



Test Management Tools Series

ApTest™ Manager Admin Guide

TEST MANAGEMENT TOOLS SERIES

ApTest Manager Admin Guide

Version 2.14
March 2006

Copyright © 2000-2006 - 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 763-786-8157 • Fax 763-786-8180

www.aptest.com

Significant Revisions

Revision	Date	
Version 2.14	March 2006	
Split off from User's Guide		

Table of Contents

PREFACE	VI
MANUAL ORGANIZATION	VI
DOCUMENTATION SET	VI
STYLISTIC CONVENTIONS	VI
CUSTOMIZATION	VII
1 MANAGING TEST SUITES	1-1
1.1 CHANGE SUITE DESCRIPTION	1-1
1.2 COPY TEST SUITE	1-1
1.3 RENAME TEST SUITE	1-1
1.4 SYNCHRONIZE TEST SUITE	1-1
1.5 DELETE TEST SUITE	1-2
1.6 TEST SUITE CONFIGURATION	1-2
1.6.1 <i>Configurable Elements</i>	1-2
1.7 CONSIDERATIONS FOR TEST SUITE DESIGN	1-3
1.7.1 <i>Defining Test Case Fields</i>	1-4
1.7.2 <i>Defining Test Case Selectors</i>	1-4
1.7.3 <i>Defining Test Cases</i>	1-4
1.7.4 <i>Defining Test Sets</i>	1-5
1.7.5 <i>Defining Test Session Fields and Variables</i>	1-5
1.7.6 <i>Defining Test Results</i>	1-6
1.7.7 <i>Defining Templates</i>	1-6
1.8 CONFIGURING TEST CASE FIELDS	1-6
1.8.1 <i>Test Case Field Editor</i>	1-6
1.8.2 <i>Test Case Field Attributes</i>	1-7
1.8.3 <i>Test Case Field Types</i>	1-8
1.8.4 <i>Test Case Field Styles</i>	1-12
1.8.5 <i>Test Case Field Flags</i>	1-15
1.8.6 <i>Reserved Test Case Fields</i>	1-16
1.9 CONFIGURING SESSION RUN DATA	1-17
1.9.1 <i>Run Data Editor</i>	1-17
1.9.2 <i>Run Data Attributes</i>	1-17
1.9.3 <i>Run Data Types</i>	1-18
1.9.4 <i>Run Data Styles</i>	1-19
1.9.5 <i>Run Data Flags</i>	1-21
1.10 CONFIGURING TEMPLATES	1-23
1.10.1 <i>Missing Field Errors in Templates</i>	1-24
1.10.2 <i>Check Templates for Errors</i>	1-24
1.10.3 <i>Report Templates</i>	1-25
1.10.4 <i>Data Entry Templates</i>	1-26
1.10.5 <i>Template Editor</i>	1-27
1.10.6 <i>Table manipulation</i>	1-27

1.10.7	<i>Field Names in Templates</i>	1-31
1.10.8	<i>Special Session Fields in Templates</i>	1-31
1.11	TIME TRACKING FIELDS.....	1-32
1.12	CONFIGURING RESULT CODES.....	1-34
1.13	CONFIGURING SESSION VARIABLES.....	1-36
1.13.1	<i>Session Variable Editor</i>	1-36
1.13.2	<i>Session Variable Attributes</i>	1-37
1.13.3	<i>Session Variable Types</i>	1-37
1.13.4	<i>Session Variable Flags</i>	1-38
2	ADMINISTRATION	2-1
2.1	NORMAL USER ADMINISTRATIVE FEATURES	2-2
2.1.1	<i>Email Notifications</i>	2-3
2.2	ADMINISTRATOR ADMINISTRATIVE FEATURES	2-3
2.2.1	<i>Setting a Timezone</i>	2-4
2.2.2	<i>Restricting Test Suite Access</i>	2-5
2.2.3	<i>Email Notifications</i>	2-7
2.2.4	<i>System Configuration</i>	2-8
2.2.5	<i>Profile Catalog Management</i>	2-13
2.2.6	<i>Viewing Login and License Information</i>	2-13
2.2.7	<i>Viewing Test Suite Information</i>	2-14
2.2.8	<i>Archive System Data</i>	2-14
2.2.9	<i>Check for Updates to ApTest Manager</i>	2-14
3	ADVANCED TOPICS	3-1
3.1	DATABASE.....	3-1
3.2	TEST CASE FILES	3-1
3.2.1	<i>Updating Tests</i>	3-2
3.2.2	<i>Copying Tests between Suites</i>	3-2
3.2.3	<i>Registering Changes</i>	3-4
3.3	EXPORTING SUITES.....	3-4
3.4	BACKING UP APTEST MANAGER FILES	3-5
3.5	USER EXTENSIONS.....	3-5
3.5.1	<i>Javascript Extensions</i>	3-5
3.5.2	<i>CSS Extensions</i>	3-6
3.6	SCRIPTS	3-6

Table of Figures

Figure 1 - Edit Test Case Fields screen	1-8
Figure 2 - Edit Session Run Data screen	1-17
Figure 3 - Display of Edit Template with Missing Fields	1-24
Figure 4 - Edit Report Templates screen	1-25
Figure 5 - Edit Template screen	1-29
Figure 6 - Example Header/Footer table	1-30
Figure 7 - Edit Result Codes screen	1-35
Figure 8 - Edit Session Variables screen	1-36
Figure 9 - Click Username to Access Administration	2-1
Figure 10 - Normal User's Edit Account screen	2-2
Figure 11 - Administrator's Management screen	2-3
Figure 12 - Administrator's Change Existing Account screen	2-6
Figure 13 - Administrator's Change Suite Access Permissions screen	2-7
Figure 14 - Test Case File	3-3

Preface

Manual Organization

This manual is divided into three 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.

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.

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 contain a red X and these functions cannot be accessed.

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

1.1 Change Suite Description

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

1.2 Copy 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, and Test Session results may optionally also be copied.

1.3 Rename Test Suite

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

1.4 Synchronize Test Suite

This function updates ApTest Manager's internal database for the Test Suite to match the contents of its Test Case files (see Section 3.2 for details on these files).

1.5 Delete Test Suite

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

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

1.6 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. 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 with. The configuration of a Test Suite can however be added to the Profile catalog, allowing this configuration to be used when creating more Test Suites 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 Test Cases are entered. This ensures complete information is entered for each Test Case and avoids introducing incompatible Test Case formats after the fact.

1.6.1 Configurable Elements

The configurable aspects of a Test Suite are:

- Test Case Fields define the information which makes up each of a Test Suite's 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 for a Test Case are the author, creation date, test procedure, required preconditions, etc.

- Run Data 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 Data Entry templates. This information is associated with a particular run of a Test Case rather than the Test Case itself. Examples of the sort of information a Run Data Field may be used to record are Problem Reports submitted for the execution of a Test Case.
- Templates define how Test Case and Run Data Fields are formatted in ApTest Manager reports and screens: which Fields are displayed and how they are laid out.
Separate Templates are provided for the screens used to 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 Test Case and Session information separately for different areas of ApTest Manager.
- 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.
- 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.

Template configuration utilizes a variant of the wysiwyg text field editor. Other configuration screens utilize a variant of the table Field, controlling a table of items with its standard row controls:



moves the row up



moves the row down



deletes the row (if there is only one row it cannot be deleted)



creates a duplicate of the current row



inserts a new row above



inserts a new row below

1.7 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 in configuring ApTest Manager.

1.7.1 Defining Test Case Fields

Test Suites may be configured to have any number and type of Fields in Test Cases. ApTest Manager contains several sample sets of 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 an example 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 example with a unique set of Fields that match the organization's specific requirements.

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

1.7.2 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 appear in the Test Set. For example Test Cases in a specific language or that apply to a particular build or type of test cycle. It is safe to change which Fields are configured as selectors at any time.

1.7.3 Defining Test Cases

When a Test Case is created it is given a name, as are the Folders into which Test Cases are grouped.

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

It also useful to remember that ApTest Manager will present Folders and Test Cases in alphabetical order by default. Although tests in Test Sets and Sessions can be reordered, it is useful to name Test Cases and Folders so they are ordered reasonably by default. Prepending numbers to the names (e.g. what will be shown first starts with a '1', next in order a '2', etc.) is a commonly used technique.

1.7.4 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 projects composed of different groups of tests used for 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 requires 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.

1.7.5 Defining Test Session Fields and 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.

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 Fields (which are known as Run Data) 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 Run Data are the Problem Reports created based on executing a Test Case.

Session Variables and Run Data 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 Run Data 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. Run Data Fields are also reported in ApTest Manager

reports and can be used to limit reports to Test Cases with specific Run Data 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 Run Data Fields for any special information testers will record when executing tests.

1.7.6 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 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 associated with a run of a Test Case in a Session.

1.7.7 Defining Templates

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

1.8 Configuring Test Case Fields

1.8.1 Test Case Field Editor

The Test Case Fields editor specifies the Test Case Fields for a Test Suite. 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 saved.

The Fields shown in Figure 1 are part of one of the Profiles in the catalog shipped with ApTest Manager. The meaning and format of the Fields are described below.

Once defined, Test Case Fields need to be included in a Test Suite template in order to be used. When included in a report template the Field's current value is shown. When included in another template a form element is presented in which to enter the value.

1.8.2 Test Case Field Attributes

Each Test Case 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 Fields defined by users. Also, names that end in 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 in a Test Case.
SIZE	Describes the size of the form element used for this Field when editing the Test Case.
STYLE	Defines how the contents of the Field are formatted when the Test Case is displayed.
VALUES	Defines default Field values.
FLAGS	Indicates properties of the Field.

Edit Test Case Fields

Manage User: [Sue Lake](#) [logout](#)
 Back to [Manage Suite](#)

Test Suite: 3_ApTest_Manager [Suite Manager]

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 6 of the ApTest Manager User's 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	b/a	n/a	selectable
cdate	Creation Date	creation date	n/a	n/a	n/a	selectable
author	Created By	user	10	n/a	n/a	selectable
atm_owner	Assigned Test Developer	userlist	10	single select	n/a	selectable mandatory
testarea	Features Tested	multi-select menu	4	plain	*Reporting, Installer, Help_Screens, Admin, Configuration	readonly mandatory selectable dependson
testcycle	Test Cycles	multi-select menu	4	plain	*Unit, Integration, System, Compatibility, Installation, Performance, Configuration	readonly mandatory selectable dependson

Figure 1 - Edit Test Case Fields screen

1.8.3 Test Case Field Types

Values for the type of a Test Case Field and their significance are:

cdate

When editing or viewing a Test Case, the creation date of the Test Case. The value of this type of Field cannot be edited.

date

When editing a 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'. When

viewing a Test Case the value of a date Field is a string representing the date selected for the Field in the Test Case.

file

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

SIZE must be a single numeric value > 1.

