associated requirements	Test Case	Sep 18 3:40:53PM CDT
	Test Specification SS	Test Case specification - Spreadsheet style	Test Case	Sep 18 3:40:53PM CDT
	Test Specification	Test Case specification	Test Case	Sep 18 3:40:53PM CDT
	Test Results SS	Results of test execution for a Session - Spreadsheet style	Test Case *	Sep 18 3:40:53PM CDT
	Test Results	Results of test execution for a Session	Test Case *	Sep 18 3:40:53PM CDT
	Requirements Test Results	Requirements with results for associated tests in Sessions	Requirements	Sep 18 3:40:53PM CDT
	Requirements Test Coverage	Requirements grouped by number of linked tests	Requirements	Sep 18 3:40:53PM CDT
	Requirements Spec SS	Requirements specification - Spreadsheet style	Requirements	Sep 18 3:40:53PM CDT
	Requirements Spec	Requirements specification	Requirements	Sep 18 3:40:53PM CDT

* Uses execution information

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 5 - Edit Report Templates screen

1.18.4 TEMPLATED TEST CASE REPORTS

Templated Test Case Reports may contain Test Session execution information as well as Test Case Fields. Templated Test Case reports that do not include execution information may also apply to the Test Case repository. These Templated reports appear on the Edit Tests screen as well as on the Select Report screen. Templated Test Case Reports generated from the Select Reports screen apply to a single Test Session.

1.18.5 TEMPLATED REQUIREMENTS REPORTS

Templated Requirements reports show information for all or part of the requirements in a Test Suite. As well for each Requirement the report can show information about the Test Cases that link to it (one Test Case per row in a table below the Requirement template). This provides a Requirements traceability matrix – which tests implement each Requirement.

The default Fields shown for a Test Case can be configured for the template and can contain execution information if desired. This provides a Requirements execution matrix – the results of the execution of Test Cases on a per Requirement basis.

Requirements templated reports are available from the Select Reports screen where they apply to one or more Test Sessions. Requirements templated reports are also available from the report list on the Edit Requirements screen, where

they can be used to create reports on the Requirements repository. If generated from this screen and the Test Case details configured includes execution information, “n/a” is displayed for this information, indicating it is not applicable.

1.18.6 EDITING AND EXECUTION TEMPLATES

Four Editing and Execution Templates are provided for each Test Suite, one for the display of Requirements information when Requirements are viewed and edited, one for the display of Test Case information when Test Cases are viewed and edited, and two used when Test Cases are executed.

Requirement Editing	View and Edit Requirement screens
Test Editing	View and Edit Test Case screens
Single Test Execution	Run Test Case screen
Multiple Test Execution	Run Multiple Tests screen

These screens contain information that is always included and is not configurable, along with template based information for a Requirement/Test Case. The Editing and Execution Templates specify what this Requirement/Test Case information consists of and how it is displayed.

Any Requirement/Test Case Fields other than those with the readonly flag set that are not included in the Requirement/Test Case Editing Template are effectively cleared on save of a Requirement/Test Case after editing.

1.18.7 TEMPLATE EDITOR

The Template Editor, shown in Figure 6, is used to create and modify templates. The Template Editor is a variant of the WYSIWYG editor for Test Case Fields discussed in Section 1.16.4. It contains additional table related commands and controls as well as the ability to insert delimited Field names into templates.

This editor can be used to create templates with a variety of data and formats. In the examples shipped with ApTest Manager much of the template formatting is done using tables.

NOTE: Use the HTML icon to edit the html for a template directly if difficulties are encountered with the graphical user interface.

The template editor also allows default values of some Customize Report settings to be set for Report templates, providing an additional level of report customization.

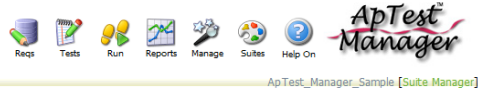
1.18.8 TABLE MANIPULATION

Additional table related controls are provided in the Template Editor:

MANAGING TEST SUITES

	Insert a new table/modify existing table.		Insert table column before.
	Table row properties.		Insert table column after.
	Table cell properties.		Delete table column.
	Insert table row before.		Split table cells.
	Insert table row after.		Merge table cells.
	Delete table row.		Toggle guidelines/invisible elements.

Edit Requirement Report Template



A report template allows you to specify what data is displayed for an ApTest Manager report and how it is formatted. Templates consist of HTML markup and ApTest Manager field names. Typically the template data is structured using HTML tables, but you can present the information in any way you wish. Use the WYSIWYG editor to edit the template or click the **HTML** icon to enter it directly as HTML.

Incorporating fields into templates is a very powerful mechanism and we suggest you carefully review Chapter 1 of the ApTest Manager Admin Guide before making template changes. Press "Save Changes" when you are done.

Template Name: Requirements Spec

Description: Requirements specification

<% RQT_id %>		<% RQT_cdate %>	
<% RQT_description %>			
<% RQT_type %>	<% RQT_priority %>	<% RQT_difficulty %>	<% RQT_source %>
<% RQT_atm_owner %>	<% RQT_files %>		
<% RQT_state %> <% RQT_phase %>		<% RQT_history %>	
<% RQT_atm_testcount %>			
<% RQT_atm_tests %>			
<% RQT_atm_best_result %>			
<% RQT_atm_average_result %>			
<% RQT_atm_worst_result %>			

Format: -- Font family: -- Font size: --

Requirements Fields: --

What type of table of contents? ☐ None ☒ By requirement ☐ By folder ☐ By all folders

Group requirements by values of:

Show requirements that have no value for group field? ☐ Yes ☒ No

Hide requirements that have no associated test cases? ☐ Yes ☒ No

Sort Requirement details in this order:

Sort by:
 and then sort by:
 and then sort by:
 and then sort by:

Hide associated test case details? ☐ Yes ☒ No

Each Requirement in a report may have details of associated Test Cases displayed. Select the default Test Case and Execution Fields to include in the report (The Test Case Name and Session Number will always be included):

Test Case Details	Field Selection	Include	Field Order
Assigned Test Developer		Number of Requirements Tested	
Requirements Tested		Product Versions	
Created By		Priority	
Creation Date		State	
Associated Files	→ Move Right	Features Tested	↑ Move Up
Change History	← Move Left	Test Cycles	↓ Move Down
Preconditions and Setup			
Menu1			
Menu2			
Verification Procedure			

Show Results By Selector tables for:

Exclude Selectors	Field Selection	Include Selectors	Field Order
Last Modified By (requirement field)			
Number of Associated Tests (requirement field)			
Object (requirement field)			
Owner Name (requirement field)			
Priority (requirement field)			
Requirement (requirement field)	→ Include		↑ Move Up
Source (requirement field)	← Exclude		↓ Move Down
State (requirement field)			
Type (requirement field)			

Show only non-zero results in Selector tables? ☐ Yes ☒ No

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 6 - Edit Template screen

The main editor display contains menus for setting the Paragraph format, Font, etc. for text entered into the template.

To edit a table select it by clicking on it. Then use the Insert/Modify table icon to set properties for the table such as table spacing. A Class for the table may also be created, from a set of CSS styles defined by ApTest Manager. Similarly select table cells and table rows and use the Table Cell or Table Row Properties icons to set their properties, such as Cell type (data or header), alignment, background color, etc.

Areas of a table may also be resized by selecting and dragging them.

To insert new rows or columns, first select an existing row/column as the insertion point and then use the appropriate icon. Merging and splitting cells allows for the creation of templates combining different row and column spans in their table layouts.

To see the outline of a table that does not have a border defined use the Toggle guidelines/invisible elements icon.

To insert a Field reference into a template select the Field name from a pull down Field menu.

Note that a column that has cells in different areas of a table (e.g. header, body, and footer) cannot be deleted in one operation. Due to difference between areas cells in each area need to be deleted separately.

A template can be previewed by clicking **Preview**. An example of the template populated with sample data ("Lorem ipsum . . .") is shown in a separate window. When previewed table fields are shown with 3 rows of sample data.

1.18.8.1 TABLES WITH HEADER/FOOTER ELEMENTS

If an HTML table is specified in a template, ApTest Manager presents that table for each Test Case shown and outputs the name of each Field along with the Field's contents.

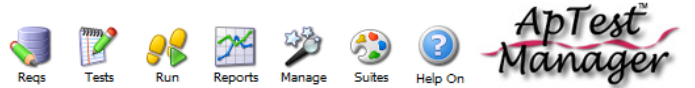
Templates can also have a layout more like a spreadsheet: a single table for all the Test Cases, with the titles of Fields shown only at the top of the table rather than in the table cells.

To specify this sort of presentation in a template, specify a table using Header and Footer elements. This causes ApTest Manager to generate one table for all Test Cases and to not include Field titles in the table (thus they must be specified in the Header/Footer elements in the template). For example, the following header/Footer template can be used for a Templated report

Use the Table Row Properties icon to set the Row in table part attributes for rows in the template. Set this to Table Head for header row(s) and Table Foot for footer row(s), and Table Body for other rows.

Note that the titles for the table columns have to be specified in the template for this style of presentation. They are not automatically drawn from the Field definitions as they would be otherwise.

Edit Test Case Report Template

User: [Bob Manager](#) [logout](#)

ApTest_Manager_Sample [Suite Manager]

Back to [Manage Suite](#) → [Edit Report Templates](#)

A report template allows you to specify what data is displayed for an ApTest Manager report and how it is formatted. Templates consist of HTML markup and ApTest Manager field names. Typically the template data is structured using HTML tables, but you can present the information in any way you wish. Use the WYSIWYG editor to edit the template or click the **HTML** icon to enter it directly as HTML.

Incorporating fields into templates is a very powerful mechanism and we suggest you carefully review Chapter 1 of the ApTest Manager Admin Guide before making template changes. Press "Save Changes" when you are done.

Template Name: Test Results SS

Description:

Test Case	Results	PRs	Features Tested	Test Cycles	Product Versions
<% FIELD_id %>	<% RESULT_note %>	<% RESULT_atm_prid %>	<% FIELD_testarea %>	<% FIELD_testcycle %>	<% FIELD_class %>
Test Case	Results	PRs	Features Tested	Test Cycles	Product Versions

-- Format -- -- Font family -- -- Font size --

-- Test Case Fields -- -- Execution Fields -- -- Session Variables --

What type of table of contents? ☐ None ☒ By test case ☐ By folder ☐ By all foldersGroup test cases by values of: Show test cases that have no value for group field? ☐ Yes ☒ No

Sort test case details in this order:

Sort by:
and then sort by:
and then sort by:
and then sort by:

Show Results By Selector tables for:

Exclude Selectors

Features Tested (test field)
ID (test field)
Menu1 (test field)
Menu2 (test field)
Number of Requirements Tested (test field)
Priority (test field)
Product Versions (test field)
State (test field)
Test Cycles (test field)
Assigned To (execution field)

Field Selection

Include Selectors

Show only non-zero results in Selector tables? ☐ Yes ☒ No

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 7 - Example Header/Footer table

1.18.9 FIELD NAMES IN TEMPLATES

Special commands are provided in the Template Editor for inserting Field names:

Requirement/Test Case Fields – this is a pulldown menu used to insert delimited user-defined Field names into templates. The pull down contains Requirements or Test Case Field names, depending on whether a Requirement or Test Case template is being edited. Selecting a Field from the menu causes the delimited Field name to be inserted at the cursor's location in the template.

Execution Fields (not used for Requirements templates or the Test Case Editing template) – this is a pulldown menu used to insert delimited Execution Field names into templates. Selecting a Field from the menu causes the delimited Field name to be inserted at the cursor's location in the template, so information can be entered or displayed for the execution of a Test Case within a Test Session.

Session Variables (not used for Requirements templates) – this is a pulldown menu used to insert delimited user-defined Session Variable names into templates. Selecting a Variable from the menu causes the delimited Variable name to be inserted at the cursor's location in the template.

To delete a Field name select the text within the template and use Backspace or Delete to remove it.

Fields contained within a table or modification history table Field may not be included in a template outside the table. They may only be included in the template by referencing the table Field. Thus these fields are not included in the list of Fields in the Template Editor. Referencing them directly results an error. This error is flagged when a template is saved and must be corrected.

1.19 TIME TRACKING FIELDS

ApTest Manager tracks information about planned and actual testing schedules using special Fields in a Test Suite's Test Case Templates.

There are two sets of time Fields, one set for staff time (using a work day that can be configured to be from 1-8 hours) and a second set for clock or calendar time (using a 24 hour day). Use neither set to not track testing time, either set to track just one kind of time, or both sets to track both staff and calendar time.

