

ORACLE®

PEOPLESOFT

PeopleTools 8.54: Automated Configuration Management

November 2016

ORACLE®

PeopleTools 8.54: Automated Configuration Management

CDSKU

Copyright © 1988, 2016, Oracle and/or its affiliates. All rights reserved.

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Contents

Preface	vii
Understanding the PeopleSoft Online Help and PeopleBooks.....	vii
PeopleSoft Hosted Documentation.....	vii
Locally Installed Help.....	vii
Downloadable PeopleBook PDF Files.....	vii
Common Help Documentation.....	viii
Field and Control Definitions.....	viii
Typographical Conventions.....	viii
ISO Country and Currency Codes.....	ix
Region and Industry Identifiers.....	ix
Using and Managing the PeopleSoft Online Help.....	x
PeopleTools Related Links.....	x
Contact Us.....	x
Follow Us.....	x
Chapter 1: Getting Started with Automated Configuration Management	13
Understanding Automated Configuration Management.....	13
Common Terms in Automated Configuration Management.....	13
Security Permissions Required for Automated Configuration Management.....	14
Chapter 2: Working with Automated Configuration Using PIA	15
Understanding Automated Configuration Using PIA.....	15
Delivered Configuration Plug-ins.....	15
ConfigurePTF.....	16
PTIBActivateDomain.....	16
PTIBActivateQueues.....	16
PTIBConfigureDBNode.....	17
PTIBConfigureGatewayNodes.....	18
PTIBConfigureGatewayProperties.....	19
PTIBNodeRegistration.....	20
PTIBRenameNode.....	20
PTPerformanceMonitor.....	21
PTURLConfiguration.....	23
PTPPConfigurePortalCluster.....	23
PTUNConfigureRemoteFolders.....	24
PTProcessSchedulerReportNode.....	24
PTProcessSchedulerServerConfig.....	26
PTSFAAdministerRemoteSearch.....	28
PTSFAAdministerSearch.....	29
PTSFCConfigureSES.....	30
PTSFCConfigureSearch.....	30
Executing Configuration Plug-ins.....	31
Executing a Template.....	31
Monitoring a Configuration.....	33
Reviewing the Configuration Details.....	34
Scheduling An Automated Configuration in PIA.....	35
Scheduling a Configuration Process.....	35
Chapter 3: Working with Custom Configuration Plug-ins and Templates	37
Understanding Custom Configuration Plug-ins and Templates.....	37

Working with Custom Configuration Plug-ins.....	37
Creating a Configuration Plug-in.....	37
Registering a Configuration Plug-in.....	39
Working with Custom Templates.....	41
Creating a Template.....	41
Editing Configuration Plug-in Properties.....	43
Defining Template Variables.....	45
Importing a Template Using PIA.....	45
Adding a Template to a Permission List.....	46
Chapter 4: Working with Automated Configuration Using the Command Line.....	49
Understanding Automated Configuration Using the Command Line.....	49
Understanding the Template File.....	49
Creating a Template File Manually.....	50
Working with the Structure and General Settings of a Template File.....	50
Specifying General Settings.....	51
Specifying Template Settings.....	52
Enabling Configuration Plug-ins.....	53
Creating a Template File in PIA.....	53
Exporting a Template.....	53
Running the PSRUNACM Script.....	54
Running PSRUNACM in Windows.....	54
Running PSRUNACM in UNIX.....	56
Running the PTEM_CONFIG Application Engine Program.....	57
Understanding the PTEM_CONFIG Program Run.....	58
Ensuring Sufficient Security Permissions.....	58
Setting the Environment Variables for the Command Line Session.....	58
Running the PTEM_CONFIG Application Engine Program.....	59
Verifying Your Configuration Settings.....	60
Chapter 5: Working with Sensitive Data.....	61
Understanding Password Encryption in Template File.....	61
Chapter 6: Migrating Configuration Data.....	63
Migrating Configuration Data Using ADS.....	63
Tables for Configuration Data Migration.....	63
ADS Definition for Template.....	63
ADS Definition for Plug-in.....	63
Exporting Configuration Data.....	64
Appendix A: Sample Template File.....	65
Sample Template File.....	65

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft Applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

PeopleSoft Hosted Documentation

You access the PeopleSoft Online Help on Oracle's PeopleSoft Hosted Documentation website, which enables you to access the full help website and context-sensitive help directly from an Oracle hosted server. The hosted documentation is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support, because that documentation is now incorporated into the hosted website content. The Hosted Documentation website is available in English only.

Note: Only the most current release of hosted documentation is updated regularly. After a new release is posted, previous releases remain available but are no longer updated.

Locally Installed Help

If your organization has firewall restrictions that prevent you from using the Hosted Documentation website, you can install the PeopleSoft Online Help locally. If you install the help locally, you have more control over which documents users can access and you can include links to your organization's custom documentation on help pages.

In addition, if you locally install the PeopleSoft Online Help, you can use any search engine for full-text searching. Your installation documentation includes instructions about how to set up Oracle Secure Enterprise Search for full-text searching.

See *PeopleTools Installation* for your database platform, "Installing PeopleSoft Online Help." If you do not use Secure Enterprise Search, see the documentation for your chosen search engine.

Note: Before users can access the search engine on a locally installed help website, you must enable the Search portlet and link. Click the Help link on any page in the PeopleSoft Online Help for instructions.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The two main types of common help are:

- Application Fundamentals
- Using PeopleSoft Applications

Most product families provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product family. Whether you are implementing a single application, some combination of applications within the product family, or the entire product family, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

In addition, the *PeopleTools: Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft Applications.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

<i>Typographical Convention</i>	<i>Description</i>
Key+Key	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For Alt+W, hold down the Alt key while you press the W key.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe ().
[] (square brackets)	Indicate optional items in PeopleCode syntax.

Typographical Convention	Description
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.
=>	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region and Industry Identifiers

Information that applies only to a specific region or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in the PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in the PeopleSoft Online Help:

- USF (U.S. Federal)
- E&G (Education and Government)

Using and Managing the PeopleSoft Online Help

Click the Help link in the universal navigation header of any page in the PeopleSoft Online Help to see information on the following topics:

- What's new in the PeopleSoft Online Help.
- PeopleSoft Online Help accessibility.
- Accessing, navigating, and searching the PeopleSoft Online Help.
- Managing a locally installed PeopleSoft Online Help website.