When viewing a Test Case the value of a file Field are links to the files entered for the Field in the Test Case.

ID

When editing or viewing a Test Case, the ID of the Test Case (the folders containing it and its name, separated by slashes). The value of this type of Field cannot be edited.

SIZE specifies the height of a menu of Folders displayed for this Field on a query screen, e.g. on the Define Test Set screen (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.

mdate

When editing or viewing a Test Case, the date of the last modification of the Test Case (through the GUI). When a Test Case is created, this value is initialised to the creation date. The value of this type of Field cannot be edited.

modification history table

A Field type that supports integrating a modification history into a 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	Modification History	modification history table	AUTO	n/a	mdate, muser, mcomments	n/a
	mdate	Last Modification Date	last modification date	n/a	n/a	n/a	selectable
	muser	Last Modified By	last modifying user	20	n/a	n/a	selectable
	mcomments	Change Details	textarea	AUTOx5	formatted		readonly mandatory selectable

This Field is similar to a Field of type table but has special semantics as follows:

- Only the first row of the table can be edited
- Rows cannot be reordered - basically, no table controls are available
- When a test case is copied, this revision history is erased

A revision history table Field cannot contain another table or revision 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 revision 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 Test Case the values in the VALUES column values are displayed as a multi-selection menu. SIZE specifies the number of values displayed at once. Default values (which are assigned to new Test Cases as they are created) may be specified by placing an asterisk before a value in the VALUES column. When viewing a Test Case the value of a multi-select menu Field is the values selected for the Field in the Test Case.

muser

When editing or viewing a Test Case, the user who last modified the Test Case (through the GUI). When a Test Case is created, initialized to the author of the Test Case. The value of this type of Field cannot be edited.

single-select menu

When editing a Test Case the values in the VALUES column values are displayed as a single-selection menu. SIZE specifies the number of values displayed at once. A default value (which is assigned to new 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. When viewing a Test Case the value of a single-select menu Field is the value selected for the Field in the Test Case.

To utilize the mandatory flag with a single-select menu Field and require the user to make a selection, define a default value for the Field which is blank. For example:

NAME	SELECTABLE	TYPE	STYLE	SIZE	LABEL	VALUES
priority	N	single-select menu	plain	1	Priority	High, Medium, Low

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

table

Strings in 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:

NAME	SELECTABLE	TYPE	STYLE	SIZE	LABEL	VALUES
Files	N	table	none	1	"Required Files"	"fileName, fileDesc"
fileName	N	text	plain	20	"File Name"	none
fileDesc	N	textarea	text	50x2	"File Description"	none

When editing a 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 while editing a Test Case. A table Field cannot contain another 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).

When viewing a Test Case a table is displayed with the values entered into each column of the table in the Test Case.

text

When editing a 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 when editing, and VALUES specifies the default value for the Field. When viewing a Test Case the value of a text Field is the value for the Field in the Test Case.

textarea

When editing a 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 maybe be set to AUTO (e.g. AUTOx2) in which case the size of the Field will automatically adjust to 95% of the size of the containing element (e.g. a table a cell). VALUES specifies the default value for the Field. When viewing a Test Case the value of a textarea Field is the value for the Field in the Test Case.

uuid

When editing or viewing a Test Case, the identifier of the Test Case. This is a 32 character string divided into 5 groups by hyphens. The value is unique across all Test Cases in all Suites and is used to identify Test Cases independent of their name. It is assigned when a Test Case is created and is never modified. The value of this type of Field cannot be edited.

user

When editing or viewing a Test Case, the author of the Test Case. The value of this type of Field cannot be edited.

userlist

When editing a 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. When viewing a Test Case the value of a userlist Field is the value(s) selected for the Field in the Test Case.

1.8.4 Test Case Field Styles

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

For menu and text Fields the style is generally set to "plain" though the minutes8, minutes24, and number styles are also valid.

minutes8

The value of the Field is expected to be an ordinal number and are rendered as hdm using 8 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.

minutes24

The value of the Field is expected to be an ordinal number and are 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 alphanumeric order.

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.

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.

ApTest Manager offers four styles of textarea Fields.

These styles format information in different ways when it is edited, viewed, or shown in a report.

The primary difference between styles has to do with whether or not HTML is entered directly to control how the browser displays the Field's information.

Entering HTML allows control of text formatting with HTML commands. However some common typing habits, such as hitting Enter and expecting a blank line when the test is viewed or executed, will not work in these Fields.

Non-HTML styles allow text to be entered without HTML it, much like typing in an editor, and help with how the browser formats this information. After editing a Test Case the Test Case can be viewed (by clicking its name on the Edit Tests screen) to ensure the information entered is formatted as expected.

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

Supported textarea Field styles and the way their contents are formatted are as follows. 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 source code or printable data into a Field.

formatted

This style allows text to be entered much like typing on a typewriter. Plus, lists can be created automatically.

- 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 "<").
- 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) will be 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.

1.8.5 Test Case Field Flags

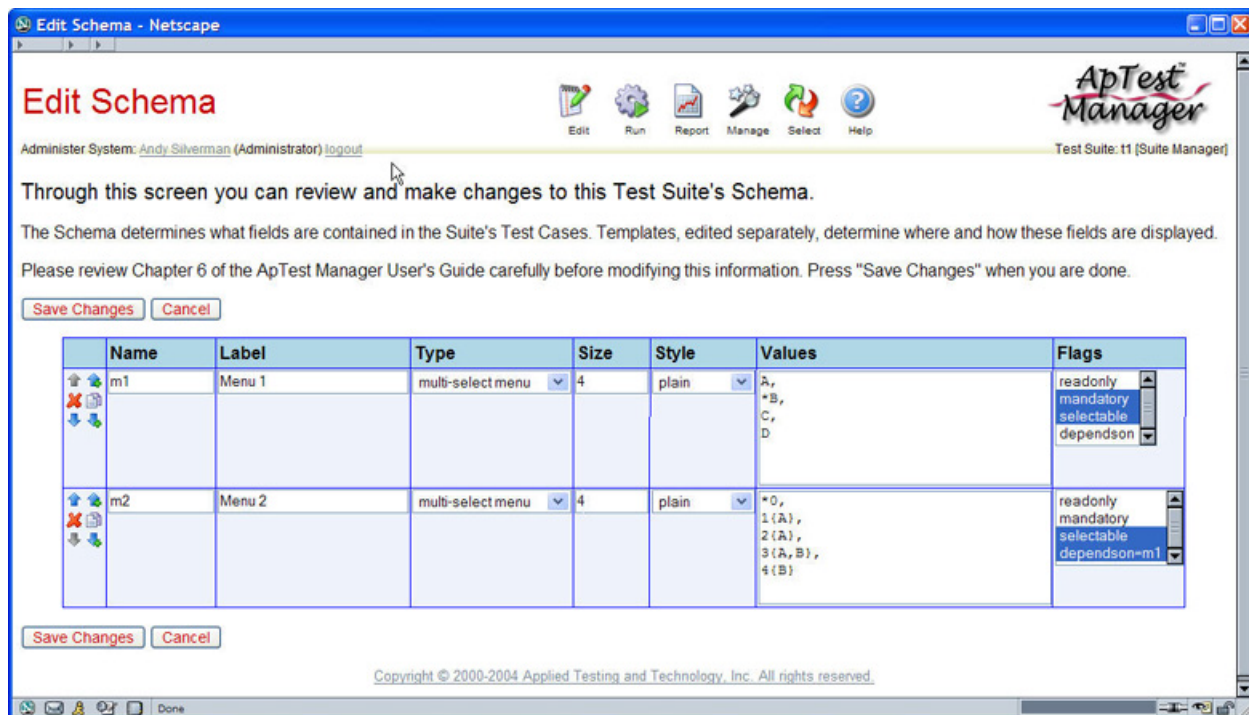
In absence of a flag, the indicated functionality does not apply to the Field.

dependson

The value of the flag should be another menu Field (single- or multi-select). 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



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 is a multi-select menu, then 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.

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

mandatory

The Field must be given a value in a Test Case. Mandatory Fields are indicated with a red asterisk when editing a Test Case.

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 current value is shown.

selectable

Indicates whether the Field can be used to select Test Cases when defining a Test Set. Only a few Fields should be specified as Selectable, least Test Set selection become overly complex. Fields of type table and type cdate cannot be made selectable.

1.8.6 Reserved Test Case Fields

Reserved Test Case Fields trigger special ApTest Manager behavior if they are used in the Test Suite configuration. The names of these Fields begin with atm_, and are thus reserved.