To track staff time perform the following steps. This causes ApTest Manager to gather planned and actual staff time for each Test Case and to report this information in the Progress and User Reports. The Test Suite Profiles shipped with ApTest Manager already contain staff time tracking.

- Place the following Field in the Test Suite's Test Case Field definitions:

NAME: plannedtimestaff

SELECTABLE: either N or Y (to be able to define Test Sets using it)

TYPE: Single Select Menu

MANAGING TEST SUITES

STYLE: minutes1-minutes8 (to select the length of the work day, from 1-8 hours)

SIZE: 1

LABEL: Planned Staff Time

VALUE: "1, 2, 5, 10, 30, 60, 120" (or other values for time to run a test. Values may also be specified as hours (e.g. 3h) or days (e.g. 5d). Fractional values are supported (e.g. 1.5h or 1.5d). Combinations of units (e.g. 1d3h) are not supported.)

- Add the Test Case Field `plannedtimestaff` to the Test Case Editing Template. This allows the staff time planned to execute a test to be entered as part of its definition.
- Add the Execution Field `exectimestaff` to the Single and Multiple Execution Templates. This allows the actual staff time to run a test to be entered as part of its execution.

In addition, the planned and/or actual staff time may be displayed in other reports by:

- Placing the Test Case Field `plannedtimestaff` in a report template to show the planned staff time
- Placing the Execution Field `exectimestaff` in a report template to show the actual staff time for a Test Session.

Time data is stored as minutes, based on the selected work day length. Thus if the work day length is changed the time data does not change, rather it is displayed as per the new day length. For example, time data set to 2d with a work day of 2 hours will become 1d if the work day length is changed to 4 hours.

To track clock time perform the following steps. This causes ApTest Manager to gather planned and actual clock time for each Test Case and to report this information in the Project Progress and Users Reports. Note that the Test Suite Profiles shipped with ApTest Manager are not configured to track clock time.

- Place the following Field in the Test Suite's Test Case Field definitions:

NAME: `plannedtimeclock`

SELECTABLE: either N or Y (to be able to define Test Sets using it)

TYPE: Single Select Menu

STYLE: minutes24

SIZE: 1

LABEL: Planned Clock Time

VALUE: "1, 2, 5, 10, 30, 60, 120" (or other values for time to run a test. Values may also be specified as hours (e.g. 3h) or days (e.g. 5d). Fractional values are supported (e.g. 1.5h or 1.5d). Combinations of units (e.g. 1d3h) are not supported.)

- Add the Test Case `plannedtimeclock` to the Test Case Editing Template. This allows the clock time planned to execute a test to be entered as part of its definition.

MANAGING TEST SUITES

- Add the Execution Field `exectimeclock` to the Test Case Execution Template. This allows the actual clock time to run a test to be entered as part of its execution.

In addition, the planned and/or actual clock time may be displayed in other reports by:

- Placing the Test Case Field `plannedtimeclock` in a report template to show the planned clock time.
- Placing the Execution Field `exectimeclock` in a report template to show the actual clock time for a Test Session.

1.20 CONFIGURING RESULT CODES

Figure 8 shows the Edit Result screen used to specify the result codes for a Test. Result codes are specified by testers when running tests and are displayed in many reports.

Result codes can be added, removed, and their attribute's values altered in order to define the desired results for the Test Suite. For example, adding a result code indicating a test needs to be rerun to verify a fix. If a result that was already used in this Test Suite is removed or renamed it is not possible to select reports on tests that match the result.

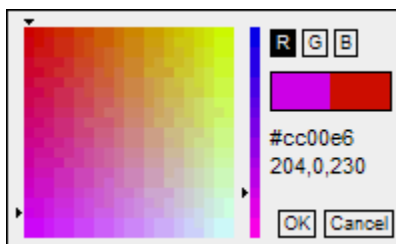
Result codes can be rearranged, so they are presented in the desired order. In calculating best/worst/average results the order of result codes defined in this screen is used. Results should thus be ordered from 'best' to 'worst'.

Please note the result code `UNTESTED` is used for those Test Cases which have not been run yet, and so that result must be defined. Also note that the result code `INCOMPLETE` is used to indicate that a test case has been started, but is not yet finished. This result must also be defined.

The `NAME` column defines a short name used to display the result in reports.

The `LABEL` column defines a long name used to define the result on the Run Test screens.

The `COLOR` column determines the color used to display the result code. Colors which are portable across browsers and display resolutions are recommended to avoid display issues. A color can be entered directly as a hexadecimal number preceded by '#'. Or a color picker may be used by clicking the palette icon .



Click on the vertical color grid to move the slider and select a color range. Click on the square color grid to select a particular color. The left side of the horizontal color bar shows the last color value that was clicked; the right side shows the current color under the cursor. Click OK to select the last clicked value or click Cancel to discard it.

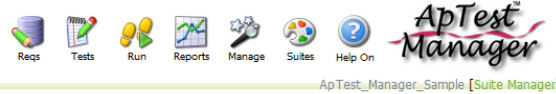
MANAGING TEST SUITES

It is common to select the same color for results of a similar nature, e.g. red for all failure results, green for all non-failures, and gray for results that are indeterminate.

The FLAGS column allows you to specify attributes for the result code. Available attributes include:

- Exclude – the result will be excluded from the count of completed tests in a test session.
- Hidden – the result code is defined, but is never displayed in the list of results that a user can select (e.g., the result might be used as part of test automation or issue tracking integration).
- Incomplete – the result means that the test has been started, but does not yet have a final result. If you are using step-by-step testing in a test suite, at least one result should have this flag set so that the system has something to which to default the result of an 'in progress' test case.
- Notify – a notification is sent to subscribers whenever the result is used for a test case in a test session.
- Obsolete – the result code is not selectable when determining the results of newly run tests, but previous test results that used this value continue to be valid and meaningful.

Edit Result Codes



Administrator: [Bob Manager logout](#)
 Back to [Manage Suite](#)

ApTest_Manager_Sample [Suite Manager]

Through this screen you can review and make changes to the result codes codes for this Test Suite.

Please review Chapter 1 of the ApTest Manager Admin Guide carefully before modifying this information. *Note that if you remove or rename a result that was already used in a test session in this test suite, you will be unable to select tests that match that result.*

Press "Save Changes" when you are done.

[Save Changes](#) [Cancel](#)

	Name	Label	Color	Flags
   	UNTESTED	The feature was not tested	#808080 	 exclude  notify
   	PASS	The test passed	#00bf26 	 exclude  hidden  notify  obsolete
   	UNSUPPORTED	Feature not supported	#00bf26 	 exclude  hidden  notify  obsolete
   	BLOCKED	Test case could not be starte	#808080 	 exclude  hidden  notify  obsolete
   	UNRESOLVED	Result could not be determin	#ff0000 	 exclude  hidden  notify  obsolete
   	FAIL	The test failed	#ff0000 	 exclude  hidden  notify  obsolete
   	INCOMPLETE	The test has been started	#0000 	 exclude  notify

[Save Changes](#) [Cancel](#)

Copyright © 2000-2010 Applied Testing and Technology, Inc. All rights reserved.

Figure 8 - Edit Result Codes screen

1.21 CONFIGURING SESSION VARIABLES

The Session Variable editor specifies the characteristics of test environments configured for a Test Suite.

1.21.1 SESSION VARIABLE EDITOR

The Variables shown in Figure 9 are part of one of the Profiles in the catalog shipped with ApTest Manager.

Session Variables do not need to be included in Templates, though they can be for non-Requirements templates. If a Variable is deleted it needs to be removed from all templates it is used in.

1.21.2 SESSION VARIABLE ATTRIBUTES

Each Session Variable has six attributes:

NAME	The name of the Variable. Variable names may only contain alphanumeric characters as well as the period (.), hyphen (-) and underscore (_) characters, and are case-insensitive. The name of the Variable is presented in circumstances where using the Variable prompt is not practical (e.g. as a table column heading) so short, descriptive names should be used. The name is also used to reference Variable values from within Test Case Fields. Variable names beginning with atm_ are reserved for use by ApTest and are not available for use in names defined by users.
PROMPT	Defines the text label that is used when presenting this Session Variable (e.g. when setting or listing a Variable's value).
TYPE	Defines the ApTest Manager type of this Variable. The supported types are defined below.
SIZE	Describes the size of the input field used for this Variable when specifying a value for it.
VALUES	Specifies values for the Variable.
FLAGS	Indicates properties of the Variable. Multiple flags may be set per Variable.

Edit Session Variables



User: [Bob Manager logout](#)
Back to [Manage Suite](#)

ApTest_Manager_Sample [Suite Manager]

Through this screen you can review and make changes to your Test Suite's session variable definitions.

Please review Chapter 1 of the ATM Admin Guide carefully before modifying this information.

[Save Changes](#) [Cancel](#)

	Name	Prompt	Type	Size	Values	Flags
	atm_heading1	Test Environment	heading	n/a	n/a	n/a
	drop	What code drop?	menu (single-select)	5	A B C *D	readonly hidden mandatory selectable dependson
	os	What operating system?	menu (single-select)	5	Windows 95 Windows 98 Windows ME Windows 2000 Windows NT *Windows XP	readonly hidden mandatory selectable dependson
	browser	What browser?	menu (single-select)	5	IE5 IE6 *IE7 NS7 Firefox Opera	readonly hidden mandatory selectable dependson
	network	What network connection?	menu (single-select)	5	Dial-up 10baseT *100baseT 802.11g	readonly hidden mandatory selectable dependson

[Save Changes](#) [Cancel](#)

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 9 - Edit Session Variables screen

1.21.3 SESSION VARIABLE TYPES

Possible values for the Type of a Session Variable and their significance are:

check boxes

When specifying the value of the Variable (e.g. on the Create Test Session screen) the values in the VALUES column are displayed as a set of check boxes. The layout of the check boxes depends on the value of SIZE.

SIZE can be either AUTO, WIDTHxAUTO (e.g. 4xAUTO), or AUTOxHEIGHT (e.g. AUTOx4).

If SIZE is AUTO, the boxes are output in a single paragraph. The each input and its label is separated by a non-breaking space, so the inputs and labels will not be separated if the paragraph wraps.

If SIZE is WIDTHxAUTO, the inputs will be displayed in a grid WIDTH cells wide. If the number of inputs is not an integer multiple of WIDTH, there will be empty spaces in the last row, at the right.

If SIZE is AUTOxHEIGHT, the inputs will be displayed in a grid HEIGHT rows high. If the number of inputs is not an integer multiple of HEIGHT, there will be empty spaces in the last column, at the bottom.

Default values (which are assigned to new Test Sessions as they are created) may be specified by placing an asterisk before a value in the VALUES column. In other screens the value of the Variable is the values of the selected check boxes.

date

When specifying the value of the Variable (e.g. on the Create Test Session screen) a date value may be selected from a calendar. A default value may also be selected from a calendar. The default value may be cancelled by clicking the adjacent 'x'. Elsewhere the value of a date type Variable is a string representing the date selected for the Variable. The value of the Session is specified and displayed as a month, day, and year.

date and time

When specifying the value of the Variable (e.g. on the Create Test Session screen) a date and time value may be selected from a calendar. A default value may also be selected from a calendar. The default value may be cancelled by clicking the adjacent 'x'. Elsewhere the value of a date type Variable is a string representing the date selected for the Variable. The value of the Variable is specified and displayed as the month, day, year, hour, minute, and time zone.