PeopleTools Related Links

[Oracle's PeopleSoft PeopleTools 8.54 Documentation Home Page \(Doc ID 1664613.1\)](#)

"PeopleTools Product/Feature PeopleBook Index" (PeopleTools 8.54: Getting Started with PeopleTools)

[PeopleSoft Information Portal](#)

[My Oracle Support](#)

[PeopleSoft Training from Oracle University](#)

[PeopleSoft Video Feature Overviews on YouTube](#)

Contact Us

[Send us your suggestions](#) Please include release numbers for the PeopleTools and applications that you are using.

Follow Us



Get the latest PeopleSoft updates on [Facebook](#).



Follow PeopleSoft on [Twitter@PeopleSoft_Info](https://twitter.com/PeopleSoft_Info).

Chapter 1

Getting Started with Automated Configuration Management

Understanding Automated Configuration Management

With each new release, Oracle PeopleTools provides new technology to our infrastructure, which adds rich, new features to support the functional requirements of our applications. While the new technology enhances the user experience and capabilities of our applications, the new technology often brings additional steps to the environment configuration and implementation process. This can create challenges for system administrators, development teams, and testing teams who routinely set up numerous environments that need to be refreshed on a regular basis. Oracle PeopleTools and PeopleSoft application teams share that same challenge.

For this reason, Oracle PeopleTools provides an automated configuration management framework that enables you to store your environment configuration settings in a template or a template file. PeopleTools provides two methods of running the automated configuration program — using the Automated Configuration Manager in PIA or using the command line. Command line can be used to run a batch file or shell script depending on your environment platform or to use the `psae` command. After setting up the basic infrastructure of a PeopleSoft environment, including database, application server, Process Scheduler server, and PIA domain, you run the configuration program (either using PIA or command line). This configuration program reads your environment properties from the template or template file, and inserts the stored values into the database, saving you from updating the settings manually each and every time you create or refresh an environment.

The settings stored in the template or template file are those that you typically enter on a configuration page in PeopleTools and save to the database. For example, the value for your Integration Gateway URL, which you would normally add manually on the Gateways page, can be entered in your template or template file once, and then retrieved by the automated configuration management framework and inserted into the database each time you refresh that environment.

A template is a composite of configuration plug-ins in which the properties and its values are defined. The configuration program fires configuration plug-ins that take the properties specified in the plug-in and configures the feature associated with that plug-in. You enable configuration plug-ins by referencing them in your template or template file and specifying the settings for that configuration plug-in.

Common Terms in Automated Configuration Management

Common terms used in automated configuration management:

Term	Description
Plug-in	A plug-in is an application class that corresponds to a configuration. For example, to configure IB Gateway nodes, the application class(plug-in) PTIBConfigureGatewayNodes in the application package PTEM_CONFIG is used.
Template	Templates are used for organizing, editing property values, and running the configuration program through PIA and command line.
Property	A property is the parameter for setting the value of a configuration. For example, env.default_local_node is the property name used for setting the value of default local node while configuring Integration Broker. This property will be present in the template if configuring through PIA or in the template file if configuring through command line.
Template Variable	A template variable can be assigned a value, which can be in-turn used to assign values for the properties in cases where values are recurring in multiple places in a template or template file.

Security Permissions Required for Automated Configuration Management

Before you begin using the automated configuration management framework, ensure that your user ID has the required roles.

Ensure that the following roles are added to your user ID:

- ACM Administrator
- Portal Administrator
- Search Administrator
- Search Server
- Search Developer
- Integration Administrator

When you use or create a template, you must ensure that the template is added to the required permission list.

See [Adding a Template to a Permission List](#).

Chapter 2

Working with Automated Configuration Using PIA

Understanding Automated Configuration Using PIA

Automated configuration management framework allows you to automate product configurations either through PIA or through command line.

This section describes automated configuration management using the Automated Configuration Manager feature in PIA. Automated Configuration Manager allows you to configure an environment with the required product configurations without navigating to the product's actual configuration pages.

Important! PeopleSoft recommends that product configuration through PIA be used only to test a template. The actual product configurations in a system must be performed through the command line.

Automated Configuration Manager uses a template to run a configuration program. In a template, you can include all the product configurations called plug-ins that you require for an environment, and you can group the plug-ins based on the product. For example, plug-ins required for Integration Broker configuration can belong to one group, whereas the plug-ins required for SES configuration can be another group in the same template. This enables you to configure more than one product in a single configuration program run.

Additionally, templates can be imported from and exported to an environment, so you can reuse templates in different environments by editing the configuration plug-in properties and values as required for an environment.

Automated Configuration Manager allows you to:

- Create, edit, manage, import, and export templates.
- Define template variables.
- Register configuration plug-ins.
- Specify template processing modes.
- Monitor a configuration run.

Delivered Configuration Plug-ins

PeopleTools delivers the following plug-ins. The properties of each plug-in are described here.

ConfigurePTF

Application class - PTEM_CONFIG:ConfigurePTF

The ConfigurePTF plug-in grants PTF User role to the PTF user and allows to accept non SSL requests.

Property	Default Value	Description
env.ptf_user	@userid@	Enter the PTF user. The userid variable is a template variable and @userid@ resolves at run time to the provided value. See Defining Template Variables .
env.ptf_allow_untrusted	true	Flag to allow untrusted SSL certificates.
env.ptf_use_page_prompt	true	Select to use Page Prompt and Prompt OK steps during recording in place of menu navigation. The Use Page Prompt option is also available on the PTF Test Recorder toolbar. The option selected here is the default for all users in this environment. The option selected on the PTF Test Recorder toolbar overrides this selection for that recording session.
env.ptf_use_message_recognition	true	Select to create entries for the Message Recognition feature during recording automatically. The Use Message Recognition option is also available on the PTF Test Recorder toolbar. The option selected here is the default for all users in this environment. The option selected on the PTF Test Recorder toolbar overrides this selection for that recording session.