FIELD NAME	DESCRIPTION
-------------------	--------------------

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 intended to be used to capture the user(s) assigned to develop the Test Case. When specifying the value of this Field (i.e. 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 Execute and Edit or better. A report template can then be configured to list the Test Cases that have been assigned to each user.
------------------	---

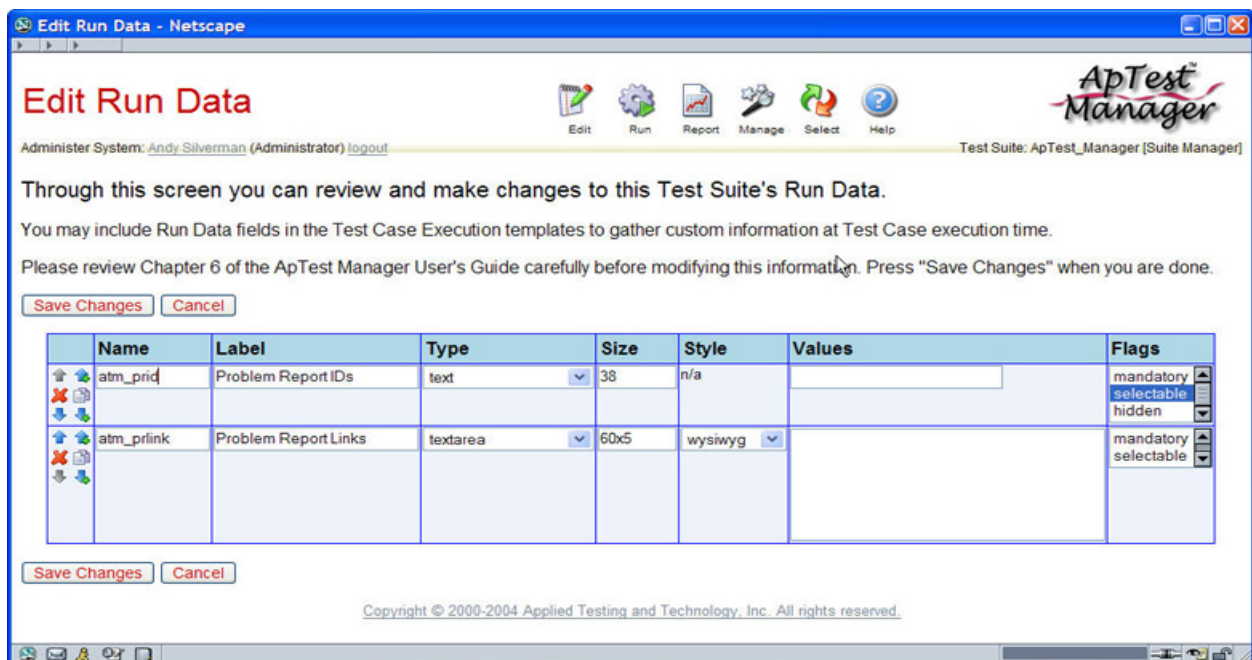
1.9 Configuring Session Run Data

1.9.1 Run Data Editor

The Session Run Data editor specifies Fields used to gather information during Test Case execution and store it in a Test Session.

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 saved.

Once defined, Run Data Fields need to be included in a template in order to be used. When included in a report template the Field's current value is shown. When included in a Test Case execution template a form element is presented that allows a value for the Field to be entered.



Edit Run Data

Administer System: [Andy Silverman \(Administrator\) logout](#) Test Suite: [ApTest_Manager \[Suite Manager\]](#)

Through this screen you can review and make changes to this Test Suite's Run Data.

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

Please review Chapter 6 of the ApTest Manager User's Guide carefully before modifying this information. Press "Save Changes" when you are done.

Name	Label	Type	Size	Style	Values	Flags
atm_prid	Problem Report IDs	text	38	n/a		mandatory selectable hidden
atm_prlink	Problem Report Links	textarea	60x5	wysiwyg		mandatory selectable

Copyright © 2000-2004 Applied Testing and Technology, Inc. All rights reserved.

Figure 2 - Edit Session Run Data screen

1.9.2 Run Data Attributes

Each Run Data 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 Fields defined by users. Also, names that end in 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 the 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	Various flags that relate to the Field.

1.9.3 Run Data Types

Values for the Type of a Run Data Field and their significance are:

date

A Field which contains a date/time value. When editing a test case a value may be selected from a calendar. A default value may be selected from a calendar. The default value may be cancelled by clicking the adjacent 'x'.

When executing a 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'. In reports the value of the Field the value of a date Field is a string representing the date selected for the Field in the Test Case.

multi-select menu

When executing a Test Case the values in the VALUES column values are displayed as a multi-selection menu. SIZE specifies the number of values displayed at once. Default values (which are assigned to new Test Cases as they are created) may be specified by placing an asterisk before a value in the VALUES column. In reports the value of the Field is the values selected from the menu.

single-select menu

When executing a Test Case the values in the VALUES column values are displayed as a single-selection menu. SIZE specifies the number of values displayed at once. A default value (which is assigned to new 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. 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 a default value for the Field which is blank. For example:

	priority	Priority	single-select menu	1	plain	High, Medium, Low	readonly
							selectable
							dependson

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

text

When executing a 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 when editing, and VALUES specifies the default value for the Field. When viewing a Test Case the value of a text Field is the text.

textarea

When executing a 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 maybe be set to AUTO (e.g. AUTOx2) in which case the size of the Field will automatically adjust to 95% of the size of the containing element (e.g. a table a cell). VALUES specifies the default value for the Field. When viewing a Test Case the value of a textarea Field is the text entered.

1.9.4 Run Data Styles

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

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.

ApTest Manager offers four styles of Run Data textarea Fields.

These styles format information in different ways when it is edited, viewed, or shown in a report.

The primary difference between styles has to do with whether or not HTML is entered directly to control how the browser displays the Field's information.

Entering HTML allows control of text formatting with HTML commands. However some common typing habits, such as hitting Enter and expecting a blank line when the test is viewed or executed, will not work in these Fields.

Non-HTML styles allow text to be entered without HTML it, much like typing in an editor, and help with how the browser formats this information. After editing a Test Case the Test Case can be viewed (by clicking its name on the Edit Tests screen) to ensure the information entered is formatted as expected.

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

Supported textarea Field styles and the way their contents are formatted are as follows. 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 source code or printable data into a Field.

formatted

This style allows text to be entered much like typing on a typewriter. Plus, lists can be created automatically.

- 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 "<").
- 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) will be 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.

1.9.5 Run Data Flags

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

dependson

The value of the flag should be another menu Field (single- or multi-select). 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.

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 is a multi-select menu, then 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.

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

Edit Run Data

Administer System: [Andy Silverman \(Administrator\)](#) [logout](#) Test Suite: t1 [Suite Manager]

Through this screen you can review and make changes to this Test Suite's Run Data.

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

Please review Chapter 6 of the ApTest Manager User's Guide carefully before modifying this information. Press "Save Changes" when you are done.

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

Name	Label	Type	Size	Style	Values	Flags
m1	Menu 1	multi-select menu	4	n/a	A, *B, C	mandatory selectable dependson
m2	Menu 2	multi-select menu	4	n/a	*0, 1 {A}, 2 {A}, 3 {A, B}, 4 {B}	mandatory selectable dependson=m1

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

Copyright © 2000-2004 Applied Testing and Technology, Inc. All rights reserved.

hidden

The Field is not presented to the user if referenced in a template. It may only be used when referenced a script.

mandatory

This Field must be given a value in a Test Session. Mandatory Fields are indicated with a red asterisk in Data Entry 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 current 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

This Field is available to select Test Cases for inclusion in reports on the Customize Report screen.

settable

This Field is available have its values set for a Test Set or Test Session with the Assign Test Set/Session screens. Setting this flag and 'reset on rerun' is not recommended, as any value assigned is reset when the test is run.

1.10 Configuring Templates

Templates define the ways in which Test Case information is presented in ApTest Manager screens and reports. Test Case or Run Data Fields must be included in templates in order to be displayed.

The content of a template is HTML markup and Test Case and/or Run Data 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 Test Case or Run Data Field into a template, the name of the Field is enclosed within a pair of delimiters, preceded by the string "FIELD" (e.g. <% FIELD_domain %>) for Test Case Fields or the string "RESULT" for (e.g. <% RESULT_version %>) for Run Data Fields. The Field is displayed based on how it is specified in the Test Case or Run Data Field definitions, as text, a table, etc.

In Report templates the data for a Field is displayed. In Data Entry templates (for editing or executing a Test Case) a form element is displayed allowing data to be entered or edited. In both 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.

- Fields can be added or deleted
- More or fewer rows or columns can be employed
- The order of rows or columns can be changed
- Some or all of the information can be formatted other than as a table, as an HTML list for example.

1.10.1 Missing Field Errors in Templates

If a Template contains a reference to a Field which is not defined the Field reference 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 3.

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

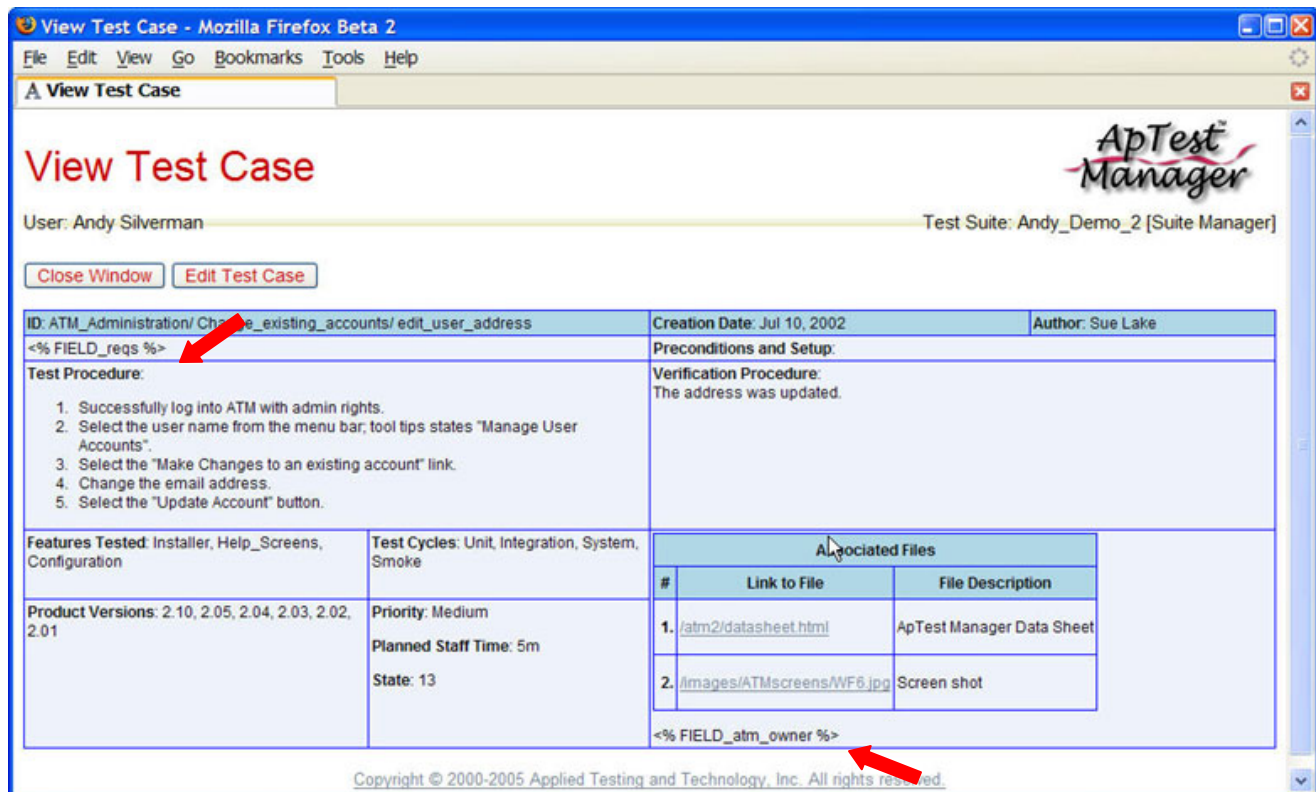


Figure 3 - Display of Edit Template with Missing Fields

1.10.2 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 checked for errors when they are saved.