Derived

Derived Variables are constant values based on the values of other Session Variables. The flags (see `derivedfrom` and `derivedfunc` below) specify one or more Session Variables and a function to which the value(s) of the Variables are passed. The return value from the function is the value of the derived Variable. The values of derived Variables are evaluated whenever the values of the Variables they are derived from are changed (by changing them at the Session, Set, or Suite level). SIZE indicates the size of the text Field to be used to enter a search string for the Variable on the Select Report and Run Tests screens if it has the selectable flag set

heading

The value of LABEL is displayed as title for the Variables following it. Other fields must be present but are ignored.

multi-select menu	When specifying the value of the Variable (e.g. on the Create Test Session screen) the values in the VALUES column are displayed as a multi-selection menu. If there is enough space on the page the menu will be placed adjacent to its label. Otherwise it will move onto the line below. SIZE specifies the number of values displayed at once. Default values (which are assigned to new Test Sessions as they are created) may be specified by placing an asterisk before values in the VALUES column. In other screens the value of the Variable is the values selected from the menu.
radio buttons	When specifying the value of the Variable (e.g. on the Create Test Session screen) in the VALUES column are displayed as a set of radio buttons. The layout of the buttons depends on the value of SIZE. SIZE can be either AUTO, WIDTHxAUTO (e.g. 4xAUTO), or AUTOxHEIGHT (e.g. AUTOx4). If SIZE is AUTO, the buttons are output in a single paragraph. The each input and its label is separated by a non-breaking space, so the inputs and labels will not be separated if the paragraph wraps. If SIZE is WIDTHxAUTO, the buttons will be displayed in a grid WIDTH cells wide. If the number of buttons is not an integer multiple of WIDTH, there will be empty spaces in the last row, at the right. If SIZE is AUTOxHEIGHT, the buttons will be displayed in a grid HEIGHT rows high. If the number of buttons is not an integer multiple of HEIGHT, there will be empty spaces in the last column, at the bottom. A default value (which is assigned to new Test Session as they are created) may be specified by placing an asterisk before one of the values in the VALUES column. Setting a second default value is silently ignored. . In other screens the value of the Variable is the values of the selected radio buttons.
single-select menu	When specifying the value of the Variable (e.g. on the Create Test Session screen) the values in the VALUES column are displayed as a single selection menu. If there is enough space on the page the menu will be placed adjacent to its label. Otherwise it will move onto the line below. SIZE specifies the number of values displayed at once. A default value (which is assigned to new Test Sessions as they are created) may be specified by placing an asterisk before one of the values in the VALUES column. If no default value is specified, the first value becomes the default. In other screens the value of the Variable is the value selected from the menu. To utilize the mandatory flag with a single-select menu Variable and require the user to make a selection, define its first value to be empty. For example:

MANAGING TEST SUITES

	priority	Priority	menu (single-select)	1	plain	High, Medium, Low	readonly mandatory selectable graphable Depends On Name part
--	----------	----------	----------------------	---	-------	-------------------------	---

Otherwise the user is able to leave the Field with the default value selected, but without necessarily making a selection explicitly.

text

When entering the value of the Variable (e.g. on the Create Test Session screen) an input field of type="text" is displayed and text may be entered. SIZE specifies the width of the Field when editing, and VALUES specifies the default value for the Variable. In other screens the value of the Variable is the text entered.

textarea

When entering the value of the Variable (e.g. on the Create Test Session screen) an input field of type="textarea" is displayed and text may be entered. SIZE specifies the width and height of the Field when editing, using the form "WxH". The width maybe be set to AUTO (e.g. AUTOx2) in which case the size of the Field automatically adjusts to 95% of the size of the containing element (e.g. a table cell). VALUES specifies the default value for the Field. In other screens the value of the Variable is the text entered. Session Variables to not have styles for their types, a Session Variable of type textarea operates much like a Test Case Field of type textarea and style html.

1.21.4 SESSION VARIABLE FLAGS

In absence of a flag, the indicated functionality does not apply to the Variable.

dependson

The value of the flag should be another menu Variable (single- or multi-select). Note that a Variable that has other Variables dependent on it cannot have its name changed and must be selectable.

When setting a Variable, the values displayed for this Variable depend on the value(s) of the other menu. Entries for values are separated by newlines and optional commas and have dependencies specified in braces after them. Please note these are braces and NOT parentheses. The contents of the braces are a comma separated list of dependencies that this value is displayed for. If two or more entries are included in the list for the same value, e.g.

1{A},
1{B}

the last value is used and previous entries are discarded.

MANAGING TEST SUITES

Multiple menu Variables may depend on a single menu Variable. Dependent Variables can be cascaded (i.e. a Variable may both be dependent on another Variable and have another Variable dependent on it).

A multi-select menu may have no value selected (e.g. if no default value is specified), in which case any Variable that depends on it will display only the values that have no dependencies specified.

The behavior of circular dependencies (m1 depends on m2, m2 depends on m1) is not defined, but is almost certainly bad.

The dependencies are OR-ed. There is no AND and no NOT.

Edit Session Variables



User: [Bob Manager logout](#)
Back to [Manage Suite](#)

ApTest_Manager_Sample [Suite Manager]

Through this screen you can review and make changes to your Test Suite's session variable definitions.

Please review Chapter 1 of the ATM Admin Guide carefully before modifying this information.

[Save Changes](#) [Cancel](#)

	Name	Prompt	Type	Size	Values	Flags
    	atm_heading1	Test Environment	heading	n/a	n/a	n/a
    	m1	Menu1	menu (multi-select)	5	*A, B, C, D	readonly hidden mandatory selectable dependson
    	m2	Menu2	menu (multi-select)	5	*0, 1{A}, 2{B}, 3{A,B}, 4{B}	readonly hidden mandatory selectable dependson=m1

If m1 == A, m2 will contain 0, 1, 2, and 3.

If m1 == B, m2 will contain 0, 3, and 4.

If m1 == C, m2 will contain 0.

If m1 == [A,B], m2 will contain 0, 1, 2, 3, and 4.

derivedfrom	Specifies the names of one or more Session Variables to be used to derive the value of a Variable of type <code>derived</code> . Variable names are separated by spaces. For example derivedfrom= a b c .
derivedfunc	Specifies the name of a function to be called to derive the value of a Variable of type <code>derived</code>. For example derivedfunc=my_function. The function must be written in Perl and placed under the directory in which ApTest Manager is installed, in the file <code>suites/derived/Derived.pm</code>. This file is the common location for all derivation functions and as distributed contains an example function. A derivation function is passed 2 parameters: the name of Variable being derived and a reference to an array of values for the Variables specified in the <code>derivedfrom</code> flag. A derivation function must return one value. The return value is treated as a string and used as the value of the derived Variable. Values for both derivedfrom and derivedfunc must be specified for correct operation of a derived Session Variable. Both flags must also be selected in the list of flags. The keystrokes required to do this vary from browser to browser; consult the browser's documentation for details. Often the Shift and Control keys can be used to select multiple values.
graphable	The variable can be graphed against results in various reports.
hidden	The Variable is not shown in the set of Session Variables, e.g. on the Select Report or Run Sessions screens. It may only be set from the Create Test Session screen and referenced from a Test Case.
mandatory	The Variable must be given a value in a Test Session (though not in a Test Set). Mandatory Variables are indicated with a red asterisk when specifying Variable values.
readonly	The value of the Variable is always the default value set for it in the Session Variable editor. The user may not otherwise set the value of the Variable.
selectable	The Variable is used to filter Test Sessions on the Run Tests and Select Report screens.

2 ADMINISTRATION

ApTest Manager administrative functions are accessed by logging into the system as the special user “admin” or by clicking on the Username of the currently logged in user when that user is a designated “Administrator”. The Username is displayed on the Suite bar, next to a link to log out of ApTest Manager.

Select an Operation

Administrator: [Jim Administrator](#) [logout](#)



Reqs



Tests



Run



Reports



Manage



Suites



Help On



ApTest_Manager_Sample [Suite Manager]

What would you like to do?



- Specify Test Requirements for this Test Suite



- Specify Test Cases for this Test Suite



- Execute Test Cases in this Test Suite



- Generate reports for this Test Suite



- Configure and administer this Test Suite



- Change the suite being worked with or create a new one



- Turn help displays off and on

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 10 - Click Username to Access Administration

ApTest Manager provides two types of users: users with the ability to administer the ApTest Manager installation, called Administrators, and those without these privileges, termed Normal users.

Administrators have access to extensive administrative features, including the ability to administer the accounts of other users. Normal users have access to more limited administrative capabilities that apply only for their own ApTest Manager account.

When ApTest Manager is first installed a special Administrator account named admin is automatically created. This account can be used after installation to create accounts for other actual users, some of which should also be given administrative privileges.

When an Administrator is logged in the Username on the Suite bar is followed by (Administrator).

2.1 IMPORTANT ADMINISTRATIVE FEATURES

2.1.1 MAKE SURE YOU DO NOT RUN OUT OF DISK SPACE

ApTest Manager stores information to disk on the server. Most ApTest Manager operations require sufficient free disk space to complete those operations. Should the partition storing ApTest Manager data on the server fill up, *DATA WILL BE LOST!* Symptoms can include empty Test Cases and Requirements as well as missing Set and Sessions. Therefore, always monitor the server to ensure that sufficient disk space is available.

If disk space on the ApTest Manager partition runs out, the only available recovery procedure is to restore all ApTest Manager data from a backup before ApTest Manager will become operational again.

2.1.2 EMAIL NOTIFICATIONS

Email notifications are emails that ApTest Manager can send to automatically when specific events occur. A number of email notifications are available for events relating to Test Case editing and execution and for Test Suite management.

Email notifications are configured separately for each Test Suite for each user. If Email notifications on? is set to Yes for a user, notifications are sent to that user for the selected events for a specific Test Suite. A global System Configuration item enabling an installation's support for email notifications must also be enabled by an Administrator.

2.1.3 TIMEZONES

Time information is maintained on the server but each user's account can be configured to present this information in a local timezone. A timezone is set by selecting a region, and then a city therein. An installation's timezone defaults to a global configuration setting or, in the absence of the global setting, to the timezone of the server.

2.1.4 TIME STYLES

Time may be displayed in 24-hour or 12-hour format on a per user basis. An installation's time style defaults to a global configuration setting.

2.2 LDAP ACCOUNTS

Some user account information can be provided by an LDAP Directory Service. Use of LDAP is an optional feature which must be configured and enabled in order for it to be made available to users (see Section 2.4.5).

An account created with LDAP is an "LDAP account" and will use LDAP. An account created without LDAP is a "non-LDAP account" and will not use LDAP. If LDAP is enabled the account creation screens allow LDAP as well as non-LDAP accounts to be created. Otherwise only non-LDAP accounts can be created.

ADMINISTRATION

To create an account the user enters a name into the Account name field on the Normal user's or Administrator's Create Account screen; this will become the name of the account created in ApTest Manager.

For LDAP accounts the account name must match a user ID in the LDAP Directory. LDAP field values can be retrieved for the account by the user clicking the Search LDAP button. These values cannot be entered manually. The account password is not entered for LDAP accounts, as passwords are not stored in Aptest Manager for these accounts; rather the password is stored in the LDAP directory. See the *ApTest Manager User Guide* for how the user name and password are validated via LDAP during login for LDAP accounts.

For non-LDAP accounts the values of the LDAP fields and the user's password are entered manually.

Non-LDAP accounts can be converted to LDAP accounts, providing their account name matches a user ID in the LDAP Directory Service via the Administrator's Change Existing Account screen.

If LDAP is enabled this screen contains three additional buttons:

Convert Accounts to LDAP converts some/all non-LDAP accounts whose account name matches a user ID in the LDAP Directory Service into LDAP accounts. Note that the account for the user admin will not be converted to an LDAP account (if it was an LDAP account and the LDAP server went down ApTest Manager could become inaccessible).

Synchronize All LDAP Accounts sets the values of all the ApTest Manager account fields for all LDAP accounts to their current values in the LDAP Directory Service.

Cleanup Removed LDAP Accounts removes from ApTest Manager's database of user accounts all LDAP accounts that are no longer defined in the LDAP Directory Service.

For LDAP accounts the values of LDAP fields may not be edited manually once the account is created. If LDAP is later disabled the values of these fields may be manually edited. If LDAP is later reenabled the LDAP values will again be used for these fields.

Create Account



Create account using LDAP?	☒ Yes ☐ No	
New account name:		Search LDAP
Full name:		<i>Provided by LDAP</i>
E-mail address:		<i>Provided by LDAP</i>
Phone number:		<i>Provided by LDAP</i>
Password:	<i>Authenticated via LDAP</i>	
Password (again):		
Rows per Page for reports:	1000	
User's time style:	☐ 24 Hour ☒ 12 Hour	
User's timezone:	America	Chicago
Email notification settings		
Email notifications on?	☒ Yes ☐ No	

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 11 – Normal User's Create Account screen with LDAP enabled

When LDAP is disabled all LDAP accounts become non LDAP accounts. A local password must be created for each of these accounts before it can be used. An attempt to login with such an account will be flagged as an error until an administrator configures a password for it. When LDAP is disabled and then reenabled former LDAP accounts will again become LDAP accounts and LDAP will again be used to provide their LDAP field values and password.

The LDAP fields for an LDAP account are updated with the current values in the LDAP directory each time the account's user logs in to ApTest Manager, overriding any local values that may have been set in the meantime (i.e. if LDAP was disabled in the interim).

2.3 NORMAL USER ADMINISTRATIVE FEATURES

2.3.1 ACCOUNT CREATION

If ApTest Manager is configured to allow Normal Users to create new accounts a link is provided on the login screen for new account creation. If ApTest Manager has been configured so only Administrators can create new accounts this link is not shown.