PTIBActivateDomain

Application class - PTEM_CONFIG:PTIBActivateDomain

The PTIPActivateDomain plug-in activates pub/sub domain.

Property	Default Value	Description
domain.activate_retry_count	10	Enter the number of retries.
domain.activate_wait_time	10	Enter the wait time for domain activation.

PTIBActivateQueues

Application class - PTEM_CONFIG:PTIBActivateQueues

The PTIBActivateQueues plug-ins activates the Integration Broker queue.

Property	Default Value	Description
queue.activate_queue_list	PS_ALL	Use PS_ALL to activate all queues. If you want to activate selected queues, enter queue names separated by commas.
queue.activate_queue_status	1	Valid values: <ul style="list-style-type: none"> • 1 - activate a queue • 0 - pause a queue Default value is 1.

PTIBConfigureDBNode

Application class - PTEM_CONFIG:PTIBConfigureDBNode

The PTIBConfigureDBNode plug-in configures the database node.

Property	Default Value	Description
env.pia_webserver_host	@host@.@domain@	Server host of the PIA domain.
env.pia_webserver_port	@httpport@	HTTP port on which the PIA domain listens.
env.pia_webserver_ssl_port	@sslport@	HTTPS port on which the PIA domain listens.
env.pia_site_name	PS	PeopleSoft site name.
env.gateway_host	@host@.@domain@	If using multiple machines, enter the gateway server host.
env.gateway_port	@httpport@	If using multiple machines, enter the gateway HTTP port.
env.gateway_ssl_port	@sslport@	If using multiple machines, enter the gateway HTTPS port.
env.use_ssl_gateway	false	This Boolean value specifies whether an SSL gateway is configured for the PeopleSoft system.
env.use_ssl_webserver	false	This Boolean value specifies whether SSL is used or not. It is based on this flag that the security mode is set. If the flag is not set, HTTP is used, else HTTPS is used for node URI.
env.default_user_id	@userid@	Default user ID for the environment.
env.default_local_node_pass		Default local node password.

Property	Default Value	Description
env.anonymous_default_user_id	@userid@	Specifies the default user ID for message node name ANONYMOUS.
env.default_local_node	@nodename@	Default local node
env.wsdl_external_user_id	@userid@	Specifies the external user ID for wsdl_node.
env.wsdl_external_pass		Specifies the external user password for wsdl_node.
env.configure_wsdl_node	false	Flag to configure WSDL node
env.wsdl_node_tokentype	NONE	Specifies the authentication token type for WSDL. Valid values: <ul style="list-style-type: none"> • NONE • STSD - SAML token • USRT - User name token
env.wsdl_node_tokenencrypted	0	Encryption for WSDL node
env.wsdl_node_tokensigned	0	Digital Signature for WSDL node
env.wsdl_node_ibencryptionlevel	A	Specifies the encryption level for WSDL node. Valid values: <ul style="list-style-type: none"> • A - All • B - Body • H - Header level

PTIBConfigureGatewayNodes

Application class - PTEM_CONFIG:PTIBConfigureGatewayNodes

The PTIBConfigureGatewayNodes plug-in configures gateway URL, load connectors, and defines node in the gateway.

Property	Default Value	Description
env.gateway_host	@host@.@domain@	Gateway host
env.gateway_port	@httpport@	Gateway port
env.gateway_ssl_port	@sslport@	Gateway SSL port

Property	Default Value	Description
env.use_ssl_gateway	false	Flag to determine whether the configuration is secure or non secure. Valid values: <ul style="list-style-type: none"> False - indicates non secure (http) configuration True - indicates secure (https) configuration
env.default_local_node	@nodename@	Default local node
env.gateway_user	administrator	Specify the gateway administrator user ID.
env.gateway_password		Specify the password for the gateway administrator user ID.
env.ib_appserver_host	@host@.@domain@	Server host of the application server domain.
env.ib_jolt_port	@joltport@	Jolt port.
env.ib_node_proxy_userid	@userid@	User ID for proxy, if used.
env.ib_node_proxy_password		Password for proxy, if used.
env.tools_release	@tools_release@	PeopleTools release level.
env.ib_appserver_domain_password		Password for the application server domain.
env.ib_virtual_node		When you want a PeopleSoft node to process an inbound request that was sent without a destination node, the integration system directs the request to the node specified for this property.
env.ib_set_as_default_node	true	This flag is set for the node which hosts the common gateway. The flag is true only for the cluster node which hosts the common gateway.
env.ib_synchronize_map_files	Y	Execute MAP File Synchronization

PTIBConfigureGatewayProperties

Application class - PTEM_CONFIG:PTIBConfigureGatewayProperties

The PTIBConfigureGatewayProperties plug-in sets keystore password, proxy host, proxy port, and non proxy host in gateway properties file.

Property	Default Value	Description
env.gateway_host	@host@.@domain@	Gateway host.
env.gateway_port	@httpport@	Gateway port.
env.gateway_ssl_port	@sslport@	Gateway SSL port.
env.use_ssl_gateway	false	Flag to determine whether the configuration is secure or non-secure. Valid values: <ul style="list-style-type: none"> • True - HTTPS • False - HTTP
env.gateway_user	administrator	Gateway user name.
env.gateway_password		Gateway user password.
env.gateway_keystore_password		Gateway keystore password.
env.gateway_proxy_host		Gateway proxy host.
env.gateway_proxy_port		Gateway proxy port.
env.gateway_non_proxy_hosts	@host@.@domain@ localhost *.us. oracle.com *.oracle.com *.oraclecorp. com *.peoplesoft.com	Gateway non proxy hosts.

PTIBNodeRegistration

Application class - PTEM_CONFIG:PTIBNodeRegistration

The PTIBNodeRegistration plug-in registers new nodes to the Integration Broker network.

Property	Default Value	Description
integration_network.NODEGROUP1	NODE1,NODE2,NODE3,NODE4	Comma-separated list of nodes to register.

PTIBRenameNode

Application class - PTEM_CONFIG:PTIBRenameNode

The PTIBRenameNode plug-in renames the default local node.