1.10.3 Report Templates

Report templates specify the information to be displayed and the display format to be used for ApTest Manager reports that apply to a single Test Session. There can be an unlimited number of report templates, on a per Test Suite basis.

Each report template is included in the list of reports available on the Select Report Screen. As well, if they do not contain execution related information Templated reports can apply to all the tests in a Test Suite, and are available from the report list on the Edit Tests screen. ApTest Manager automatically determines which type of report a template is for and includes it in the appropriate list(s).

A report template defines the information displayed for a single Test Case. When information about several Test Cases is displayed in a report the template is applied to each Test Case. The report templates shipped with ApTest Manager often display Test Case information as a table. So, for example, a report covering ten 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 4 shows the screen for editing report templates. New templates can be created or an existing template can be modified, including editing, renaming, or copying it. There can be an unlimited number of report templates, customized to produce reports on different Test Case Fields.

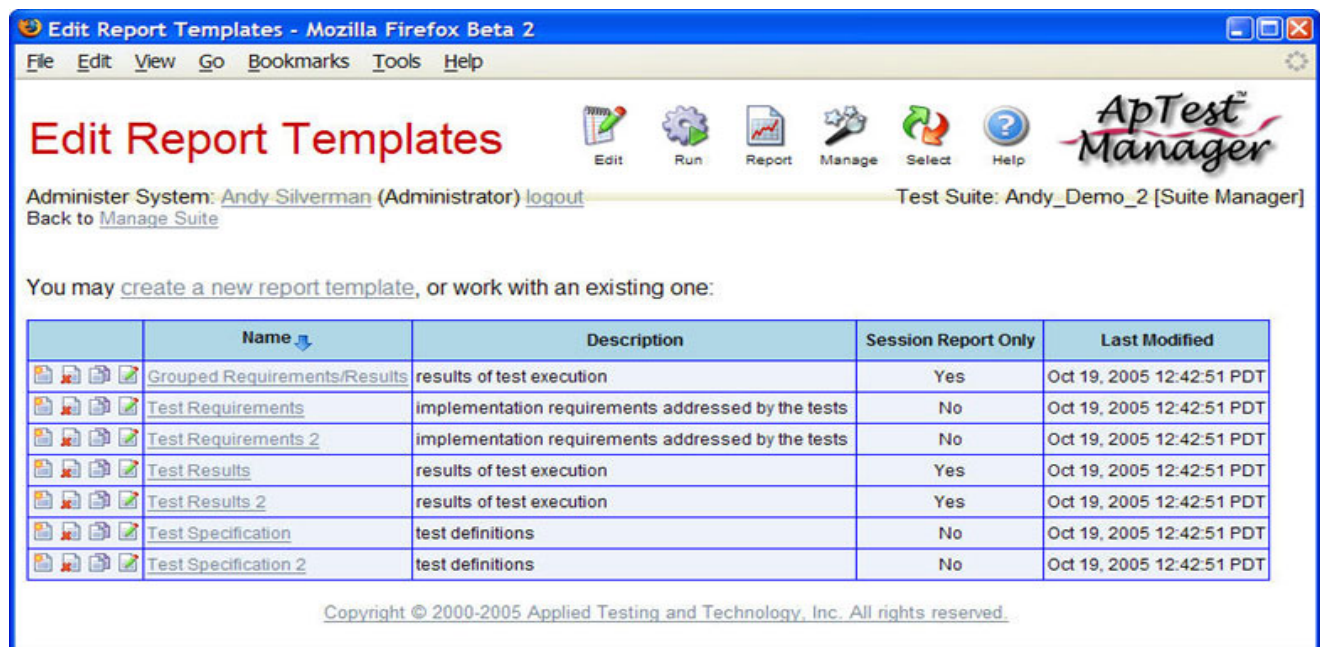


Figure 4 - Edit Report Templates screen

By default, six report templates are provided:

- Test Requirements Report and Report 2
- Test Specification Report and Report 2
- Test Results Report and Report 2

Two templates are provided for each report, allowing two different versions to be generated. One version of each report is defined with a Header/Footer element table (see Section 1.10.6.1) and presents overview data for each Test Case. A second version presents more complete data for each Test Case in a more complex format.

Report templates can be modified for a variety of reasons. For example to:

- Change what Fields are shown in a report. Generally, different Fields are included by the Templates for different reports. In a Requirements Report Template, for example, only those Fields that are requirements related are usually included, while a Test Results Report Template often includes Fields for the entire Test Case and its results.
- Modify the format used to display information, to display information in some other way or just to alter the layout of a table.
- 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.

1.10.4 Data Entry Templates

Three data entry Templates are provided for each Test Suite, one for the display of Test Case information when Test Cases are viewed and edited and two used when Test Cases are executed.

TEMPLATE	USED BY SCREENS
Test Editing	View Test Case screen Edit Test Case screen
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 Test Case. The Data Entry Templates specify what this Test Case information consists of and how it is displayed.

Data Entry Templates can be modified to change what Test Case and Run Data Fields they include. This might be done to:

- Change what Fields are shown when editing or running a test. Generally, different Test Case Fields are included by the Templates for these functions.
- Modify the format used to display information, to display information in some other way or just to alter the layout of a table.
- 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.

1.10.5 Template Editor













The Template Editor, shown in Figure 5, is used to create and modify templates. The Template Editor is a variant of the WYSIWG editor for Test Case Fields described in Section **Error! Reference source not found.** 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.

1.10.6 Table manipulation

Additional table related controls are provided in the Template Editor:

	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.

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 manipulated 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 menu.

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.

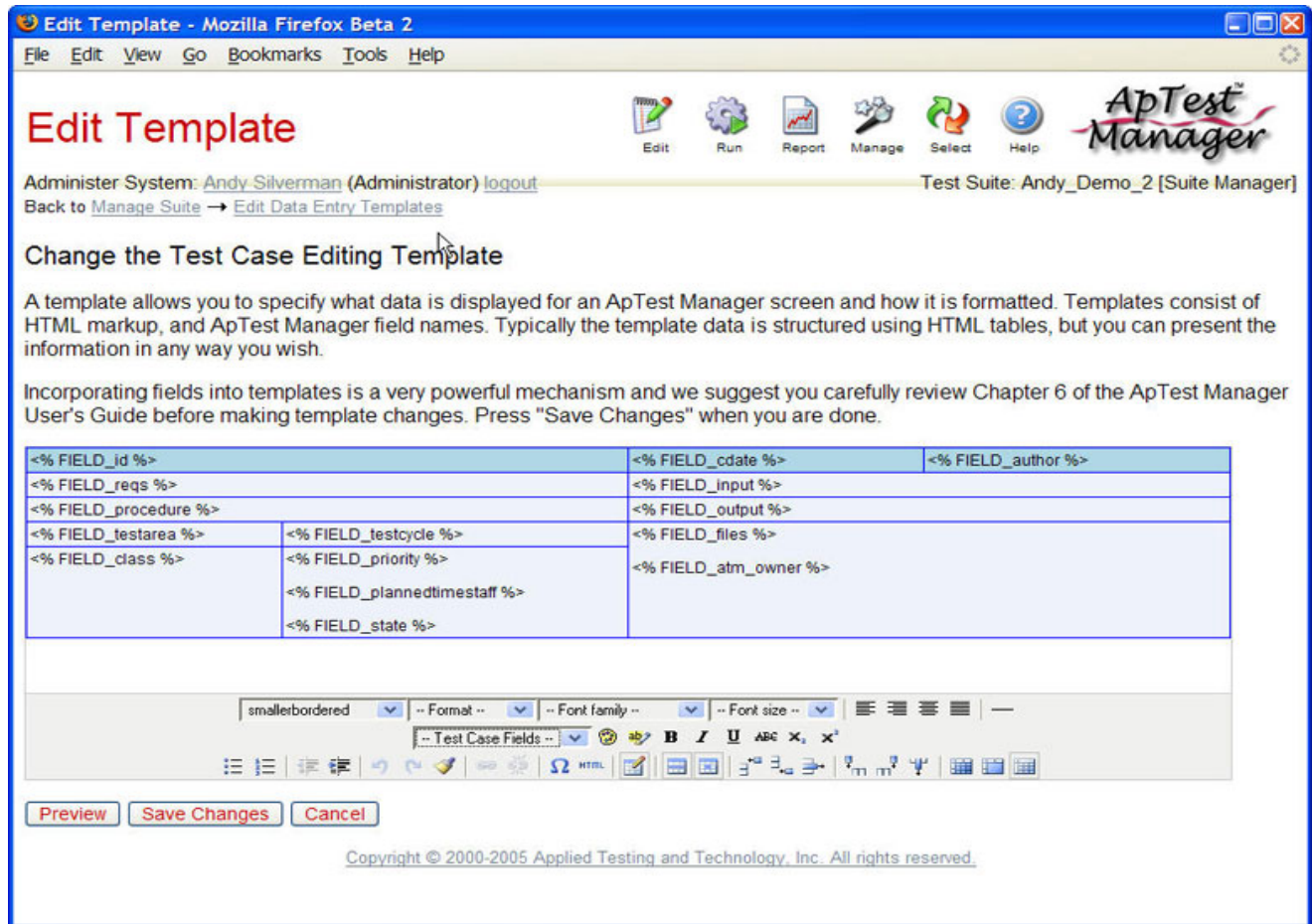


Figure 5 - Edit Template screen

1.10.6.1 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.