To create a new account entails entering information about the user and the account:

ADMINISTRATION

- Account Name
- User Name*
- Email Address*
- Phone Number*
- Password
- Rows to show per page for pagination
- Timezone
- Time style
- Whether email notifications are enabled

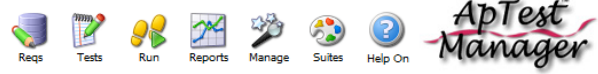
* For LDAP accounts the value of this information is provided by an LDAP directory

2.3.2 ACCOUNT MODIFICATION

Normal users can access to a screen for editing their ApTest Manager account:

- User Information (as above)
- Email Notifications

Edit Account Information

User: [Bob Manager logout](#)

ApTest_Manager_Sample [Suite Manager]

Full name:	Bob Manager														
E-mail address:	manager@aptest.com														
Phone number:	+1 763 786 8160														
Password (if you want to change it):															
Password (again):															
Rows per Page for reports:	3														
Your time style:	☐ 24 Hour ☒ 12 Hour														
Your timezone:	America Chicago														
Email notifications on?	Yes														
Suite email notifications															
Suite	Edit						Manage		Run						
	Requirement linked to owned test case changed ☐	Test case linked to owned requirement changed ☐	Requirement changed ☐	Owned requirement changed ☐	Test case changed ☐	Owned test case changed ☐	Suite config file changed ☐	Suite template file changed ☐	Session changed ☐	Session complete ☐	Session start ☐	Set changed ☐	Tests assigned ☐	Tests assigned to/from user ☐	
Agile_Example	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	
ApTest_Manager	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	
ApTest_Manager_Group	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	

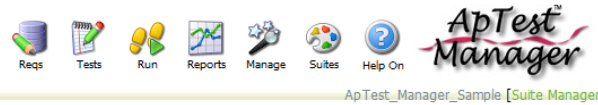
Figure 12 - Normal User's Edit Account screen

2.4 ADMINISTRATOR ADMINISTRATIVE FEATURES

Clicking on the Username of an Administrator brings up a screen with administrative features beyond those presented to a Normal User.

Manage ApTest Manager

Administrator: [Jim Administrator](#) [logout](#)



ApTest_Manager_Sample [Suite Manager]

Account Management

- [Show accounts](#)
- [Create an account](#)
- [Make changes to existing accounts](#)
- [Make changes to suite access](#)
- [Make changes to suite email notifications](#)

System Status

- [View login and license information](#)
- [View test suite information](#)

Configuration Management

- [Manage system configuration](#)
- [Manage test suite profile catalog](#)
- [Manage LDAP configuration](#)
- [Manage SQL configuration](#)
- [Initialize SQL datastores](#)

System Maintenance

- [Archive system data](#)
- [Check for updates to ApTest Manager](#)
- [Test email configuration](#)

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 13 - Administrator's Management screen

These include the features to manage the accounts of all defined users as well as to:

- Manually initiate a check for new versions of ApTest Manager, this check normally happens automatically
- Manage the contents of the catalog of Profiles for creating new Test Suites
- Manage a variety of system configuration options
- Manage the connection to an LDAP directory
- Manage the connection to an SQL database
- View information about an installation's license and the currently logged in users
- View information about all existing Test Suites
- Create a portable archive of all existing Test Suites
- Test the email configuration the user has defined for notification emails to ensure it emails are sent correctly.

2.4.1 ACCOUNT MANAGEMENT

Administrator account management functions are similar to the features provided to Normal users for creating and modifying their account information, with several important differences:

ADMINISTRATION

An Administrator can create, delete, view and modify accounts for all users. Note that an Administrator cannot delete their own account nor that of the user admin.

- An Administrator can give other users Administrator privilege.
- An Administrator can configure the access of users to specific Test Suites.
- Administrators can set special email notifications for themselves as well as set the email notifications of other users.
- Administrators can mark an account as 'disabled' – the user can no longer log into ApTest Manager, but their information will be retained in the system.
- Administrators can copy information from another account when creating a new account.

2.4.1.1 RESTRICTING TEST SUITE ACCESS

A user's access to a Test Suite is one of:

- No Access – the user cannot access this Test Suite.
- View Reports – the user can view reports but cannot view, edit, or run tests.
- Browse and View Reports – the user can view reports, Requirements, and Test Cases but cannot edit or run tests.
- Run Assigned - the user can view reports, Requirements, and Test Cases and execute tests assigned to them but cannot run other tests, edit tests or modify Test Sets or Sessions.
- Run Any Test – the user can view reports, Requirements, and Test Cases and execute tests but cannot edit tests or modify Test Sets or Sessions.
- Manage Execution – the user can view reports, Requirements, and Test Cases, execute tests, and modify Test Sets and Sessions but cannot edit tests.
- Manage Execution and Edit – the user can view reports, Requirements, and Test Cases, execute tests, modify Test Sets and Sessions, and edit Test Cases.
- Suite Manager – the user can view reports, Requirements, and Test Cases, execute tests, modify Test Sets and Sessions, edit Test Cases, and modify the configuration of the Test Suite.

When a new Test Suite is created the user creating the Suite is given Suite Manager access to it, other users are given the default access level (see Section 2.4.4.4).

ADMINISTRATION

When creating a new account or modifying an existing account an Administrator can specify the access level to be applied to each existing Test Suite.

Make changes to existing accounts

Administrator: [Jim Administrator](#) [logout](#)
Back to [Manage ATM](#)



ApTest_Manager_Sample [Suite Manager]

[Convert Accounts to LDAP](#) [Synchronize All LDAP Accounts](#) [Cleanup Removed LDAP Accounts](#)

Select an account

administrator
developer
lake
lu
manager

 or user name

-- user list --
Bob Manager
Jim Administrator
Joe Developer
Joe Rice

[Edit](#)

[Update Account](#) [Delete Account](#) [Cancel](#)

Account:	manager							
Account Disabled?	☐ Yes ☒ No							
Full name:	Bob Manager							
Last login:	Sep 21 2:54PM CDT							
E-mail address:	manager@aptest.com							
Phone number:	+1 763 786 8160							
Password (if you want to change it):	password							
Password (again):	password							
Rows per page for reports:	3							
User's time style:	☐ 24 Hour ☒ 12 Hour							
User's timezone:	America Chicago							
Feature access controls:	Feature	Yes	No					
	Administrator	☐	☒					
Suite access controls								
Test Suite	Access Level							
	None select all	View Reports select all	Browse and View Reports select all	Run Assigned select all	Run Any Test select all	Manage Execution select all	Manage Execution and Edit select all	Suite Manager select all
Agile_Example	☐	☐	☐	☐	☐	☐	☐	☒
ApTest_Manager	☐	☐	☐	☐	☐	☐	☐	☒
ApTest_Manager_Group	☐	☐	☐	☐	☐	☐	☐	☒

Figure 14 - Administrator's Change Existing Account screen

An Administrator can also view and change the access permissions for a Test Suite on a user-by-user basis.

Make changes to suite access



Administrator: [Jim Administrator logout](#)
 Back to [Manage ATM](#)

Select a Test Suite and Review/Change its access permissions:

Agile_Example
 ApTest_Manager
 ApTest_Manager_Group
 ApTest_Manager_New
 ApTest_Manager_Sample

Edit

Access Permissions for suite ApTest_Manager_Sample									
Account	User Fullname	Access?							
		None select all	View Reports select all	Browse and View Reports select all	Run Assigned select all	Run Any Test select all	Manage Execution select all	Manage Execution and Edit select all	Suite Manager select all
administrator	Jim Administrator	☐	☐	☐	☐	☐	☐	☐	☒
developer	Joe Developer	☐	☐	☐	☐	☐	☐	☒	☐
lake	Sue Lake	☐	☐	☐	☐	☐	☐	☒	☐
liu	Norm Liu	☐	☐	☐	☐	☐	☐	☒	☐
manager	Bob Manager	☐	☐	☐	☐	☐	☐	☐	☒
reporter	Sam Reporter	☐	☒	☐	☐	☐	☐	☐	☐
rice	Joe Rice	☐	☐	☐	☐	☐	☐	☒	☐
tester	Sue Tester	☐	☐	☐	☐	☐	☒	☐	☐

Update

Cancel

Figure 15 - Administrator's Change Suite Access Permissions screen

2.4.1.2 EMAIL NOTIFICATIONS

Using the Change Existing Account or Make Changes to Suite Email Notifications screens an Administrator can set the email notifications for any user or for all users for a specific Test Suite. As well, an Administrator may have several additional email notifications beyond those available for a Normal user.

To enable email notifications several global configuration entries also need to be set from the Manage System Configuration screen, to specify the SMTP server to be used, etc.

A user must have email notifications enabled (as set on the screens for creating and modifying user accounts) in order to receive email notifications

2.4.2 CHECK FOR UPDATES TO APTTEST MANAGER

From time to time ApTest Manager checks to see if a new revision is available by sending a request to a server at Applied Testing and Technology. This request includes license information along with a "signature" to uniquely identify

ADMINISTRATION

this copy of ApTest Manager. It does not include any other information about users, tests, nor anything that is related to how an installation of ApTest Manager is used.

If new updates are available a notification email is sent to Administrators for which this notification is enabled and a message is displayed on the login screen.

Click this link to initiate a check for updates manually.

2.4.3 PROFILE CATALOG MANAGEMENT

The Manage Test Suite Profile Catalog screen allows administration of the catalog of Profiles used to provide an initial configuration when creating a new Test Suite.

A number of Profiles are provided with ApTest Manager, offering different Field definitions and Suite configurations. These Profiles may not be modified.

From the Manage Test Suite Profile Catalog screen additional Profiles can be added to the catalog in two ways:

- Import a Profile from an existing Test Suite. Click the import link and specify a Test Suite, Profile name, and Profile description.
- Copy an existing Profile. Click on the Copy link for one of the existing Profiles in the catalog and specify a name for the new Profile.

Profiles that have been added to the catalog may be edited, deleted, or renamed by clicking on the appropriate link for the Profile. Clicking the Profile name displays information about its configuration.

Editing a Profile follows the same procedure as defined in Chapter 1 for Managing Test Suite Configuration. Changes made to a Profile in the catalog apply to new Test Suites created in the future but not to existing Test Suites previously created using that Profile.

2.4.4 SYSTEM CONFIGURATION

The Manage System Configuration screen provides control over a number of installation-wide configuration items.

2.4.4.1 SHOULD APTTEST MANAGER BE CLOSED TO NON-ADMINISTRATIVE USERS?

Selecting Yes displays a message on the login page saying ApTest Manager is temporarily only available to Administrators and to try back later. Only users with administrative privilege are able to login.

ApTest Manager is closed to non-administrative when the product is first installed.

To open ApTest Manager to Normal users:

ADMINISTRATION

- Login as a user who is an Administrator and click on the username on the Suite bar.
- Click on Manage system configuration.
- Select "No" for the question "Should ApTest Manager be closed to non-administrative users?" and click Update Configuration.

2.4.4.2 WHAT MESSAGE SHOULD BE DISPLAYED ON THE LOGIN PAGE?

A message in HTML may be specified that is displayed on the login screen (where a user is asked for their name and password). This is useful as an installation wide message of the day.

2.4.4.3 BUG TRACKING INTEGRATION

ApTest Manager provides support for problem tracking by allowing users to invoke a third party problem tracking tool. There are many packages, both commercial and Open Source, available in the market and we did not want to duplicate their functionality or limit our customers' choice of tool vendors.

When this feature is configured, a link to the problem tracking tool appears on each ApTest Manager screen used to run a Test Case. Users can click this link to invoke the problem tracking tool to submit bugs from within ApTest Manager. Depending on what the problem tracking system allows, ApTest Manager automatically populates the information for creating a new problem report from its database without requiring the user to reenter this information.

Problem tracking integration is configured by setting up a URI to initiate the problem tracking tool. This URI can point directly to the problem tracking tool if the system is web based. Or it can point to a script that invokes the command(s) for submitting a bug report; using an API, a CLI, SOAP, RPC, etc.. Consult with the administrator of the problem tracking tool to determine the best way to invoke it.