Property	Default Value	Description
env.default_local_node	@nodename@	Default local node.
env.app_msg_purge_all_dms	true	Purge application server messages.

PTPerformanceMonitor

Application class - PTEM_CONFIG:PTPerformanceMonitor

Before you run the PTPerformanceMonitor plug-in, ensure that the following pre-requisites are set in the monitoring system and in the monitored system:

Note: The PTPerformanceMonitor plug-in must be run in the monitoring and in the monitored systems.

- Perf Collator (option No 10 in psadmin) should be set to Yes for the corresponding application server domain.
- In the psappsrv.cfg file and the psprcs.cfg file, the EnablePPM Agent parameter should be set to 1.
- Process scheduler domain must be up and running so that the reaper and archive jobs can be scheduled successfully.
- In the web profile configuration, set the following:
 - In the General tab, select the EnablePPM Agent parameter.
 - In the Custom Properties tab, set the value of PPMConsole parameter to True.

If the PPMConsole parameter is not present, add the parameter. The parameter must be of Boolean type.

Additionally, to enable Usage Monitor as part of the performance monitor configurations, the following pre-requisites are required:

- In the psappsrv.cfg file, the Usage Monitoring State parameter should be set to 1 or 2.
- On the System Defaults page (PeopleTools, Performance Monitor, Administration, System Defaults), the Enable Usage Monitoring check box should be selected. Alternatively, in the PTPerformanceMonitor plug-in properties, you can set the env.usemon_sw property to 1.

Important! After executing the PTPerformanceMonitor plug-in, clear the cache and restart the application server and Web server for the configurations to take effect.

Property	Default Value	Description
env.pm_sampling_rate	0	Agent PMU sample rate (1/X).
env.pm_buff_int	0	Agent buffering interval (in seconds).
env.pm_sample_int	0	Agent event sample rate (in seconds).
env.pm_ping_int	0	Agent heartbeat interval (in seconds).
env.pm_user_trace		Allow performance trace.
env.pm_max_buff	0	Agent maximum buffer size (in bytes).
env.pm_max_hist_age	0	Retention period (in days).

Property	Default Value	Description
env.usemon_sw	0	Enable usage monitoring.
env.pm_max_trans_tmout	0	PMU timeout (in days).
env.pm_archive_mode		Valid values for archiving: <ul style="list-style-type: none"> • 0 - Archive nothing • 1 - Archive data • 2 - Delete data • 3 - Delete system
env.pm_filter_level2		Filter level values for log documentation: <ul style="list-style-type: none"> • 01 - Standby • 02 - Error • 03 - Warning • 04 - Standard • 05 - Verbose • 06 - Debug
env.monitor_url		Enter an URL in the following format or enter NONE. Format: http(s)://host[:port]/monitor/[site]/ If you enter NONE, the performance monitor components on the monitored system are not enabled.
env.ppmi_user_name		PPMI user ID.
env.ppmi_password		PPMI password.
env.collator_row_limit		Collator row limit. Use 0 for unlimited rows.
env.ppmi_url		PPMI URL. Format: http://host[:port]/monitor/[site]/
env.schedule_reaper_job	Y	Flag to determine whether the Reaper job needs to be scheduled.
env.schedule_lookup_job	Y	Flag to determine whether the Lookup job needs to be scheduled.

Property	Default Value	Description
env.schedule_archive_job	Y	Flag to determine whether the Archive job needs to be scheduled.
env.schedule_umlookup_job	Y	Flag to whether the Usage Monitor Lookup job needs to be scheduled.
env.cluster_list		List of performance monitor clusters.
env.insert_cluster_urls	N	Flag to determine whether performance monitor clusters need to be inserted or not.
env.delete_cluster_urls	N	Flag to determine whether performance monitor clusters need to be deleted or not.

PTURLConfiguration

Application class - PTEM_CONFIG:PTURLConfiguration

The PTURLConfiguration plug-in configures URL configurations.

Property	Default Value	Description
env.url_config_url_id	TEST	URL ID.
env.url_config_url_descr	DESCR	URL description.
env.url_config_url	http://	URL name.
env.url_config_comments	Comments	Comments
env.url_config_property_name		Property name.
env.url_config_property_value		Property value.

PTPPConfigurePortalCluster

Application class - PTEM_CONFIG:PTPPConfigurePortalCluster

The PTPPConfigurePortalCluster plug-in configures portal host nodes of the portal cluster.

Property	Default Value	Description
env.portal_cluster_nodes	PA,HRMS,ERP,EPM,CRM	Comma separated values determine the names of nodes that needs to be part of the portal cluster.

PTUNConfigureRemoteFolders

Application class - PTEM_CONFIG:PTUNConfigureRemoteFolders

The PTUNConfigureRemoteFolders plug-ins configures unified navigation remote folders.

Property	Default Value	Description
env.remote_folders	HRMS	Comma separated values specify remote node names that need to be accessed for Unified Navigation.
env.remote_folders	NODE1	Comma separated values specify remote node names that need to be accessed for Unified Navigation.
env.portal_cluster_nodes		Comma separated values determine the names of nodes that need to be part of the portal cluster.
NODE1.label		Label for the folder, which will appear in the portal system's drop-down menu.
NODE1.portal_name	EMPLOYEE	Portal name.
NODE1.remote_node_name	NODE1	Remote node name.
NODE1.remote_folder_name	PORTAL_ROOT_OBJECT	Remote folder name.
NODE1.local_parent_folder_name	PORTAL_ROOT_OBJECT	Parent of the remote folder.

PTProcessSchedulerReportNode

Application class - PTEM_CONFIG:PTProcessSchedulerReportNode

The PTProcessSchedulerReportNode plug-in configures Process Scheduler report node configuration.

Property	Default Value	Description
env.pt_prop_name		Property name.
env.wrkpassword		Node password.
env.wrknfpassword		Confirm password.
env.wrkoperpswd		Operator password (encrypted).
env.cdm_ssl_mode		Valid values for SSL mode: <ul style="list-style-type: none"> • 0 - EXPLICIT • 1 - IMPLICIT
env.url		URL ID.