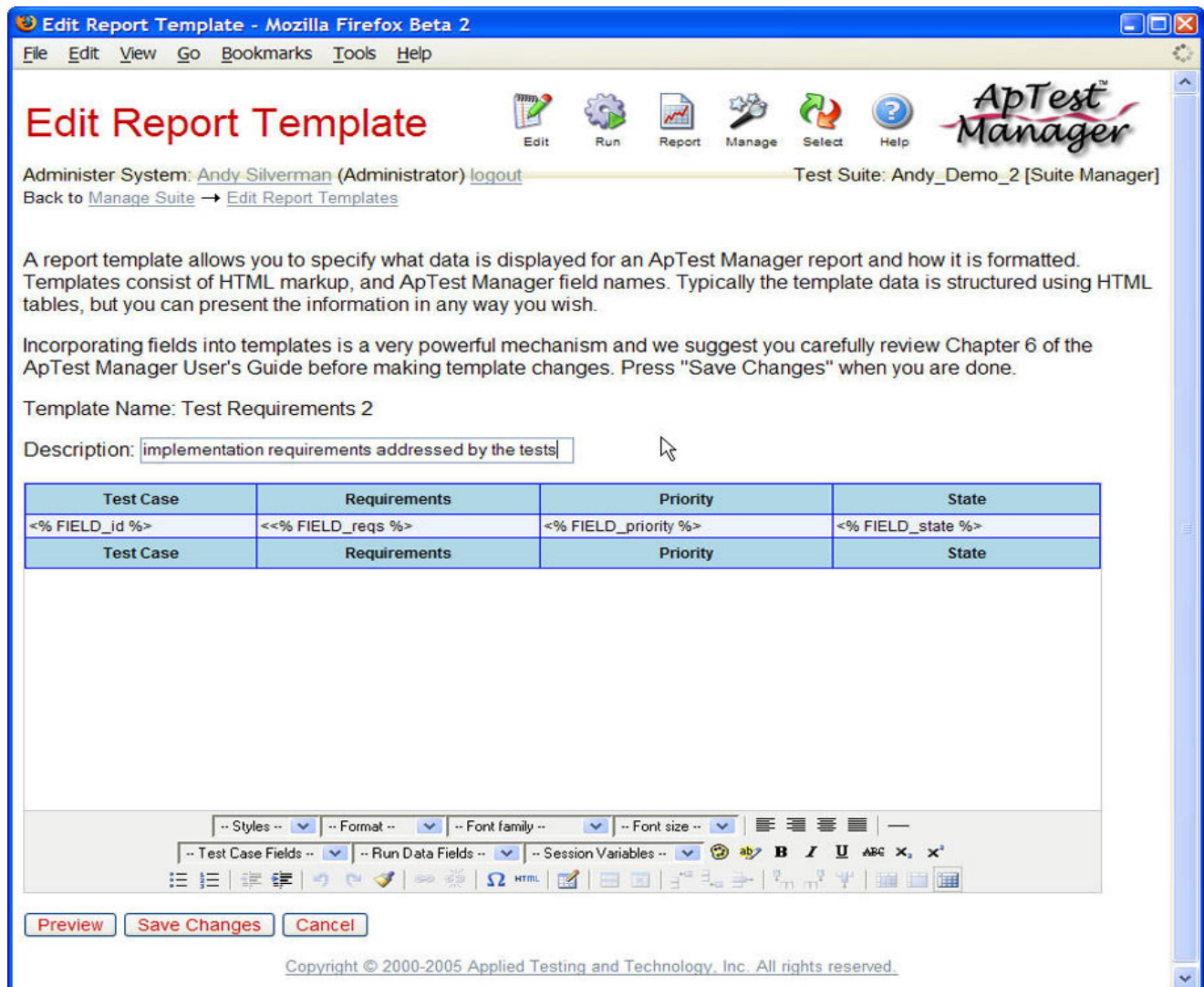


Figure 6 - Example Header/Footer table

1.10.7 Field Names in Templates

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

- **Test Case Fields** – this is a pulldown menu used to insert delimited user-defined Field names into templates. Selecting a Field from the menu cause it to be inserted at the cursor's location in the template.
- **Run Data Fields** (not used for the Test Case Editing template) – this is a pulldown menu used to insert delimited user-defined Run Data Field names and other runtime Field names into templates. Selecting a Field from the menu cause it to be inserted at the cursor's location in the template.
- **Session Variables** (not used for the Test Case Editing template) – this is a pulldown menu used to insert delimited user-defined Session Variable names and the names of other Session attributes into templates. Selecting a Variable from the menu causes it 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. Referencing them directly will result in errors. Thus these fields are not included in the list of test Case Fields in the Template Editor. This scenario will be flagged when a template is saved and must be corrected.

1.10.8 Special Session Fields in Templates

In addition to user defined Test Case and Run Data Fields, templates may also include some special Fields related to executing a Test Case. Such Fields can be used for instance in the templates for reporting on a Test Session to report this information.

These Fields have the suffix RESULT, like Run Data Fields. The supported Fields and how they are formatted when displayed are:

FIELD NAME	INFORMATION DISPLAYED	FORMATTING
RESULT_assignedto	Users to whom the test is assigned	text
RESULT_execTimeStaff	Actual staff time to execute the test	days,hours,minutes (8 hour days)
RESULT_execTimeClock	Actual clock time to execute the test	days,hours,minutes

RESULT_note	Execution information from the last time the Test Case was run in this Test Session	(24 hour days) list
RESULT_notes	Execution history for the Test Case in this Test Session	list
RESULT_notetext	Execution note from the last time the Test Case was run in this Test Session	list
RESULT_results	The result of the last time the Test Case was run	text
RESULT_user	The user who last ran the Test Case	text
RESULT_when	When the Test Case was last run	date and time

1.11 Time Tracking Fields

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

There are two sets of time Fields, one set for staff time (using an 8 hour day) 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. Some of the Test Suite Customization Profiles shipped with ApTest Manager contain the staff time Fields; others have no time tracking enabled.

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. Note that three of the Test Suite Customization Profiles shipped with ApTest Manager already are configured to track staff time.

- 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: Menu

STYLE: minutes8

SIZE: 1

LABEL: Planned Staff Time

VALUE: "1, 2, 5, 10, 30, 60, 120" (or other values for time to run a test, in minutes)

- Add the Field 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 line FIELD_exectimestaff to the Test Case Execution Template, directly above the line FIELD_execResults. This causes the user to be asked for the amount of staff time it took to actually execute a test.

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

- Placing the Field FIELD_plannedtimestaff in a report template to show the planned staff time
- Placing the Field RESULT_exectimestaff in a report template to show the actual staff time for a Test Session.

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: 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, in minutes)

- Add the Field FIELD_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.
- Add the line FIELD_exectimeclock to the Test Case Execution Template, directly above the line execResultsFIELD. This causes the user to be asked for the amount of clock time it took to actually execute a test.

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

- Placing the Field FIELD_plannedtimeclock in a report template to show the planned clock time.
- Placing the Field RESULT_exectimeclock in a report template to show the actual clock time for a Test Session.

1.12 Configuring Result Codes

Figure 7 shows the Edit Result screen used to specify the result codes for a Test Suite and select their name, description and associated color. 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 will not be possible to select reports on tests that match the result.


Result codes can be rearranged, so they are presented in the desired order in reports.

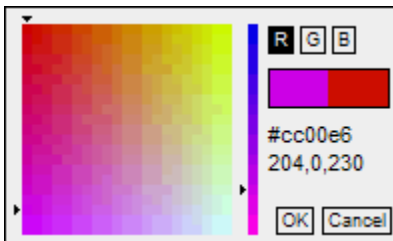
Please note the result code UNTESTED is used for those Test Cases which have not been run yet, and so that result should always be defined.

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.

It is desirable 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.

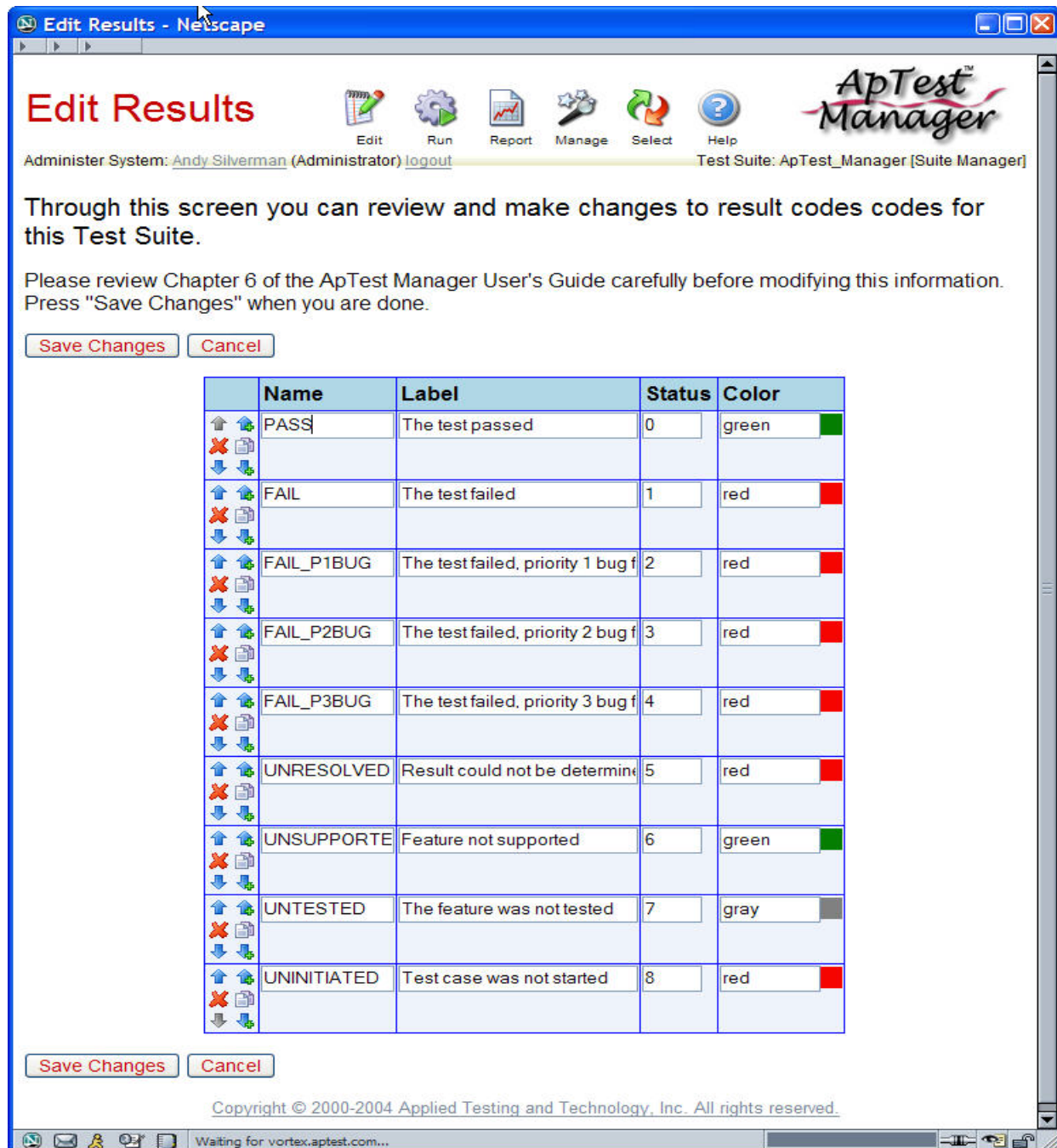


Figure 7 - Edit Result Codes screen

1.13 Configuring Session Variables

1.13.1 Session Variable Editor

The Session Variable editor specifies the characteristics of test environments configured for a Test Suite. Variables can be rearranged, so they are presented in a particular order. Attributes can be specified for each Variable. The Variables defined apply to Test Sessions for in a specific Test Suite.