The URI that can be configured can include macros that pass Test Case execution specific information as desired:

- %USER - the name of the current ApTest Manager user
- %SESSION - the number of the current Test Session
- %SET - the name of the current Test Set
- %SUITE - the name of the current Test Suite
- %ID - the name of the current Test Case
- %ATM_ID - the internal identifier of the current Test Case
- %RESULT - the result of running the Test Case

ADMINISTRATION

- %RESULT_anything – the value of any execution field (e.g., %RESULT_atm_prid would return the value of the reserved field for problem report IDs).
- %NOTES - the user's notes from running the Test Case
- %URI – the URI for the ApTest Manager screen for viewing the Test Case
- %URIRUN – the URI for the ApTest Manager Run Test screen for this Test Case in this Session
- %URISUM – the URI for the ApTest Manager Session Summary for this Session

For example, the following URI would pass the current user, Test Case, and test result to an imaginary URI:

<http://example.com/probtrack?user=%USER&testcase=%ID&result=%RESULT>

ApTest Manger includes example files, in the directory `examples`, containing scripts illustrating how a problem tracking tool integration can be implemented. Examples are provided for several Open Source and commercial problem tracking systems such as Bugzilla, JIRA, Mantis, and Test Track Pro.

These files are Perl scripts that are invoked from ApTest Manager with a URI that passes all the information in the list above. The scripts use ApTest Manager's APIs to acquire even more information and places it all in a form for the user to review and modify. Once the user is satisfied the script invokes the problem tracking tool to create a new bug using this information. Refer to the documentation in these example integration scripts for how to configure them.

As URIs for ApTest Manager are included in the problem reports it files, users of the problem tracking system can invoke ApTest Manager to view or rerun the associated Test Case.

2.4.4.4 WHAT IS THE DEFAULT TEST SUITE ACCESS MODE?

There are eight levels of access a user may have to a Test Suite:

- No Access – the user cannot access this Test Suite.
- View Reports – the user can view reports but cannot view, edit, or run tests.
- Browse and View Reports – the user can view reports, Requirements, and Test Cases but cannot edit or run tests.
- Run Assigned - the user can view reports, Requirements, and Test Cases and execute tests assigned to them but cannot run other tests, edit tests or modify Test Sets or Sessions.
- Run Any Test – the user can view reports, Requirements, and Test Cases and execute tests but cannot edit tests or modify Test Sets or Sessions.

ADMINISTRATION

- Manage Execution – the user can view reports, Requirements, and Test Cases, execute tests, and modify Test Sets and Sessions but cannot edit tests.
- Manage Execution and Edit – the user can view reports, Requirements, and Test Cases, execute tests, modify Test Sets and Sessions, and edit Test Cases.
- Suite Manager – the user can view reports, Requirements, and Test Cases, execute tests, modify Test Sets and Sessions, edit Test Cases, and modify the configuration of the Test Suite.

The user who creates a new Test Suite is given Edit and Execute access to that Suite. This configuration variable determines the access other users have to a newly created Test Suite by default.

Administrators can change a user's access to a Test Suite from this default any time after the Suite is created.

2.4.4.5 DO YOU WANT TO SEND RUNTIME ERROR NOTIFICATION TO APTEST?

If Yes is selected any runtime errors that may be encountered are automatically emailed to ApTest.

2.4.4.6 WHAT IS THE URI OF YOUR HTTP PROXY SERVER?

If using a proxy server to access the Internet, enter its URL here. Otherwise this should be left blank.

2.4.4.7 HOW MANY DAYS BEFORE INACTIVE LOGINS ARE REMOVED?

Users become inactive and their logins are made available for use by others after a period of inactivity. After the number of days specified here inactive users are removed from the list of logged in users, and thus need to re-login the next time they access ApTest Manager.

Note that if this period has not expired AND the option in 2.4.4.10 below is set to “No” then a user will not need to re-login in order to access ApTest Manager – the cookie cached by their browser will have legal credentials and they will be automatically given access.

2.4.4.8 WHAT IS THE DEFAULT TIME STYLE FOR YOUR USERS?

The default may be set to 12 or 24 hour time.

2.4.4.9 WHAT IS THE DEFAULT TIME ZONE FOR YOUR USERS?

To set a timezone select a region and then a city therein.

If it is not specified this value is set to the timezone of the server.

2.4.4.10 SHOULD USERS HAVE TO RELOGIN AFTER THEY CLOSE THEIR BROWSER?

If No is selected users need only reauthenticate the next time they access ApTest Manager after they click Logout. Selecting Yes requires the user to always reauthenticate when they next login after closing their browser. This offers increased protection in the event the user's computer may not be physically secure (i.e. an unauthorized person may bring the user's browser back up and resume their ApTest Manager Session if it was not logged out).

2.4.4.11 SHOULD THE TEST EXECUTION RESULT, NOTES, AND TIME FIELDS BE PRE-POPULATED WITH THE LAST INFORMATION ENTERED?

This is historical behavior that can be reenabled by selecting Yes this option. If No is selected these Fields are empty when run screens are initially displayed. Pre-population of other Execution Fields is controlled by the reset on rerun flag (see Section 1.17.6).

2.4.4.12 WHAT IS THAT MAXIMUM NUMBER OF SESSION VARIABLES THAT WILL BE DISPLAYED HORIZONTALLY?

At or below this number Session Variables are arranged horizontally on display. Above it they are arranged vertically. Note Session Variables are always displayed horizontally in the Test Case Details section of reports and always displayed vertically when showing default values for Test Sets.

2.4.4.13 WHAT IS THE MAXIMUM NUMBER OF SESSIONS THAT CAN BE DISPLAYED ON THE RUN TESTS AND SELECT REPORT PAGES?

The purpose of this parameter is to limit the displayed rows on these screens when there are many many test sessions that match a users filter settings. The default value is 500.

2.4.4.14 WHAT IS THE AUTHENTICATION TOKEN FOR USERS OF THE RPC INTERFACE?

ApTest Manager provides a simple RESTful RPC interface. Scripts that want to use this interface must supply an authentication token. You can define that token here. Providing no value for this token disables the RPC feature.

2.4.4.15 SHOULD THE YEAR BE INCLUDED WHEN DISPLAYING DATES THAT ARE IN THE CURRENT YEAR?

When displaying dates, ApTest Manager can suppress the year when it is the same as the current year. If you want the year displayed all the time, set this value to "yes".

2.4.4.16 HOW LONG IN SECONDS SHOULD TRANSACTIONS WAIT FOR A LOCK BEFORE ABORTING (>60!) ?

During some long running operations that need to be handled as a single, atomic transaction, ApTest Manager may lock a test suite's database. Other users who are trying to access the same test suite will be prevented from doing so until the lock is released. If this operation runs for an extended period of time, the other users will 'timeout' and be shown a warning that they could not access the suite. This parameter allows you to set how long the other users will wait before timing out.

2.4.4.17 SHOULD A USER BE PROMPTED IF A FOLDER, TEST CASE, OR REQUIREMENT NAME THEY ENTER WOULD BE AUTOMATICALLY CHANGED TO AVOID THE USE OF NON-PORTABLE CHARACTERS?

ApTest Manager stores files in a way that ensures they are named 'portably'. This means that the system will remove characters from names that are not considered portable. If you set this option to 'yes', then a user will be warned when that substitution is going to take place.

2.4.4.18 SHOULD TABLES BE PERMITTED IN WYSIWYG FIELDS?

Normally, wysiwyg and wysiwyg2 fields do not provide controls to allow embedding and editing of tables. If you want to enable embedded tables, set this option to 'yes'.

2.4.4.19 NORMAL USER CAPABILITIES

These configuration items control what ApTest Manager features are available to Normal users. Administrators always have access to these features. Features for which 'No' is selected have links/icons adjusted so that they cannot be accessed by Normal users, regards of their level of access to a Test Suite

Create their own accounts?

Edit their account information?

Delete their own accounts?

Change their email notification settings?

Suspend their email notifications?

Create test suites?

Delete test suites?

ADMINISTRATION

Change test suite configurations?

Import test cases from CSV files?

Delete test cases and folders?

Rename test cases and folders?

Lock/Unlock test cases and folders

Perform global replace operations on test cases?

Empty the trash folder in the test case tree?

Import test cases from CSV files?

Delete requirements and folders?

Rename requirements and folders?

Lock/Unlock requirements and folders?

Perform global replace operations on requirements?

Empty the trash folder in the requirements tree?

Delete test set folders/sessions?

Rename/Move test set folders/sessions?

Lock/Unlock test set folders/sessions?

Change other's private report settings?

User Run these tests/Run associated tests?

2.4.4.20 HOW MANY ROWS SHOULD BE DISPLAYED PER PAGE IN LARGE REPORTS?

This sets the system default for the pagination limit for reports and the Run Multiple screen. Modifying it also modifies the limit for users who have not changed their limit from the system default and for all subsequently created users.

2.4.4.21 EMAIL SETTINGS

These settings specify if email notifications should be sent, what email address they should be sent from, and what SMTP server should be used to send them. See Section 2.4.10 for how to test the email configuration after it is set. Incorrect email configuration can significantly degrade performance.

2.4.5 MANAGE LDAP CONFIGURATION

The Manage LDAP Server Configuration screen allows the user to select whether use of LDAP is enabled or disabled for this instance of ApTest Manager and if it is enabled to configure the connection to the LDAP directory service.

Before LDAP can be used by ApTest Manager an LDAP directory service that contains the 5 pieces of information ApTest Manager uses (password, user name, email address, and phone number) must be installed and operating.

See the LDAP directory service's documentation for how to install, configure, and administer it.



Note that LDAP service installation, configuration, and administration tasks are NOT performed by ApTest Manager. ApTest does not provide support (in any way, shape, or form) for performing these tasks or for determining the values that need to be entered here for the LDAP directory service.

The user needs to be familiar with LDAP and the LDAP Directory Service to be used in order to perform this configuration.

An LDAP connection has a number of tunable parameters that must be set to the values needed to communicate with the Directory Service.

If LDAP is enabled provide appropriate values for each parameter and click Make Changes.

LDAP Parameter	Meaning
Host Name	Internet address of the host where the LDAP Directory Service is running
Port Number	TCP/IP port to use in communicating with the Service (the default value of 389 is commonly used)
LDAP Protocol Version	Either 2 or 3 – value is Service dependent

ADMINISTRATION

LDAP communication scheme	Either "ldap" or "ldaps", a la "http" or "https"
Base for LDAP Searches	Organization-specific search filter that tells LDAP in what "domain" to search for the user id
LDAP DN	Required only if the LDAP Service requires authentication to search
LDAP DN Password	
Directory Service Field Names	Tell ApTest Manager what "fields" in the LDAP records correspond to data in ApTest Manager - user ID, full name, phone number, and email address. The default values are names commonly used.

Manage LDAP Server Configuration



Administrator: [Jim Administrator](#) [logout](#)
[Back to Manage ATM](#)

Update this installation's global LDAP server configuration settings, then press "Make Changes" to continue:

LDAP Support:	Enabled ▾
Host Name	ldap.aptest.com
Port Number	389
LDAP Protocol Version	3 ▾
LDAP Communication Scheme	ldap ▾
Base for LDAP searches	dc=aptest,dc=com
LDAP DN (if authentication needed to search)	cn=Manager,dc=aptest,dc=com
LDAP DN Password (if authentication needed to search)	••••••••
User ID Field Name	uid
Fullname Field Name	displayName
Phone Number Field Name	telephoneNumber
Email Field Name	mail

Copyright © 2000-2009 Applied Testing and Technology, Inc. All rights reserved.

Figure 16 – Manage LDAP Server Configuration screen

If LDAP is disabled this table will not be displayed. If LDAP is reenabled the last values saved will be shown in the table.

If LDAP is enabled it will be used in creating LDAP accounts and in validating user logins for them. See Section 2.2 and the *ApTest Manager User Guide* for details.

If LDAP is disabled it will not be used in creating/modifying user accounts or in validating user logins.

2.4.6 MANAGE SQL CONFIGURATION

See Chapter 4 for information on this and other SQL administration operations.

2.4.7 VIEW LOGIN AND LICENSE INFORMATION

This screen shows the number of users an installation's license allows along with the most recent activities of currently logged in users. When each user logged in and when they last accessed the system is shown along with their last major activity.

Once a user logs in they remain logged in until they log out (note that closing the browser does not necessarily cause a log out (see Section 2.4.4.10)). However:

- If a user has been idle (i.e. not communicating with the server) for approximately 10 minutes, their license becomes inactive, and available to be assigned to other users. This allows licenses for users that take a break or forget to logout to be effortlessly recycled. If the user becomes active again a license is reacquired if one is available. This process is transparent to the user and occurs automatically without requiring the user to login again. If no licenses are available however the user is told to wait for one to become available before proceeding.
- A user is automatically logged out, and will need to login again the next time they access ApTest Manager, if they are inactive for one or more days. This limit can be configured from the Manage System Configuration screen (see Section 2.4.4.7).