Property	Default Value	Description
env.uri_host		URI host.
env.pt_fileencpasswd		Encrypted password.
env.distnodename		Distribution node name.
env.ftpaddress		File transfer protocol address.
env.winnetworkpath		Windows network path.
env.cdm_trprotocol		Transfer protocol.
env.descrlong		Description
env.uri_port	0	URI port.
env.uri_resource		URI resource.
env.operpswd		Operator password (encrypted).
env.cdm_proto		Valid values for protocol: <ul style="list-style-type: none"> • 0 - HTTP • 1 - HTTPS • 2 - XCOPY • 3 - FTP • 4 - FTPS • 5 - SFTP
env.opsys		Valid values for operating system: <ul style="list-style-type: none"> • 0 - DOS • 1 - NT/Win95 Client • 2 - Windows • 3 - OS/2 • 4 - UNIX • 5 - VMS • 6 - MPE/XL • 7 - OS390 • 9- OS/400
env.ftpid		File transfer ID.
env.ftpdirectory		Directory for FTP.

Property	Default Value	Description
env.pt_prop_value		Property value.

PTProcessSchedulerServerConfig

Application class - PTEM_CONFIG:PTProcessSchedulerServerConfig

The PTProcessSchedulerServerConfig plug-in configures Process Scheduler server configuration.

Property	Default Value	Description
env.daemonprcsinst	0	Process instance.
env.maxconcurrent	0	Maximum concurrent processes.
env.notifywhensusp	0	Notify when suspended.
env.transfermaxretry	0	Maximum transfer retries.
env.srvloadbaloptn		Server load balancing option. Valid values: <ul style="list-style-type: none"> • 0 - Do Not Use for Load Balancing • 1 - Use for Load Balancing
env.redistwrkoption		Redistribute workload option. Valid values: <ul style="list-style-type: none"> • 0 - Do not redistribute • 1 - Redistribute with same O/S • 2 - Redistribute to any O/S
env.minmem	0	Minimum memory required.
env.distid		Distribution ID.
env.servername		Server name.
env.prcsnotifyfreq	0	Notify frequency.
env.start_hours		Start hours.
env.notifywhenstartd	0	Notify when started.
env.maxapiaware	0	Maximum API aware tasks.
env.daemonenabled		Daemon enabled.
env.end_hours		End hours.

Property	Default Value	Description
env.end_minutes		End minutes.
env.notifydisabled		Notify disabled.
env.prcspriority		Process priority. Valid values: <ul style="list-style-type: none"> • 1 - Low • 5 - Medium • 9 - High
env.maxcpu	0	Maximum CPU usage required.
env.prcstype		Process type.
env.day_ofweek		Day Valid values: <ul style="list-style-type: none"> • 0 - Sunday • 1 - Monday • 2 - Tuesday • 3 - Wednesday • 4 - Thursday • 5 - Friday • 6 - Saturday
env.distnodename		Distribution node name.
env.prcscategory		Process category.
env.notifyservererror	0	Server errors.
env.notifywhendown	0	Notify when down.
env.sleeptime	0	Sleep time.
env.heartbeat	0	Heartbeat
env.transferlogfiles		Transfer log files to content.
env.transferinterval	0	Interval for transfer attempt.
env.daemongroup		Daemon procedure group.
env.start_minutes		Time

Property	Default Value	Description
env.distidtype		Distribution ID type. Valid values: <ul style="list-style-type: none"> • 2 - User ID • 3 - Role name
env.daemonsleeptime	0	Daemon sleep time.
env.daemoncyclecnt	0	Daemon recycle count.
env.descr		Description
env.opsys		Operating system. Valid values: <ul style="list-style-type: none"> • 0 - DOS • 1 - NT/Win95 Client • 2 - Windows • 3 - OS/2 • 4 - UNIX • 5 - VMS • 6 - MPE/XL • 7 - OS390 • 9- OS/400
env.processtypepriority	1	Process type priority. Valid values: <ul style="list-style-type: none"> • 1 - Low • 5 - Medium • 9 - High
env.processtypemaxconcurrent	3	Process type maximum concurrent value.

PTSFAdministerRemoteSearch

Application class - PTEM_CONFIG:PTSFAdministerRemoteSearch

The PTSFAdministerRemoteSearch plug-in configures remote search groups as specified in search network.

Property	Default Value	Description
env.search_network.operation	REGISTER	Valid values: <ul style="list-style-type: none"> REGISTER UNREGISTER
env.search_network.dest_node_names	NODE1	Comma-separated list of nodes from which search groups need to be imported from.
env.search_network.NODE1		Comma-separated list of nodes from which search groups need to be imported to the NODE1 node.

PTSFAdministerSearch

Application class - PTEM_CONFIG:PTSFAdministerSearch

The PTSFAdministerSearch plug-in deploys, undeploys, and schedules index generation for search definitions/categories from a selected list.

Property	Default Value	Description
env.ptsf_selection_type	Global	Valid values: <ul style="list-style-type: none"> All Global List
env.ptsf_include_definitions	True	Comma separated list of search category names to be included. You can use % as an operator.
env.ptsf_exclude_definitions	True	Comma separated list of search category names to be excluded. You can use % as an operator.
env.ptsf_check_audit_errors	True	If true check for access to query/connected query, or invalid objects and stop if errors found.
env.ptsf_admin_operations	DEPLOY, INDEX	Comma-separated list of administration operations. Valid values: <ul style="list-style-type: none"> Deploy Index Undeploy Deploy,Index <p>You can use any combination of comma-separated values.</p>

Property	Default Value	Description
env.ptsf_index_all_languages	False	If true the schedules are created to index all languages.
env.report_schedule_status_after_minutes		Maximum minutes to wait before reporting scheduling status. Enter 0 to wait till finish; leave blank to skip report.
env.ptsf_schedule_on_server		Specify the Process Scheduler to run the indexing on. Leave this blank to use master scheduler.

PTSFConfigureSES

Application class - PTEM_CONFIG:PTSFConfigureSES

The PTSFConfigureSES plug-in configures SES identity plug in and defines proxy user on SES.