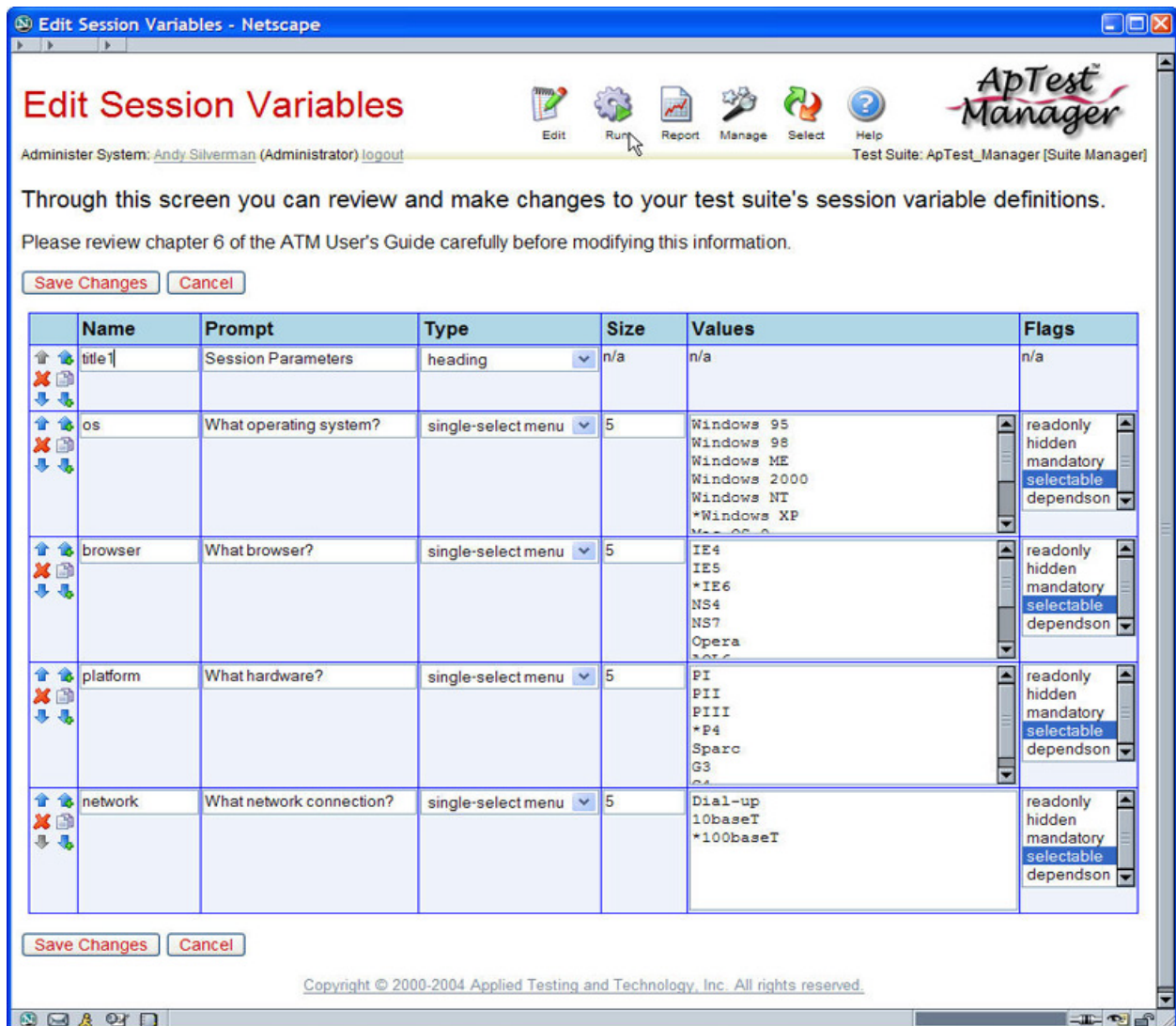


Figure 8 - Edit Session Variables screen

1.13.2 Session Variable Attributes

Each Session Variable has six attributes:

NAME	The name of the Variable. Variable names may only contain alphanumeric characters and underscores, 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 future use by ApTest.
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.

1.13.3 Session Variable Types

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

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 screen if it is 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 (i.e. on the Create Test Session screen the values in the VALUES column values are displayed as a multi-selection menu. SIZE specifies the number of

values displayed at once. Default values (which are assigned to new Test Cases 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 selected from the menu.

single-select menu

When specifying the value of the Variable (i.e. on the Create Test Session screen) these values are displayed as a single selection menu. 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 a default value for the Variable which is blank. For example:

	priority	Priority	single-select menu	1	plain	High, Medium, Low	readonly mandatory selectable dependson

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 (i.e. on the Create Test Session screen) 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 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 (i.e. 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 will automatically adjust to 95% of the size of the containing element (e.g. a table a 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.

1.13.4 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 Field (single- or multi-select). When editing a Test Case, the values displayed for this Field will 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.

	Name	Prompt	Type	Size	Values	Flags
	title1	Session Parameters	heading	n/a	n/a	n/a
	m1	Menu 1	multi-select menu	4	A, *B, C, D	readonly hidden mandatory selectable dependson
	m2	Menu2	multi-select menu	4	0, 1 {A}, 2 {A}, 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 is a multi-select menu, then 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 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. There is no AND and no NOT.

derivedfrom

Specifies the names of one or more Session Variables to be used to derive the value of a Variable of type `derived`. 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 `derived`. 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 `suites/derived/Derived.pm`. 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 `derivedfrom` 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 from in the menu 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 menu values. If a flag is deselected its value will be cleared.

hidden

The Variable will not be presented to the user as part of the set of Session Variables. It may only be used when 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 may not be changed. When a readonly Variable is displayed for edit a form element is not displayed, rather the Variable's current value is shown.

selectable

The Variable is used to select Test Sessions on the Select Report page.

2 Administration

ApTest Manager administrative functions are accessed by clicking on the Username of the currently logged in user, The Username is displayed on the Suite bar, next to a link to log out of ApTest Manager.

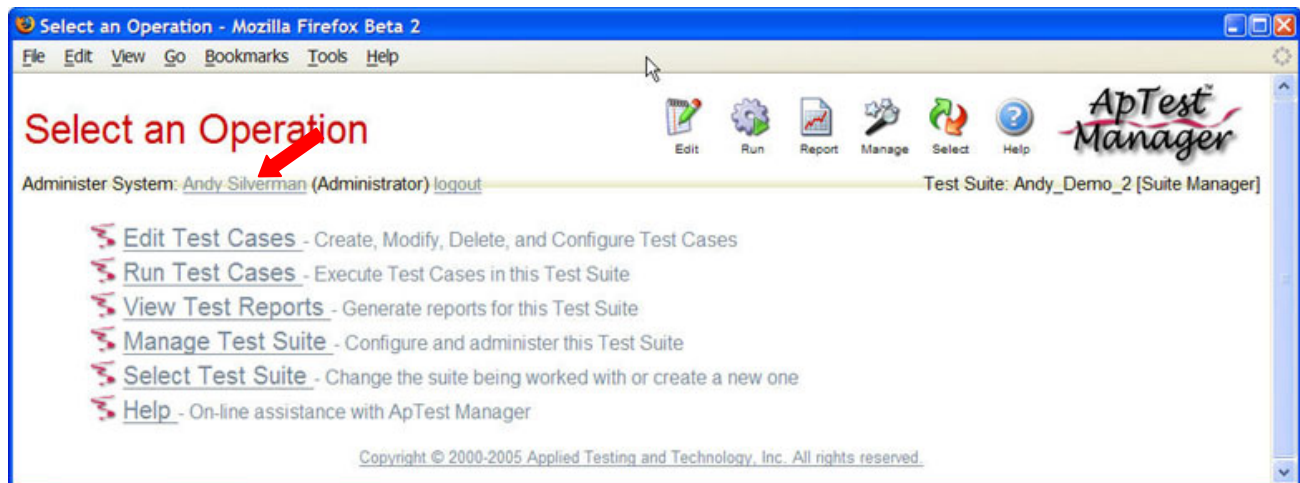


Figure 9 - Click Username to Access Administration

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 affect only 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 Normal User Administrative Features

Normal users can access to a screen for setting up their ApTest Manager account: setting their password, timezone, etc.

Click Update Account when done making changes to the account configuration.