Users with an active logins are shown in green. Users with inactive logins, i.e. those that are available to be reassigned, are shown in red. Users that have logged out are not shown.

2.4.8 VIEW TEST SUITE INFORMATION

This screen shows a list of Test Suite names and descriptions. The list is similar to the Select Test Suite screen and Test Suites can be selected from it. Several additional capabilities are provided:

- All Test Suites are shown. Those not accessible to the user viewing this screen are shown without a link to select them.
- The creator of the Test Suite and the number of its Test Cases, Test Sets, and Test Sessions can be shown.
- Which of these additional columns is displayed is configurable. This can be used to make the table smaller and to improve performance. With large number of Test Suites performance can be an issue otherwise.

2.4.9 ARCHIVE SYSTEM DATA

This link runs the bin/exportAll.pl script, generating an archive file of all Test Suites. This archive can be used to migrate Test Suites when moving to a new server, or uploaded when submitting an ApTest Manager Problem Report. See also Section 3.4. Once created the archive is downloaded to the client of the user that invoked this link. You have the option of specifying an email address that will be notified when the archive is available. In this case you can navigate away from the screen while the archive is running and download the archive at your convenience.

2.4.10 TEST EMAIL CONFIGURATION

This link sends a test message to the current user, using the specified email configuration, and verifies it is sent successfully. The dialog between the source and destination systems is shown. If this test is not successful the email settings should be modified until a working configuration is achieved. Incorrect email configuration can significantly degrade performance.

3 ADVANCED TOPICS

This Chapter documents some additional features and ways to use ApTest Manager. Experience with using ApTest Manager is a prerequisite for this Chapter. A knowledge of web programming and languages may also be required, along with access to the server on which ApTest Manager is installed.

3.1 DATABASE

The database that underlies ApTest Manager is based on the Berkeley DB from [Oracle Corporation](#) "the most widely used open source developer database in the world with over 200 million deployments".

Database installation, configuration, and administration are performed transparently to the user by ApTest Manager for its internal database.

ApTest Manager provides an API layer on top of this database for accessing ApTest Manager specific information.

Objects and their methods underlying the ApTest Manager API are very efficient, relying upon in memory and on disk caches to speed up access to the most used information.

3.2 REVISION CONTROL INTEGRATION

ApTest Manager can be configured to operate with third party revision control tools that offer a command line interface. Examples are ClearCase, CVS, Perforce, RCS, and Subversion.

If revision control is configured, operations performed on tests by ApTest Manager are also communicated, transparently to the user, to the revision control system. The revision control system's standard UI can then be used to do things like generate change histories for tests or back out changes to an earlier level.

Details on configuring of ApTest Manager to utilize a revision control tool are provided in the ApTest Manager Installation Instructions.

3.3 REQUIREMENT AND TEST CASE FILES

Support for revision control tools is facilitated by a set of files and directories containing Requirement and Test Case information.

ADVANCED TOPICS

Files are updated by ApTest Manager when Requirements or Test Cases are written. They are not involved in reading Requirement and Test Case data, e.g. when generating reports.

The files for a Test Suite are stored in a directory named `suites` under the directory in which ApTest Manager is installed (referred to here as `ATM_ROOT`). Each Test Suite has its own directory tree rooted in `suites` named `<test suite name>.ts`. Within each Test Suite directory is a hierarchy of directories for each Test Case folder, each named `<folder name>.dir` and a directory named `REQUIREMENTS` that contains a hierarchy of directories for Requirements folders.

Within each Test Case directory there is a file for each Test Case, named `<test case ID>.bts`. Within each Requirement directory there is a file for each Requirement, named `<requirement ID>.req`. The Requirement and Test Case IDs are strings entered by the user or numeric values provided by ApTest Manager, depending on the style configured for the Requirement/Test Case ID field for a Test Suite. Each directory also contains a file `DESCRIPTION` containing the description of the folder and any uploaded files that have been placed in it. The `DESCRIPTION` file in the top level of the tree for a Test Suite and in the `REQUIREMENTS` directory contains the description of the Test Suite.

Each Requirement and Test Case file contains a human readable representation of the Requirement/Test Case. Information is included for each Field in the Requirement/Test Case. A line starting with `fieldName:` starts the information for a Field, and is followed by the Field's content. The content extends until the next line that starts with a `fieldName:` or until end of file. If the contents of a Field must have a line that starts with `fieldName:` the end of the previous line can be escaped with a backslash `"\"`

- The `fieldName` is a Requirement/Test Case Field name in upper case.
- For a Field that is a list (menu) Field, values are separated with a comma followed by a space.
- Table Fields do not have entries.
- The Fields which make up a table Field do have entries. The values of these entries are the values for each row in the table, separated by a comma followed by a space.
- The value of the ID field is undefined.

Note that values for a menu Field must match the possible values in the Field's definition. Other values are silently discarded.

ADVANCED TOPICS

ID: manage_system_configuration/non-admin_closed

CDATE: 2003/02/11

AUTHOR: Norm Liu

TESTAREA: Configuration, Installer, Help_Screens

CLASS: 2.10, 2.05, 2.04, 2.03, 2.02, 2.01

OBJECTIVE: To restrict ApTest Manager to Administrators only.

ASSERTIONS: When a user opts to limit access to Administrators only, non-administrators can NOT access the system, and a screen stating only Administrators are allowed to access is displayed.

ATM_ID: 18EAF496-2BBA-11DA-BE24-A10058D67DFA

OUTPUT: Verify the non-administrator user can not use ATM, and a window displays a message stating that only administrators can use ATM.

ENVIRONMENT:

PROCEDURE:

Log into ATM with administrator privileges.

Select the user name from the menu bar.

Select the "Manage System Configuration" link.

In the System Configuration table, select the "Yes" radio button for restricting user access to Administrators only.

Press the "Update Configuration" button.

Log out by selecting the "Log out" link.

Log in to ATM using a non-administrator user name.

TESTCYCLE: Unit, Integration, System, Smoke

PLANNEDTIMESTAFF: 5

FILENAME: datasheet.html, WF6.jpg

FILEDESC: ApTest Manager Data Sheet, Screen shot

Figure 17 - Test Case File

3.3.1 COPYING TESTS AND REQUIREMENTS BETWEEN TEST SUITES

Requirements and Test Cases can be copied between Test Suites by copying Requirement/Test Case files and Folder directories on the server (via the shell or command prompt). This should be done with great care when copying between Test Suites with different Requirement/Test Case Field definitions. Data in Requirements/Test Cases for Fields that are not defined for a Test Suite are discarded.

Note: once Requirements/Test Cases have been copied from a Test Suite they may not be copied back to that Suite. Further, Requirements and Test Cases may not be copied this way within a Suite, nor may entire Test Suites be copied; the copy features of the GUI must be used for these operations.

3.3.2 REGISTERING CHANGES

When Requirement/Test Case files are changed ApTest Manager must be informed in order for it to pick up the changes and reflect them in its database. This applies any time Requirement/Test Case files are copied or modified other than from the GUI.

The script `syncdb` in the `ATM_ROOT/bin` directory causes any Requirement and Test Case files that have been changed to be reflected in the ApTest Manager database. Run this from the command line and pass it the name of the Test Suite that has been modified as an argument, e.g. `syncdb MyTestSuite`. It prints the number of Requirement/Test Case files it encountered for that Test Suite. There is also a GUI equivalent: Manage->Synchronize the Test Suite.

The `syncdb` script can also be invoked with an optional `-r` argument. This causes it to also refresh the Test Sets and Test Sessions in the Test Suite. It is equivalent to using ApTest Manager's refresh feature on all the Test Sets and Test Sessions in the Test Suite.

3.4 EXPORTING SUITES

To package up all the Test Suites and configuration files from an ApTest Manager instance in an archive and restore them use the scripts `ATM_ROOT/bin/exportAll.pl` (see also Section 2.4.9) and `ATM_ROOT/bin/restore`. The export script produces a zip that is server OS-neutral and can be unpacked using the restore script onto this or another instance of ApTest Manager, on a different server for instance. See the scripts' internal documentation for details.

The steps to package up the data in a portable manner are:

```
% cd ATMDIR
```

```
% bin/exportAll.pl myarchive
```

This produces a file `myarchive.zip`.

ADVANCED TOPICS

It is not necessary to have users log out of ApTest Manager during the execution of the export command as database locking is performed automatically where required.

Note that this archive includes three configuration files:

`config.pl` – the installation settings. This file is server specific and should NOT be restored to overwrite the `config.pl` file on another server.

`data/configData` – the system configuration settings (from the Administrator's Manage System Configuration screen). These settings are largely server-neutral and restoring this file to a new server causes the settings it contains to be used on that server. Some settings, for example the mail server to use and the URL for bug tracking, may need to be modified.

`data/userData` – the user account database. This information is server-neutral and restoring this file to a new server causes the user accounts it contains to be valid on that server.

To unpack the archive:

```
other% cd ATMDIR
```

```
other% perl bin/restore.pl -u -a mybackup.zip
```

This step restores `data/userData` and all the Test Suites. See the script's internal documentation for additional options.

```
other% perl INSTALL.pl -u
```

This step does an integrity check on all the Test Suites etc. and ensures they are ready for use on the new server.

3.5 BACKING UP APTEST MANAGER FILES

ApTest Manager can be backed up using standard OS or third-party backup tools. Be certain that all users are logged out of ApTest Manager when making a backup to ensure there is no data alteration once the process is underway.

The best way to backup ApTest Manager with an external tool is to backup all the files in the directory tree starting at `ATM_ROOT`. However an individual Test Suite may be backed up by backing up its tree under the `suites` directory (i.e. `suites/<test suite name>.ts`).

To create a system-neutral data-only archive (which thus can be restored to a different server if necessary) see Section 3.4.

3.6 MOVING TO A NEW SERVER

To migrate your ApTest Manager instance to a new server:

ADVANCED TOPICS

Following the web site instructions, download the same version of ApTest Manger currently being used and do a complete new install (not an upgrade) on the new server. To also do a version upgrade, wait until after migration has been completed successfully. Doing an upgrade and migrating at the same time is not recommended as it muddies the waters in the event of problems with the new server.

Archive your data and restore it on the new server, as per Section 3.4 above.

Note that you cannot just copy ApTest Manager or its data files from one system to another using other methods. Also note that an ApTest Manager license allows for one ApTest Manager instance only.

3.7 USER EXTENSIONS

3.7.1 JAVASCRIPT EXTENSIONS

It is possible for a user to customize ApTest Manager through the use of templates for reports and data entry. Especially in the case of data entry (editing, and execution) it may be useful to add user-defined javascript functions that are called from Fields in the templates. An example might be an embedded timer that tracks how long a test is executed, and then populates the staff run time Field automatically when the result is posted.

It is possible to embed a javascript function directly into templates. However, this means that in some instances the javascript code is repeated many times in the generated content (e.g., in the run many screen or the results report). Further, editing the javascript in templates is challenging. Finally, if the same javascript is to be used in multiple templates it will need to be duplicated in each.

In order to address these challenges, ApTest Manager allows a user to define a local collection of javascript functions. If provided, these functions are automatically included in all ApTest Manager pages. All a user needs to do is create a file called "local.js" in the top level of their ATM tree. This file should contain javascript functions, and may also contain initialization code that is executed during load. The file is loaded after other ApTest Manager javascript is loaded and initialized. Consequently, the code has access to certain ApTest Manager provided javascript variables that can be used in user-provided functions:

- IMAGE_TOP - the URI path to the folder of ApTest Manager images.
- SCRIPT_TOP - the URI path to the folder of ApTest Manager scripts.
- WEB_TOP - the URI path to the top of the ApTest Manager tree.
- ATM_STATE - indicates what ApTest Manager is currently doing. Normally its value is "idle". When a template is in use for reporting, its value is "report", for execution "run", viewing a test case "view", and when editing "edit".

ADVANCED TOPICS

In addition to these variables, there are javascript utility functions available for general use. These are in the file ATMDIR/scripts/atm.js and their documentation is included therein. Of these utility functions, the most useful may be addEvent, which allows a function to attach an event handler to an element.

3.7.2 CSS EXTENSIONS

If a valid css file named local.css is added to the top level directory of the ApTest Manager tree, the styles it contains are available for use in templates. Care should be exercised not to conflict with the styles in atm.css.