Property	Default Value	Description
env.ses_host	@seshost@	SES host name.
env.ses_port	@sesport@	SES port.
env.ses_use_ssl	False	SSL option.
env.ses_admin_user		SES administrator user name.
env.ses_admin_password		SES administrator user password.
env.ses_proxy_user		SES proxy user name.
env.ses_proxy_password		SES proxy user password.

PTSFConfigureSearch

Application class - PTEM_CONFIG:PTSFConfigureSearch

The PTSFConfigureSearch plug-in configures PeopleSoft system to talk to SES and assigns roles.

Property	Default Value	Description
env.ses_host	@seshost@	SES host name.
env.ses_port	@sesport@	SES port.
env.ses_use_ssl	False	SSL option.
env.ses_admin_user		SES administrator user name.
env.ses_admin_password		SES administrator user password.

Property	Default Value	Description
env.ses_proxy_user		SES proxy user name.
env.ses_proxy_password		SES proxy user password.
env.ses_call_back_user	@userid@	SES call back user name.
env.ses_call_back_password		SES call back user password.
env.search_administrator_user	@userid@	Search administrator user name.
env.search_developer_user	@userid@	Search developer user name.
env.gateway_host	@host@.@domain@	Gateway host.
env.gateway_port	@httpport@	Gateway port.
env.gateway_ssl_port	@sslport@	Gateway SSL port.
env.use_ssl_gateway	False	Flag to determine whether the configuration is secure or non secure. Valid values: <ul style="list-style-type: none"> • False - indicates non secure (http) configuration • True - indicates secure (https) configuration
env.default_local_node	@nodename@	Default local node.
env.enable_global_menu_search	All	Enables global search and menu search for All, Local or comma separated list of portals.

Executing Configuration Plug-ins

This section discusses:

- Executing a template.
- Monitoring a configuration.
- Reviewing configuration details.

Executing a Template

Use the Configuration Template Definition page to add or delete plug-ins, check the plug-in dependency, execute a configuration, and monitor a configuration.

PeopleTools provides a set of templates to get you started configuring elements off your system. For example, the IB_TEMPLATE is used for configuring Integration Broker, the SES_TEMPLATE is used for configuring Oracle SES and the PeopleSoft Search Framework, and so on. You can view the set of delivered templates by accessing the Configuration Template Definition page.

Note: The template you can use for configuring mobile approvals is APPROVALS_TEMPLATE. It will not appear by default in PIA. It is provided as a text file in PS_HOME\utility.

Access the Configuration Template Definition page using the following navigation path:

PeopleTools, Automated Config Manager, ACM Template, Template Definition

Image: Configuration Template Definition page

This example illustrates the fields and controls on the Configuration Template Definition page. You can find definitions for the fields and controls later on this page.

Configuration Template Definition
Define template to configure environment

Template Name IB_TEMPLATE

*Description Integration Broker Template

Long Description Template to configure Integration Broker

Configuration Plugins Find First 1 of 1 Last

*Group Integration Broker

Description Integration Broker Configurations

Plugin	Dependency Check	Properties
<input type="checkbox"/> IB Rename Node		
<input checked="" type="checkbox"/> IB Node Configuration		
<input checked="" type="checkbox"/> IB Gateway Node Configuration		
<input checked="" type="checkbox"/> IB Gateway Properties Configuration		
<input type="checkbox"/> IB Pub/Sub Domain Activation		

[Edit Template Variables](#) [Configuration Monitor](#) [Process Monitor](#)

Save Execute

Template Name

The name of the template that is entered while creating a template.

The template name cannot be edited.

Group

A group enables you to create a grouping of plug-ins related to a configuration. For example, you can group plug-ins related to Integration Broker configuration.

You may add multiple groups in a template to configure different products. For example, you can add a group of plug-ins that configure the SES product.

Plugin

Use the check box to select a plug-in that you want to include in the configuration program run.



Dependency Check

Click the Dependency Check icon to review whether any dependencies exist for the current plug-in. If any dependency exists, you must execute the plug-in before you run the current configuration plug-in.



Properties

Click the Properties icon to review the values that are defined for each property and edit if needed. If you do not edit any of the properties, the default property values will be used in the configuration program run.

See

Edit Template Variables

Click the Edit Template Variables link to review the property values that will be used at run time. The template variables are common to all the groups within a template.

Configuration Monitor

After clicking the Execute button to commence a configuration program run, click the Configuration Monitor link to review the configuration program status.

See [Monitoring a Configuration](#).

Process Monitor

Click the Process Monitor link to review the process status.

The Process Monitor link is active only if the processing mode is Scheduled.

See [Scheduling An Automated Configuration in PIA](#).

Save

If you modify a template, save the template before commencing a configuration program run.

Execute

Click the Execute button to commence a configuration program run.

Monitoring a Configuration

Use the Configuration Monitor page to review the status of the configuration template and to review the details.

Access the Configuration Monitor page using the following navigation path:

PeopleTools, Automated Config Manager, Configuration Monitor

Image: Configuration Monitor page

This example illustrates the fields and controls on the Configuration Monitor page. You can find definitions for the fields and controls later on this page.

The screenshot shows the Configuration Monitor page. At the top, there is a 'Filter' section with the following fields: 'User ID' (text input with a search icon), 'Run Status' (dropdown menu), 'From Date' (calendar icon), 'Template Name' (text input with a search icon), and 'To Date' (calendar icon). A yellow 'Filter' button is located to the right of these fields. Below the filter section is a 'Configuration Details' table with a toolbar containing 'Personalize', 'Find', 'View All', and a grid icon. The table has the following data:

Instance	Template Name	User ID	Database	Run Date/Time	Status	Details
1	IB_TEMPLATE	PTACMADMIN	QEDMO	05/14/2014 2:00:30AM PDT	Error	

- User ID** Select a user ID by which you want to filter the search results.
- Run Status** Select either Error or Success to filter the search results.
- Template Name** Select a template name by which you want to filter the search results.
- From Date** Enter a date range by which you want to filter the search results.
- To Date** Enter a date range by which you want to filter the search results.
- Filter** Click the Filter button to show results based on the filter criteria.
- Details** Click the Details icon to view the configuration details.