Edit Account Information

Manage User: [Sue Lake](#) [logout](#) Test Suite: Another_Demo_Suite [Suite Manager]

Full name:

E-mail address:

Phone number:

Password (if you want to change it):

Password (again):

Rows per Page for reports:

Your timezone:

Email notifications on? ☒ Yes ☐ No

Suite	Suite email notifications							
	Edit	Manage		Run				
	Test case changed	Suite config file changed	Suite template file changed	Session changed	Session completed	Set changed	Tests reassigned	Tests reassigned to/from user
5PR_ApTest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ad_Hoc_Ltd	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ad_Hoc_Reporting_-_PM_v2.1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Andy_2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Another_Demo_Suite	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ApTest_Manager	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Copy_of_Sample	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Eval_Test_Suite	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
history	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
mysuite	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PR_ApTest	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
shane_copy_1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
shane_new	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
shane1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
standard	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
table	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
trying_search	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 10 - Normal User's Edit Account screen

2.1.1 Email Notifications

Email notifications are emails that ApTest Manager can send to automatically when specific events occur. A number of email notifications are available on the Edit Account screen 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.

The leftmost checkboxes toggle selection of all the notifications for a given Suite, while the checkbox below the name of an event toggles selection of that event for all Test Suites.

2.2 Administrator Administrative Features

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

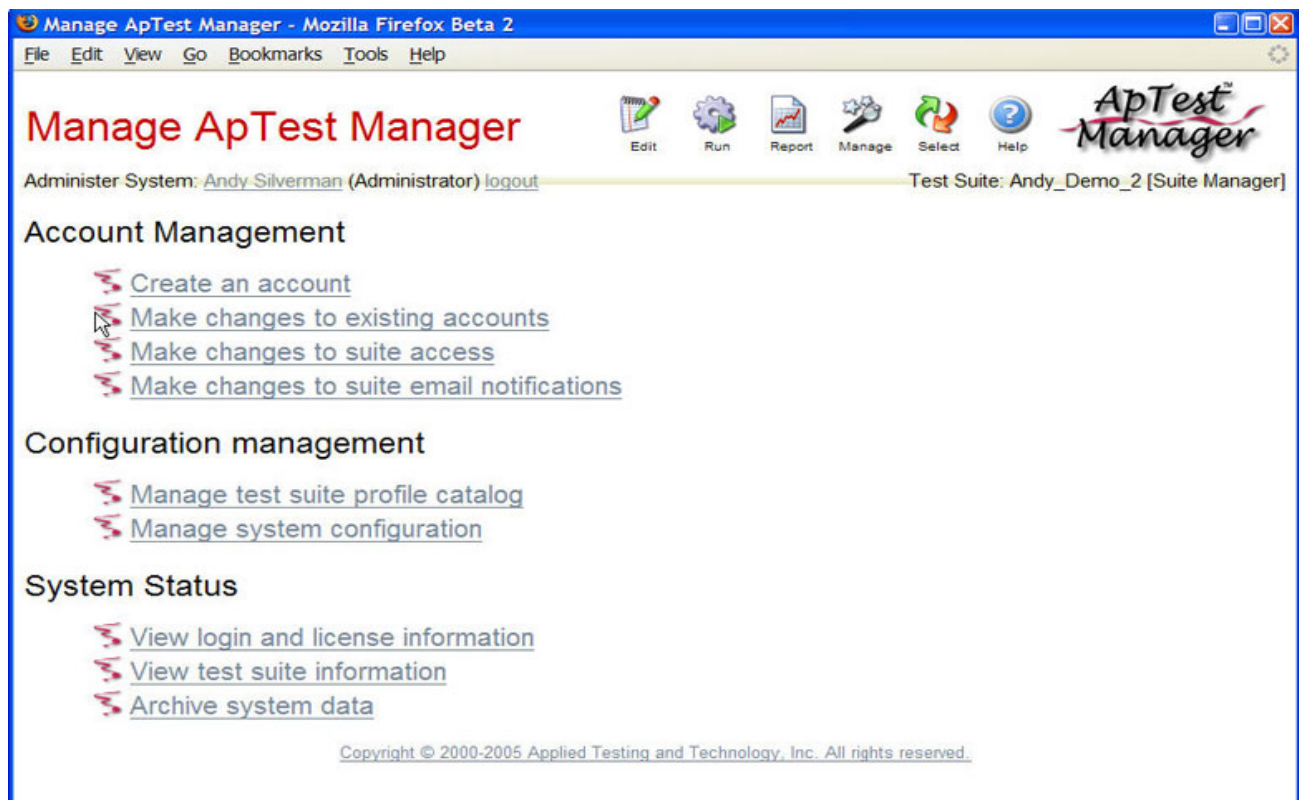


Figure 11 - Administrator's Management screen

ADMINISTRATION

These administrative functions are similar to the features provided to Normal users for creating and modifying their account information, with four important differences:

- An Administrator can create, delete, and modify accounts for other users. Note that an Administrator cannot delete their own account or that of the admin user.
- An Administrator can give other users Administrator privileges
- An Administrator can restrict the access of other users to specific Test Suites
- An Administrator can set special email notifications for themselves as well as set the email notifications of other users

As well, an Administrator can:

- Manually initiate a check for new versions of ApTest Manager, this check normally happens automatically
- Manage a variety of system configuration options
- Manage the contents of the catalog of Profiles for creating new Test Suites
- See information about an installation's license and the currently logged in users
- See information about all existing Test Suites
- Create a portable archive of all existing Test Suites

2.2.1 Setting a Timezone

2.2.1.1 Setting a User's Timezone

In the Administrator's "Make changes to existing accounts" screen a user's timezone can be set by selecting a region, and then a city therein. This defaults to a global configuration setting or, in the absence of the global setting, to the timezone of the server.

2.2.1.2 Allowing a user to set their own Timezone

A Normal user can go to their account administration page and set their preferred timezone. It defaults as above.

2.2.1.3 Setting a timezone during account creation

On the account creation page a timezone can be specified for the new user.

2.2.2 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.
- View Tests and Reports – the user can view reports and Test Cases but cannot edit or run tests.
- Run Assigned – the user can view reports 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 and execute tests but cannot edit tests or modify Test Sets or Sessions.
- Execute – the user can view reports, execute tests, and modify Test Sets and Sessions but cannot edit tests.
- Execute and Edit – the user can view reports, execute tests, modify Test Sets and Sessions, and edit Test Cases.
- Suite Manager – the user can view reports, 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 Execute and Edit access to it, other users are given the default access level (see Section 2.2.4.5).

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.

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

ADMINISTRATION

Make changes to existing accounts - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Make changes to existing accounts

Administer System: [Andy Silverman \(Administrator\)](#) [logout](#) Test Suite: copy [Execute and Edit]
[Back to Manage ATM](#)

Select an account:
 -- account list --
 Adams
 Adams32
 Adamson
 admin

or user name:
 -- user list --
 Al Lines
 Alan Reeson
 Albert Aronchik
 Alex Barnes

[Edit](#)

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

Account:	andy		
Full name:	Andy Silverman		
Last login:	Nov 14, 2005 08:50 PST		
E-mail address:	andy@aptest.com		
Phone number:			
Password (if you want to change it):			
Password (again):			
Rows per page for reports:	1000		
User's timezone:	America	Los_Angeles	

Feature access controls:	Feature	Yes	No
	Administrator	<input checked="" type="radio"/>	<input type="radio"/>

Test Suite	Suite access controls							
	Access Level							
	None select all	View Reports select all	View Tests and Reports select all	Run Assigned select all	Run Any Test select all	Execute select all	Execute and Edit select all	Suite Manager select all
2_ApTest_Manager	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
3_ApTest_Manager	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
4_ApTest_Manager	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5_ApTest_Manager	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
5PR_ApTest	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Ad_Hoc_Ltd	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Figure 12 - Administrator's Change Existing Account screen

ADMINISTRATION

Make changes to suite access

Administer System: [Andy Silverman \(Administrator\)](#) [logout](#)
[Back to Manage ATM](#)

Test Suite: copy [Execute and Edit]

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

-- suite list --
 2_ApTest_Manager
 3_ApTest_Manager
 4_ApTest_Manager
 5_ApTest_Manager

Access Permissions for suite Another_Demo_Suite

Account	User Fullname	Access?							
		None select all	View Reports select all	View Tests and Reports select all	Run Assigned select all	Run Any Test select all	Execute select all	Execute and Edit select all	Suite Manager select all
Adams	Joseph Adams	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Adams32	Steve Adams32	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Adamson	Jilly Adamson	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Akbar	Shakeel Akbar	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Akhtar	Mohammed Akhtar	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
alexander	john alexander	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Almeida	Antonio Almeida	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Anderson	Lisa Anderson	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
andy	Andy Silverman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Aronchik	Albert Aronchik	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Avila	Felicia Avila	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Azib	Hedi Azib	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bandhakavi	Ravi Bandhakavi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Barnes	Alex Barnes	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
bhatia49	harpreet bhatia49	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Biskupski	Mirek Biskupski	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Boettcher	Thomas Boettcher	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bohre	Henrik Bohre	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
BoireLavigne	Sebastien BoireLavigne	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
bosco	emerson bosco	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 13 - Administrator's Change Suite Access Permissions screen

2.2.3 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 configuration entries also need to be set in 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

2.2.4 System Configuration

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

2.2.4.1 Should ApTest Manager be closed to non-administrative users?

Answering yes will display 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 will be able to login.

This message is displayed if this configuration item is enabled and when the product is first installed.

To open ApTest Manager to Normal users:

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

2.2.4.2 What message should be displayed on the login page?

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

2.2.4.3 What is the URI of a bug tracking system you would like to use in conjunction with this copy of ApTest Manager?

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

When this feature is configured, a link to the problem tracking system appears on each ApTest Manager screen used to run a Test Case with which users can invoke the problem tracking system to submit bugs from within ApTest Manager. Depending on what the problem tracking system allows, ApTest Manager will automatically populate the information for creating a new problem report.

The problem tracking system URI can point directly to the problem tracking system if the system is web based. For problem tracking application programs (and for web based ones as well) the URI can be configured to point to a script that will invoke the command(s) for submitting a bug report. Consult with the administrator of the problem tracking system to determine the best way to invoke the problem tracking system.

The URI that can be configured can include macros that pass Test Case specific arguments to the problem tracking system if 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
- %RESULT - the result of running the Test Case
- %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 a problem tracking system:

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

ApTest Manger includes examples, in the directory `examples`, of how a problem tracking system integration can be provided. Examples are provided for several Open Source and commercial problem tracking systems such as Bugzilla, JIRA, and Test Track Pro.

These files are Perl scripts that are invoked from ApTest Manager with a URI that passes it all the information above. It then uses 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 bug tracking application and populates its new bug submission form with a complete set of information.