3.7.3 ADDING A LOGO TO REPORTS

To add a logo to the menu bar of ApTest Manager reports, place the logo into the file ATMDIR/images/locallogo/gif. The logo should be 71 pixels high or less. The logo will be displayed to the left of the ApTest Manager logo.

3.8 SCRIPTS

The ApTest Manager bin directory contains several scripts that can be executed from the command line on the server to perform various functions. These scripts may be employed as-is, or as examples programming with the ApTest Manager API that can be used as starting points for developing scripts for other functions.

NAME	FUNCTION
addAtmUser	Add a user to the system
archiveSuite.pl	Create a zip archive of a test suite
buildHistory.pl	Create execution history entries from notes in test sessions
checkSessions.pl	Check to ensure sessions in a test suite are not damaged
checkSQL	Check the status of SQL integration for a suite
checkUUID	Check for duplicate UUIDs in a suite
cleanHiddenFiles	Removed unused hidden files from a test suite

ADVANCED TOPICS

convertFormatted	Convert formatted fields to wysiwyg fields
convertHTML	Convert an HTML or wysiwyg field to a formatted field
countTests	Determine that number of tests that match given selection criteria
delAtmUser	Remove a user from the system
encryptPasswords	Encrypted any unencrypted passwords in the system
export	Make a portable package of some your ATM data
exportAll.pl	Make a portable package of all ApTest Manager data
exportSuite.pl	Make a portable package of a Test Suite
exportUsers	Export user data
importResults.pl	Import session results from CSV file (e.g. from test automation)
importFromCSV	Import requirements and test from a CSV file
readCache	Read a cache associated with one of a test suite's databases
readdb	Read a Test Suite's test case database
readReportDB	Read a Test Suite's report database
readReqDB	Read a Test Suite's requirement database
readSession	Read and display a Test Session

ADVANCED TOPICS

readSet	Read and display a Test Set
readSuiteDb	Read a suite metadata database
readUserDb	Read the user database
restore	Restore from the output on exportAll.pl
syncdb	Synchronize a Test Suite's test database with the file system tree
syncSQL	Synchronize the SQL datastore for one or more Test Suites
updatefields	Update field contents in bulk
updatePrefs.pl	Update user preference structures
updateProfile.pl	Ensure that the test suite profiles are up to date
updateSets.pl	Ensure that the set definition is up to date. Once you restore a session folder from a backup, you need to run the updateSets.pl script on the test suite in question - this will reattach the session to its associated test set. Otherwise the system doesn't know that it is there.
updateSuite.pl	Ensure that the test suite and its cases are up to date
updateUsers.pl	Update fields of type "user" in a database

To see documentation on a script change the working directory to the ApTest Manager bin directory and enter the command:

```
perldoc <scriptname>
```

4 SQL DATASTORE

ApTest Manager keeps its data in an embedded database (see Section 3.1). As well, data may optionally be mirrored in a relational database management system that supports the Structured Query Language (SQL), in an 'SQL datastore'.

A datastore is automatically updated by ApTest Manager as tests are created, modified, and executed so the datastore contains the latest Test Suite data.

Data in an SQL datastore can be queried with tools that understand SQL; from scripting languages to data mining applications. Many tools, open source and commercial, are available. See the documentation for a specific tool for information on its operation.

4.1 SUPPORTED DATABASES

ApTest Manager supports SQL datastores on a number of relational database management systems.

The ApTest Manager Installation Instructions list the database products and versions supported.

The SQL supported by these databases varies. See the documentation for a specific database for information on the variant of SQL it supports and how to use it.

4.1.1 DATABASE INSTALLATION

There are two prerequisites for the procedures in this Chapter.

- A relational database must be installed and operating.

See a database's documentation for how to install, configure, and administer it.

Note that RDBMS installation, configuration, and administration tasks are NOT performed by ApTest Manager. ApTest does not provide support (in any way, shape or form) for performing these tasks.

- A Perl driver for the database must be installed.

See the *ApTest Manager Installation Instructions* for installation procedures for Perl drivers needed to communicate with each RDBMS.

4.2 DATASTORE OPERATION

An SQL datastore represents data in tables (relations). ApTest Manager constructs and maintains a series of tables that represent the Test Cases and Test Sessions for a Test Suite.

Each Test Suite can have its own SQL datastore, with tables reflecting the Test Case Fields, Execution Fields, and Session variables for the Suite.

Once a database is operational and a Perl driver has been installed for it, a two-step setup process is performed by the user through the ApTest Manager GUI to set up the datastores to be maintained on the database:

- Configure SQL datastore support for the ApTest Manager installation. It is disabled by default when ApTest Manager is installed.
- Initialize the Datastores; creating their relational database tables.

Once it is initialized, a datastore's tables are automatically updated by ApTest Manager as tests are created, modified, and executed so the datastore contains the latest Test Suite data.

If a Test Suite configuration is modified its SQL datastore needs to be re-initialized so its tables reflect the new configuration.

Note that data flows only from ApTest Manager into an SQL datastore, not from the datastore into ApTest Manager. So if a datastore is restored from backup, for example, it has no impact on ApTest Manager.

4.3 SQL DATASTORE CONFIGURATION

An ApTest Manager installation's support for SQL datastores is configured from the Administrative users' management screen (see Section 2.4). Click **Manage SQL Configuration** to set this configuration.

The SQL Support setting is a menu of 3 possible values that controls the installation's support of SQL datastores:

- Disabled: datastores are disabled for all Test Suites. This is the default value.
- Enabled - Global: an SQL datastore is enabled for each Test Suite and configured globally. A single RDBMS is used.
- Enabled - per Test Suite: an SQL datastore is enabled and configured separately for each Test Suite. A separate RDBMS can be used for each datastore.

Note that when using exportAll or exportSuite to archive Suite(s), when the archive is unpacked the SQL datastore(s) for the Suite(s) are disabled. Copying Test Suites by other means, e.g. zip or tar, is not recommended as their SQL datastores will not be disabled, though the database configured for them may well not be present when the Test Suites are unpacked.

4.3.1 GLOBAL SQL DATASTORE CONFIGURATION

If Enabled Global is selected as the value of SQL Support then an SQL datastore is automatically enabled for each existing Test Suite and for Test Suites that are created subsequently.

A single database is used for all Test Suites' SQL datastores. The type of database to be used and how to connect to it must be configured. This varies depending on the type of database selected. For most databases a host, port, user name, password, and database/system identifier are necessary. The list of databases presented for an installation consists of those supported databases for which a driver has been installed so that ApTest Manager can communicate with it. Databases not on the list are not properly configured for use by this ApTest Manager installation.

Tables created for each Test Suite's datastore have the Test Suite name automatically prefixed to their names. With multiple datastores residing on the same database this allows their tables to be distinct. Otherwise the data from one Test Suite would overwrite that from another.

The global database connection is automatically tested and any errors reported when it is configured.

4.3.2 PER TEST SUITE SQL DATASTORE CONFIGURATION

If Enabled – per Test Suite is selected as the value of SQL Support then an SQL datastore is separately enabled and configured for each Test Suite.

To setup a Test Suite's SQL datastore, click the Manage icon from the Menu Bar for the Test Suite and then click **Manage the Test Suite's SQL Datastore** and then **Configure Test Suite SQL Datastore**.

The SQL Datastore setting is a menu of 2 possible values that controls the Test Suite's support of an SQL datastore:

- Disabled – the SQL datastore for the Test Suite is not enabled. This is the default value when a Test Suite is created when Enabled – per Test Suite has been selected as the value of SQL Support (the datastore is enabled by default if Enabled – Global is the selected value).
- Enabled – the SQL datastore for the Test Suite is enabled.

The type of database to be used and how to connect to it must be configured. This varies depending on the type of database selected. For most databases a host, port, user name, password, and database or system identified are necessary.

The same or different database type and connection information can be configured for different Test Suites. If the same database is configured for multiple Test Suites, multiple datastores will reside on that database. If different databases are configured for different Test Suites different datastores will reside on the different databases.

SQL DATASTORE

For most databases a Table Name Prefix may be specified. This is prefixed to the names of all the tables in the Test Suites SQL datastore. If multiple datastores reside on the same database this allows their tables to be distinct. Otherwise the data from one Test Suite will overwrite that from another.

A database connection is automatically tested and any errors reported when it is configured

4.4 INITIALIZING SQL DATASTORES

If global SQL datastore support is configured, datastores are enabled for all Test Suites.

If per-suite SQL datastore support is configured then datastores are enabled by the user on a per Test Suite basis.

Each enabled Test Suite datastore needs to be initialized:

- The datastore must be initialized before it is used, in order to create the tables that store the data for its Test Suite.
- The datastore must be re-initialized when it's Test Suite's configuration changes. This recreates the datastore's tables so they correspond to the new configuration.
- The datastore must be re-initialized when a connection to the database cannot be established for a transaction or is lost mid-transaction. In such a case the datastore's tables must be recreated so they are synchronized with the current ApTest Manager data. When a connection problem occurs E-mails are sent automatically to ApTest Manager administrators who have the `Suite changed` email notification selected (see Section 2.4.4.21).

Datastores must be initialized by the user – they are not initialized automatically.

4.4.1 INITIALIZING SQL DATASTORES FOR MULTIPLE TEST SUITES

Click **Initialize SQL datastores** from the Administrative users' management screen (see Section 2.4) to initialize the datastores for multiple Test Suites. This link is not active if SQL datastores are disabled for this installation.

A table of the Test Suites that have their SQL datastore enabled is shown. The Test Suites for which datastores are to be initialized may be selected/deselected by clicking the checkbox next to their name. All Test Suites with datastores can be selected/deselected by clicking the checkbox in the table header row.

On entry to this screen those Test Suites whose SQL datastores need initialization are automatically selected. I.e. those Suites for which a) the datastore is enabled and b) the datastore does not match the Test Suite's configuration or the datastore has never been initialized.

Click **Initialize** to perform the datastore initialization for the selected Test Suites.

SQL DATASTORE

Note: The initializations may take some time, depending on connection speed, the size of the Test Suites, etc.

The **bin/syncSQL** script from the server command line can also be used to initialize all Test Suites' SQL datastores.

4.4.2 INITIALIZING A SINGLE TEST SUITE'S SQL DATASTORE

To initialize a specific Test Suite's SQL datastore, click the Manage icon from the Menu Bar for the Test Suite and then click **Manage the Test Suite's SQL Datastore** and then **Initialize Test Suite SQL Datastore**. These links are not present if SQL datastores are disabled for this installation.

Note: The initialization may take some time, depending on connection speed, the size of the Test Suite, etc.

4.5 DATASTORE LAYOUT

ApTest Manager keeps each SQL datastore up to date with information about the Test Cases for a Test Suite and the results of running those Test Cases. This data is kept in a collection of SQL "tables" that are unique to the Test Suite and its configuration. SQL queries can be made against these tables to retrieve information.

To see the exact table structure of the SQL datastore for a Test Suite click the Manage icon from the Menu Bar for the Test Suite and then click **Manage the Test Suite's SQL Datastore** and then **View Test Suite SQL Datastore Schema**.

Depending on how SQL datastore support is configured a prefix may be applied to each table name to ensure it is distinct from other Test Suite's datastore tables. E.g., a prefix of MySuite would cause the table of users to be called MySuite_REQ instead of REQ. Prefixes are not included in the table names given below.

The datastore contains the following tables. Each table contains a number of rows, one for each record in the table, and a number of columns, one for each data field a table row. Each column has a data type associated with it that indicates how the column's data is stored. The data type is specific to the database used. See the database's documentation for details.

4.5.1 TABLES RELATED TO REQUIREMENTS

- Table REQ contains a row for each Requirement in the Test Suite. The table is indexed by the "atm_rid" field, a series of characters that identifies Requirement Case and is unique within the Test Suite.

Columns in the REQ table contain data for the Requirement fields configured for the Test Suite. Fields' of type multi-select menu, userlist, table, or modification history table can have multiple values per Requirement and have their own tables, as described below.

- For each Requirement field of type multi select menu and type userlist with style multi select there is a REQ_L_* table. These tables' names end with the name of the Requirement field.

Each table has a row for each value selected from the field for each Requirement. A table is indexed by the Requirement's atm_rid and the atm_index field, which contains the numerical index of the selected value (i.e. the index of the first selected value for each Requirement is 0, the index of the second is 1, etc.). The value is contained in its own column (with the same name as the Requirement field).

- For each Requirement table and modification history table field there is a REQ_T_* table. These tables' names end with the name of the Requirement table field.