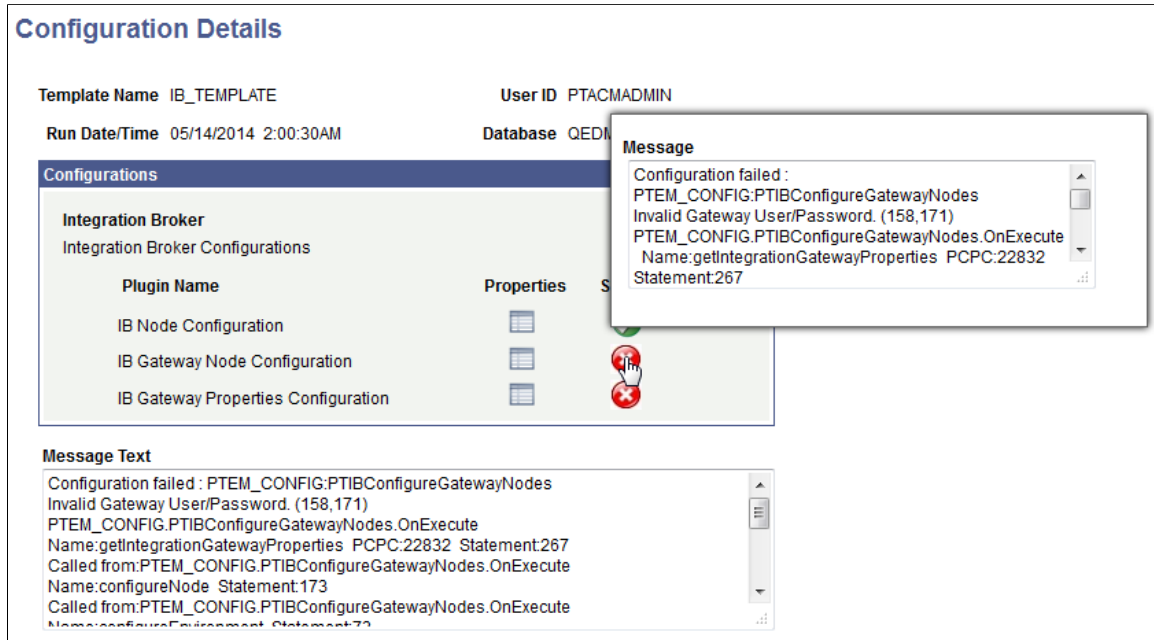
Reviewing the Configuration Details

Use the Configuration Details page to review the configuration details after running a template and to review the errors that caused a configuration to fail.

To view the Configuration Details page, click the Details icon on the Configuration Monitor page.

Image: Configuration Details page

This example illustrates the fields and controls on the Configuration Details page. You can find definitions for the fields and controls later on this page.



 **Success Status icon**

Indicates a successful configuration.

 **Properties icon**

Click the Properties icon to display a read-only listing of property values for the selected plug-in.

 **Error Status icon**

Hover on the error status icon to view details of the failed configuration.



Indicates that the verification of configuration failed.

Scheduling An Automated Configuration in PIA

Automated configuration management framework provides two processing modes for a configuration process. The two processing modes are:

- Interactive Mode
- Scheduled Mode

Scheduling a Configuration Process

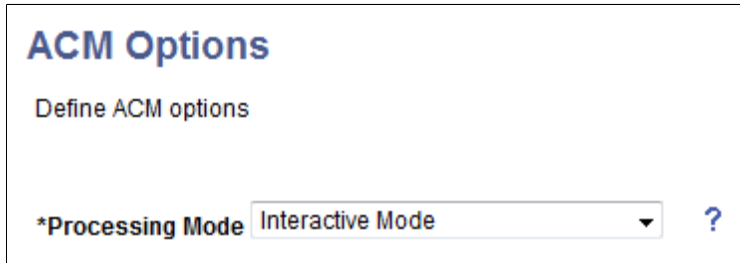
Use the ACM Options page to choose a processing mode for a configuration process.

Access the ACM Options page using the following navigation path:

PeopleTools, Automated Config Manager, ACM Utilities, ACM Options

Image: ACM Options page

This example illustrates the fields and controls on the ACM Options page.



Interactive Mode

In interactive mode, the configuration process commences when the Execute button is clicked on the Configuration Template Definition page and does not allow a user to navigate out of the page.

Scheduled Mode

In scheduled mode, when the Execute button is clicked, the configuration process is scheduled in Process Scheduler and while it is being processed a user can navigate to other pages. If you select scheduled mode, you should ensure that the Process Scheduler domain is running.

Chapter 3

Working with Custom Configuration Plug-ins and Templates

Understanding Custom Configuration Plug-ins and Templates

In addition to the plug-ins and templates delivered by PeopleTools, PeopleTools allows you to create custom configuration plug-ins required for your specific environment and create templates.

A plug-in is an application class that corresponds to a configuration. For example, to configure IB Gateway nodes, the application class, PTIBConfigureGatewayNodes, in the application package PTEM_CONFIG is used. Similar plug-ins can be grouped together, which is called a group. A template is used to organize a group of plug-ins or multiple groups of plug-ins. A template also allows you to edit property values of plug-ins, and run the configuration program through PIA or command line.

Working with Custom Configuration Plug-ins

In addition to the plug-ins delivered by PeopleSoft, PeopleSoft allows you to create custom configuration plug-ins required for your specific environment.

A plug-in is an application class that corresponds to each configuration within a template. For example, to configure IB Gateway nodes, the application class (plug-in) PTIBConfigureGatewayNodes in the application package PTEM_CONFIG

Working with configuration plug-ins includes the following:

- Creating a configuration plug-in.
- Registering a configuration plug-in.
- Adding a configuration plug-in to a template.

See [Creating a Template](#).

- Migrating a plug-in using ADS definition.

See [Migrating Configuration Data Using ADS](#).

Creating a Configuration Plug-in

To create a configuration plug-in:

1. Use the PTEM_CONFIG application package to create a plug-in. If you need to use a custom package, ensure that the custom package starts with PTEM_CONFIG.