2.2.4.4 Should users be automatically logged out if they close their browser?

If No is selected users will only be logged out from ApTest Manager if they click Logout. Selecting Yes 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).

2.2.4.5 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.
- View Tests and Reports – the user can view reports and Test Cases but cannot edit or run tests.
- Run Assigned - the user can view reports and execute tests assigned to them but cannot run other tests, edit tests or modify Test Sets or Sessions.
- Run Andy Test – the user can view reports and execute tests but cannot edit tests or modify Test Sets or Sessions.
- Execute – the user can view reports, execute tests, and modify Test Sets and Sessions but cannot edit tests.
- Execute and Edit – the user can view reports, execute tests, modify Test Sets and Sessions, and edit Test Cases.
- Suite Manager – the user can view reports, 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 at any time after the Suite is created.

2.2.4.6 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.

2.2.4.7 What is the default time zone for your users?

If a default is not specified this value is set to the timezone of the server or, in the case of Windows servers, to UTC as it is not possible to determine what timezone Windows servers are in.

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

2.2.4.8 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 this option. By default execution fields other than Run Data are currently empty when run screens are initially displayed. Pre-population of Run Data Fields is controlled by the reset on rerun flag (see Section 1.9.5).

2.2.4.9 Normal User Capabilities

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

- Create test suites?
- Delete test sets/sessions?

ADMINISTRATION

- Delete test cases and folders?
- Empty the trash?
- Import test case from CSV files?
- Lock/Unlock test cases and folders?
- Change test suite configurations?
- Rename test cases and folders?
- Perform global replace operations?
- Change other's private report settings?
- Lock/Unlock test sets/sessions?
- Create their own accounts?
- Delete their own accounts?
- Edit their account information?
- Change their email notification settings?
- Suspend their email notifications?

2.2.4.10 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.

2.2.4.11 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.

2.2.5 Profile Catalog Management

The Manage Test Suite Profile Catalog screen allows administration of the catalog of Profiles used when creating a new Test Suite.

There are 5 Profiles provided with ApTest Manager. These Profiles may not be modified.

- Standard – a full IEEE 829 Test Case definition with time tracking enabled. The Test Case definition for this Profile is shown in Chapter 6.
- Standard without Time Tracking – same as the Standard Profile but without tracking of planned and actual test execution time.
- Standard With Table – a variant of the Standard Profile with a table of Procedure and Verification Fields for each step in tests. This can be useful for complex step-by-step test specifications.
- Tiny – a more simple Test Case definition than the Standard Profile, intended for more streamlined testing projects.
- TinyNoTime – same as the Tiny Profile but without tracking of planned and actual test execution time.

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.

Editing a Profile follows the same procedure defined in Chapter 6 for Changing Test Suite Configuration. Changes made to a Profile in the catalog will apply to new Test Suites created in the future but not to existing Test Suites previously created using that Profile.

2.2.6 Viewing 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, and thus occupy a license, until they log out (note that closing the browser does not necessarily cause a log out (see section 2.2.4.4)). However:

- If a user has been inactive for approximately 10 minutes, their license becomes available to be assigned to other users. If the user becomes active again after 10 minutes they will transparently reacquire a license if one is available. If no licenses are available however the user will be told to wait for one to become available before proceeding.
- A user will also be logged out if they have been inactive for one or more days. This limit can be configured from the Manage System Configuration screen.

Users with an active license are shown in green. Users with inactive are shown in red.

2.2.7 Viewing 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.

2.2.8 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 ApTest Manager Problem Reports. See also Section 3.3.

2.2.9 Check for Updates to ApTest Manager

From time to time ApTest Manager will check 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 this copy of ApTest Manager. It does not include any

A D M I N I S T R A T I O N

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.

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 WWW 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 upon the Berkeley DB from [Sleepycat Software](#) "the most widely used open source developer database in the world with over 200 million deployments". The Perl libraries on top of this C-Language database provide for very fast access to data items. ApTest has built a large collection of objects atop these libraries that provide an abstract layer for accessing ApTest Manager specific information. These objects and their methods are very efficient, relying upon in memory and on disk caches to speed up access to the most used information.

3.2 Test Case Files

ApTest Manager maintains Test Case information in its database. This enables it to provide extraordinary performance. In addition to the ApTest Manager API, external access to Test Case information is facilitated by a set of files that mirror the content of the Test Case database and which can be modified to import, edit, or modify Test Cases. These modifications can be made manually, using a text editor, or with a program. These files are updated when a Test Case is written. They are not involved in reading 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 folder, each named `<folder name>.dir`. Within each folder there is a file for each Test Case, named `<test case name>.bts`. Each directory also contains a file `DESCRIPTION` containing the description of the folder. The `DESCRIPTION` file in the top level of the tree for a Test Suite contains the description of the Test Suite.

Each Test Case file contains a human readable representation of the Test Case. Information is included for each Field in the 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 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.

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

3.2.1 Updating Tests

Test Case files can be updated using various application tools. The script `ATM_ROOT/bin/updateFields` provides the ability to transform a Field value for a group of Test Cases using Perl regular expressions. See the script's internal documentation for details.

Once the files have been modified, select **Synchronize the test suite** from the Manage Test Suite screen to update the database from the information in the files.

3.2.2 Copying Tests between Suites

Test Cases can be copied between Test Suites by copying Test Case and Folder files. This should be done with care between Test Suites with different Test Case Field definitions. Data in Test Cases for Fields that are not defined for a Test Suite are discarded.

Once the files have been copied, select **Synchronize the test suite** from the Manage Test Suite screen for the Suite they were copied to update the database from the information in the files..

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: <a href="http://www.aptest.com/atm2/datasheet.html"

target="_blank">datasheet.html, WF6.jpg

FILEDESC: ApTest Manager Data Sheet, Screen shot

Figure 14 - Test Case File

3.2.3 Registering Changes

When 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 Test Case files are created or modified.

Selecting **Synchronize the test suite** from the **Manage Test Suite** screen in ApTest Manager causes any Test Case files that have been changed to be picked up.

The script `syncdb` in the `ATM_ROOT/bin` directory will also do this. 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 will print the number of Test Case files it encountered for that 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.

3.3 Exporting Suites

To package up all the Test Suites and data files from an ApTest Manager instance in an archive that is portable and can be dropped into another instance of ApTest Manager, use the script `ATM_ROOT/bin/exportAll` (see also Section 2.2.8). This script will produce a zip file that can be unpacked into the `suites` directory of an ApTest Manager instance. See the script's internal documentation for details.

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

```
% cd ATMDIR
```

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

This will produce a file `myarchive.zip`.

To unpack the archive, transfer `myarchive.zip` to another machine and:

```
other% cd ATMDIR
```

```
other% rm data/userData.dbm
```

```
other% unzip myarchive.zip
```

This step removes the userdata file that was created on the new machine when it was set up.

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

This step will do an integrity check on all the Test Suites etc. and ensure they are ready for use on the new machine.

Be certain that all users are logged out of ApTest Manager on the original system before starting the migration process to ensure there is no data alteration once the process is underway. See also Section 2.2.8.

3.4 Backing up ApTest Manager Files

The best way to backup ApTest Manager 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 backup that is system neutral (and which thus can be restored to a different server) see Exporting Suites, above.

3.5 User Extensions

3.5.1 Javascript Extensions

It is possible for a user to customize ApTest Manager through the use of templates for reports and for 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 in the templates. However, this will mean 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 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 will have 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 ATM 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".

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.

3.5.2 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 will be available for use in templates. Care should be exercised not to conflict with the styles in `atm.css`.

3.6 Scripts

The ApTest Manager bin directory contains several scripts that can be executed from the command line on the server to perform various functions.

NAME	FUNCTION
exportAll.pl	Make a portable package of all ApTest Manager data
exportSuite.pl	Make a portable package of a Test Suite
importResults.pl	Import session results from CSV file (e.g. from test automation)
importTests	Import tests from CSV file
readdb	Read a Test Suite's test database
readReportDB	Read a Test Suite's report database
readSession	Read and display a Test Session
readSet	Read and display a Test Set
readSuiteDb	Read a suite metadata database
readUserDb	Read the user database
syncdb	Synchronize a Test Suite's test database with the file system tree
updatefields	Update field contents against patterns

INDEX

Access Level	2-5	uuid	1-12
Execute	2-5, 2-10	IEEE 829	vii
Execute and Edit	2-5, 2-10	Licensing	2-13
No access	2-5, 2-10	Login message	2-8
Run Any Test	2-5, 2-10	Normal user	2-2
Run Assigned	2-5, 2-10	Problem Report	
Suite Manager	2-5, 2-10	configuration	2-9
View Reports	2-5, 2-10	Reports	
View Tests and Reports	2-5, 2-10	templated	1-25
Administrator	2-1	Run Data	
atm_owner field	1-16	field types	1-18
Backup	3-5	flags	1-21
Closed to non-administrative users	2-8	styles	1-19
Database	3-1	Session Variables	1-5
Default Access	2-10	types	1-38
Email notifications	2-3, 2-7	Syncdb script	3-4
ExportAll script	3-4	System Configuration	2-8
Field Flag		Template	1-23
dependson	1-15, 1-21, 1-40	check for errors	1-24
derivedfrom	1-41	data entry	1-27
derivedfunc	1-41	errors	1-24
hidden	1-22, 1-41	reports	1-25
mandatory	1-16, 1-22, 1-41	time tracking	1-33
readonly	1-16, 1-22, 1-41	Test Case	
reset on rerun	1-22	copy	3-2
selectable	1-16, 1-23, 1-41	field types	1-8
settable	1-23	files	3-1
Field Style		Test Suite	
asis	1-20	copy	1-1
code	1-14	delete	1-2
date and time	1-13, 1-19	description	1-1
date only	1-13, 1-19	export	3-4
formatted	1-14, 1-20	rename	1-1
html	1-14, 1-20	Textarea Field Style	
minutes24	1-13	wysiwyg	1-21
minutes8	1-12	Time Tracking	1-33
number	1-13	Timezone	
wysiwyg	1-14	configuration	2-4, 2-11
Field Type		UpdateFields	3-2
cdate	1-8	User	
date	1-18	Administrator	2-1
file	1-9	normal	2-2
id 1-9		User Extensions	
modification history table	1-10	CSS	3-6
muser	1-10, 1-12	javascript	3-5
user	1-12		