Each table has a row for each row in a table field for each Requirement. A table is indexed by the Requirement's atm_rid and the atm_tablerow field, which contains the numeric index of a row within a Requirement table field. Each REQ_T table row has a column for each Requirement field that is in the table field (e.g., if an ApTest Manager table field "files" contains Requirement fields "filename" and "filedesc", a corresponding SQL table would be named "REQ_T_FILES" and it would have columns "atm_id", "atm_tablerow", "filename", and "filedesc").

4.5.2 TABLES RELATED TO TEST CASES

- Table TC contains a row for each Test Case in the Test Suite. The table is indexed by the "atm_id" field, a series of characters that identifies the Test Case and is unique within the Test Suite.

Columns in the TC table contain data for the Test Case fields configured for the Test Suite. Fields' of type multi-select menu, userlist, table, or modification history table can have multiple values per Test Case and have their own tables, as described below.

- For each Test Case field of type multi select menu and userlist with style multi select there is a TC_L_* table. These tables' names end with the name of the Test Case field.

Each table has a row for each value selected from the field for each Test Case. A table is indexed by the Test Case's atm_id and the atm_index field, which contains the numerical index of the selected value (i.e. the index of the first selected value for each Test Case is 0, the index of the second is 1, etc.). The value is contained in its own column (with the same name as the Test Case field).

- For each Test Case table and modification history table field there is a TC_T_* table. These tables' names end with the name of the Test Case table field.

Each table has a row for each row in a table field for each Test Case. A table is indexed by the Test Case's atm_id and the atm_tablerow field, which contains the numeric index of a row within a Test Case table field. Each TC_T table row has a column for each Test Case field that is in the table field (e.g., if an ApTest Manager table field "files" contains Test Case fields "filename" and "filedesc", a corresponding SQL table would be named "TC_T_FILES" and it would have columns "atm_id", "atm_tablerow", "filename", and "filedesc").

4.5.3 TABLES RELATED TO TEST SESSIONS

- Table TS contains a row for each Test Session in the Test Suite. The table is indexed by "atm_sessionid", a test-suite unique Session identifier.

Columns in the TS table contain data for one of the Session Variables configured for the Test Suite (with names prefixed by "atm_sv_") and the standard fields defined for all Test Suite's Sessions:

FIELD	MEANING
description	Session description
numtests	The number of Test Cases in the Session
numuntested	The number of Test Cases in the Session that have not yet been executed
locked	If the Session is locked (0 = no, 1 = yes)
mtime	The time a test in the Session was last executed
summary	Session summary (name)
testset	The Test Set from which the Test Session was created
user	The account ID of the user who last executed a test in the Session

Session Variables of type multi select menu and the Session's assignedto field can have multiple values per Test Session and have their own tables, as described below.

- For each Session Variable of type multi select menu and the Session's assignedto field there is a TS_L_* table. These tables' names end with the name of the variable (with names prefixed by "atm_sv_") or field.

Each table has a row for each value selected from the field for each Test Session. A table is indexed by the Test Session's atm_sessionid and the atm_index field, which contains the numerical index of the selected value (i.e. the index of the first selected value for each Test Session is 0, the index of the second is 1, etc.). The value is

contained in its own column (with the same name as the Session Variable (with names prefixed by "atm_sv_") or "assignedto").

4.5.4 TABLES RELATED TO TEST SESSION RESULTS

- Table SD contains a row for each Test Case in a Test Session containing information on its last execution in the session. The table is indexed by a Test Case's "atm_id" or "atm_tid" (its numeric index within the session), and a Test Session's "atm_sessionid".

Columns in the SD table contain data for the user-defined Execution Fields for the Test Suite and the standard session execution fields:

FIELD	MEANING
atm_tid	The Test Case's index in the Session's ordered list of tests)
user	The account ID of the user who last executed the test in the Session
exectimeclock	The actual clock time to run the Test Case in the Session (only significant if the plannedtimeclock Test Case field is defined for the Test Suite)
exectimestaff	The actual staff time to run the Test Case in the Session (only significant if the plannedtimestaff Test Case field is defined for the Test Suite)
id	The Test Case's id
note	Execution information from the last time the Test Case was run in the Session
notetext	Execution note from the last time the Test Case was run in this Test Session
notes	Execution history for the Test Case in the Session
results	The result of the last time the Test Case was run in the Session
when	When the Test Case was last run in the Session

SQL DATASTORE

Execution fields of type multi-select menu and the assignedto field for a Test Case can have multiple values per Test Case in a Test Session and have their own tables, as described below.

- Table SD_T_ATM_AUDIT is identical to the SD table but contains a row for each time a Test Case is executed in a Test Session. The table is indexed by a Test Case's "atm_id" or "atm_tid" (its numeric index within the session), and a Test Session's "atm_sessionid" and the execution's "when" field (as each execution of a test Case has a unique time stamp)..

Columns in the SD_T_ATM_AUDIT table contain data for the user-defined Execution Fields for the Test Suite and the standard session execution fields, as described for the SD table.

Note that values are stored for all suitable execution fields, not just those with the "audit trail" flag set. That flag just governs which fields appear in the audit trail in the GUI.

- For each Execution field of type multi-select menu and the assignedto field for a Test Case in a Test Session there is an SD_L_* table. These tables' names end with the name of the field.

Each table has a row for each value selected from a multi-select menu Execution field and the assignedto field for each Test Case in each Test Session. A table is indexed by the Test Session's atm_sessionid, the Test Case's atm_id or atm_tid, and the atm_index field, which contains the numerical index of the selected value (i.e. the index of the first selected value for each Test Case in a Test Session is 0, the index of the second is 1, etc.). The value is contained in its own column (with the same name as the Execution field or "assignedto").

4.5.5 TABLES RELATED TO USER IDS

- SQL table fields that contain usernames (e.g., for Test Case fields of type user, muser, userlist) store the user's account ID, not their full name. Account IDs can be looked up and mapped to a full name and other user information using the SQL datastore table ATM_USERINFO (e.g., SELECT atm_fullname FROM ATM_USERINFO where atm_userid = "someUserId").

The access level associated with each user is stored in the SQL datastore table ATM_AUTH (e.g., SELECT atm_authlevel FROM ATM_AUTH where atm_suitename = "someSuite" and atm_userid = "someUserID")

4.5.6 TABLE RELATED TO TEST SUITE INFORMATION

- Metainformation about the test suite is available in the table ATM_SUITEINFO. This table is indexed by suiteaname. Fields in the table include:

FIELD	MEANING
-------	---------

SQL DATASTORE

atm_suitename	Test Suite Name
description	Suite description
atm_prefix	The SQL table prefix used for this suite's tables.

4.5.7 NOTES

- Date information is stored in the context of the underlying database, using whatever format it uses to represent dates. See the database's documentation for details.
- SQL table names may not contain periods or hyphens. If a period or hyphen is present in a table name prefix it is thus removed when creating a datastore's tables.
- How to determine what tables and columns exist and how to count the rows in a table depends on the database being used. See the database's documentation for details.

Access Level	2-9	textarea.....	1-32
execute and edit	2-8, 2-14	ExportAll script	3-4
no access	2-8, 2-13	ID	
run any test	2-8, 2-13	convert string to numeric	1-2
run assigned	2-8, 2-13	renumber	1-2
suite manager	2-8, 2-14	Styles	1-21
view reports	2-8, 2-13	IEEE 829	xii
view tests and reports	2-8, 2-13	LDAP	
Administrator	2-1	accounts	2-2
Automatic Numbering		Cleanup Removed LDAP Accounts ...	2-3
numbering by folder	1-21	configuration	2-18
numbering by suite	1-21	Convert Accounts to LDAP	2-3
outline numbering	1-21	Synchronize All LDAP Accounts	2-3
Backup	3-5	Licensing	2-21
Closed to non-administrative users	2-12	Login message	2-12
Convert a Test Suite to use Requirements	1-3	Moving to a new server	3-5
Database	3-1	Normal user	2-5
Default Access	2-13	Problem Report	
Email notifications	2-2, 2-10	configuration	2-12
Execution Field		Report Logo	3-7
flags	1-33	Requirement	
styles	1-32	configuration	1-10
types	1-29	create from Test Case	1-3
Execution Field Flag		Requirement Field	
audittrail	1-33	flags	1-24
Depends On	1-33	special fields	1-12
hidden	1-35	styles	1-21
mandatory	1-35	types	1-15
readonly	1-35	Requirement Field Flag	
reset on rerun	1-35	Depends On	1-24
selectable	1-35	mandatory	1-25
settable	1-35	Name part	1-25
Execution Field Style		readonly	1-25
date and time	1-32	Requirement Field Reserved	
date only	1-32	atm_mcomments	1-15
formatted	1-32	atm_owner	1-15
html	1-33	Requirement Field Special	
wysiwyg	1-23, 1-33	atm_average_result	1-12
Execution Field types		atm_best_result	1-12
date	1-30	atm_locked	1-12
multi-select menu	1-30	atm_testcount	1-13
single-select menu	1-31	atm_tests	1-13
text	1-31	atm_worst_result	1-13

INDEX

author.....	1-13	Session Variable Type	
cdate.....	1-13	date.....	1-51
id 1-13		date and time.....	1-51
mdate.....	1-14	derived.....	1-51
muser.....	1-14	heading.....	1-51
Requirement Field Style		multi-select menu.....	1-52
auto outline number.....	1-21	single-select menu.....	1-52
auto-number by folder.....	1-21	text.....	1-53
auto-number by suite.....	1-21	textarea.....	1-53
code.....	1-23, 1-32	SQL databases.....	4-1
date and time.....	1-22, 1-51	SQL datastore.....	4-1
date only.....	1-22, 1-51	configuration.....	4-2
formatted.....	1-23	layout.....	4-5
html.....	1-23	Syncdb script.....	3-4
minutes24.....	1-22	System Configuration.....	2-11
minutes8.....	1-22	Template.....	1-35
multi-select.....	1-22	check for errors.....	1-37
number.....	1-22, 1-24	data entry.....	1-39
plain.....	1-21	errors.....	1-36
single select.....	1-22	report.....	1-37
Requirement Field Type		time tracking.....	1-44
date.....	1-16	Test Case	
file.....	1-16	configuration.....	1-10
modification history table.....	1-17	copy.....	3-4
multi-select menu . 1-15, 1-18, 1-29, 1-30,		create from Requirement.....	1-4
1-50, 1-52		Test Case Field	
single-select menu.....	1-19	flags.....	1-24
table.....	1-19	special fields.....	1-12
text.....	1-20	styles.....	1-21
textarea.....	1-20	types.....	1-15
userlist.....	1-20	Test Case Field Flag	
Revision control.....	3-1	Depends On.....	1-24
scripts.....	3-7	mandatory.....	1-25
Scripts.....	3-7	Name part.....	1-25
Session Variable		readonly.....	1-25
flags.....	1-53	Test Case Field Reserved	
types.....	1-50	atm_mcomments.....	1-15
Session Variable Flag		atm_owner.....	1-15
Depends On.....	1-53	Test Case Field Special	
derivedfrom.....	1-55	atm_locked.....	1-12
derivedfunc.....	1-55	atm_reqcount.....	1-12
hidden.....	1-55	atm_reqlink.....	1-13
mandatory.....	1-55	author.....	1-13
readonly.....	1-55	cdate.....	1-13
selectable.....	1-55	id 1-13	

INDEX

mdate	1-14
muser	1-14
Test Case Field Style	
auto outline number	1-21
auto-number by fodler	1-21
auto-number by suite	1-21
code	1-23, 1-32
date and time.....	1-22, 1-51
date only	1-22, 1-51
formatted	1-23
html.....	1-23
minutes24	1-22
minutes8	1-22
multi-select	1-22
number.....	1-22, 1-24
plain	1-21
single select	1-22
Test Case Field Type	
date	1-16
file.....	1-16
modification history table	1-17
multi-select menu . 1-15, 1-18, 1-29, 1-30,	
1-50, 1-52	
single-select menu	1-19
table	1-19
text	1-20
textarea	1-20
userlist.....	1-20
Test Email Configuration	2-22
Test Suite	
configuration	1-6
configuration editors	1-7
copy	1-2
create requirements from tests	1-3
create tests from requirements	1-4
delete.....	1-2
description.....	1-2
design.....	1-8
enable requirements	1-3
export and restore.....	3-4
rename.....	1-2
renumber	1-2
synchronize	1-5
Time style.....	2-2
Time Tracking	1-44
Timezone	
configuration	2-2, 2-14
User	
Administrator	2-1
normal	2-5
User Extensions	
CSS	3-7
javascript.....	3-6
Work Day Length.....	1-44, 1-45