2. Write the plug-in class. The new plug-in class must extend from the base class — PTEM_CONFIG:EMConfigurationPlugin.

Override and implement the four base class methods:

- `getProperties` - to provide plug-in properties and their details.

For example:

```
method getProperties
  /* Returns Array of PTEM_CONFIG:PTEMVariableProperty */
  /* Extends/implements PTEM_CONFIG:EMConfigurationPlugin.getProperties =>
  */

  Local array of PTEM_CONFIG:PTEMVariableProperty &propArray = Null;

  ---add the properties as shown below-----
  &variableProperty = create PTEM_CONFIG:PTEMVariableProperty("property=>
  _name", "datatype", True/False (is required?), True/False(is password?), =>
  "default value", Message set number of property description, message numb=>
  er, "default message", Null); &propArray.Push(&variableProperty); ----=>
  ----->
  -----
  Return &propArray;
end-method;
```

Add the following base class methods:

- `validateVariables` - to validate the plug-in property values.
- `configureEnvironment` - to configure the environment with the provided values.
- `validateConfigurations` - to validate whether the values are correctly configured.
- `dependant_plugins` - to provide the names of dependent plug-ins.

For example:

```
method dependant_plugins
  /* Returns Array of String */
  /* Extends/implements PTEM_CONFIG:EMConfigurationPlugin.dependant_plug=>
  ins */
  Local array of string &dependant_array;
  &dependant_array = CreateArray(""); -----add the plugins as comma sepr=>
  ated values, that this plugin depends on for its execution
  Return &dependant_array;
end-method;
```

- `getPluginHelpMessage` - to provide a brief description of a plug-in.

For example:

```
method getPluginHelpMessage
  /* Returns PTEM_CONFIG:PTEMHelpMessage */
  /* Extends/implements PTEM_CONFIG:EMConfigurationPlugin.getPluginHelpM=>
  essage */
  Local PTEM_CONFIG:PTEMHelpMessage &tempMessage = Null;
  &tempMessage = create PTEM_CONFIG:PTEMHelpMessage(0, 0, "", Null); ->
  --- "add the message set number, message number and default message of th=>
  e help message"
  Return &tempMessage;
end-method;
```

- `isInternalPlugin` - to differentiate between plug-ins that can be used internally by developers and externally by customers.

Valid values are:

- `False` - specifies that the plug-in can be used by customers, and the plug-in can be added in a template in PIA.
- `True` - implies that the plug-in can be used only through command line and will not be available in PIA.

For example:

```
method isInternalPlugin
  /+ Returns Boolean +/
  /+ Extends/implements PTEM_CONFIG:EMConfigurationPlugin.isInternalPlug=>
in +/
  Return False;
end-method;
```

Note: When you create a configuration plug-in an application package, the plug-in must be registered through the automated configuration management framework so that it can be included in a template for configuration through PIA. If a plug-in is not registered, it can be used only for execution through command line.

Registering a Configuration Plug-in

Use the Register Plugin page to register a configuration plug-in, which includes assigning a category for the plug-in.

Access the Register Plugin page using the following navigation path:

PeopleTools, Automated Config Manager, ACM Utilities, Register Plugin

Image: Register Plugin page

This example illustrates the fields and controls on the Register Plugin page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web form titled "Register Plugin" with the subtitle "Add plugin to plugin repository and assign category". The form contains four input fields:

- *Plugin Name**: A text input field with a magnifying glass icon to its right.
- *Description**: A text input field.
- Long Description**: A larger text area with a magnifying glass icon to its right.
- *Plugin Category**: A text input field with a magnifying glass icon to its right.

Plugin Name

Select the plug-in that you created.

Note: If the `isInternalPlugin` method of a plug-in class is defined as false, the plug-in will be available in the Plugin Name list.

Description

Enter a description for the plug-in.

Long Description

Optionally, you may enter a long description for the plug-in.

Plugin Category

Select a category to which the plug-in will belong.

Currently, the following categories are available:

- IB
- Miscellaneous
- Network Setup
- Process Scheduler
- SES

See [Creating a Configuration Plug-in](#).

Working with Custom Templates

Automated configuration management uses templates to organize configuration plug-ins, edit the configuration property values, and run the configuration process in PIA and command line.

Working with custom templates includes:

- Creating a template.
- Editing configuration plug-in properties.
- Defining template variables.
- Importing a template using PIA.
- Adding a template to a permission list.
- Migrating template data.

See [Migrating Configuration Data Using ADS](#).

Creating a Template

Use the Template Definition page to create a template. Use the Configuration Template Definition page to add configuration plug-ins, edit properties, and define template variables.

Access the Configuration Template Definition page using the following navigation path:

PeopleTools, Automated Config Manager, ACM Templates, Template Definition

Image: Configuration Template Definition page

This example illustrates the fields and controls on the Configuration Template Definition page. You can find definitions for the fields and controls later on this page.

Configuration Template Definition
Define template to configure environment

Template Name IB_TEMPLATE

*Description Integration Broker Template

Long Description Template to configure Integration Broker

Configuration Plugins Find First 1 of 1 Last

*Group Integration Broker + -

Description Integration Broker Configurations

Plugin	Dependency Check	Properties		
<input type="checkbox"/> IB Rename Node				
<input checked="" type="checkbox"/> IB Node Configuration				
<input checked="" type="checkbox"/> IB Gateway Node Configuration				
<input checked="" type="checkbox"/> IB Gateway Properties Configuration				
<input type="checkbox"/> IB Pub/Sub Domain Activation				

[Edit Template Variables](#) [Configuration Monitor](#) [Process Monitor](#)

Save Execute

Template Name

The name of the template that is entered while creating a template.

The template name cannot be edited.

Description

Enter a short description for the template.

Group

A group enables you to create a grouping of plug-ins related to a configuration. For example, you can group plug-ins related to Integration Broker configuration.

In a template, you can create multiple groups to configure different products. For example, in addition to the Integration Broker group, you can add a group of plug-ins that configure the SES product.

Plugin

Choose a plug-in that you want to include as part of the group.

In a template, you can add a plug-in multiple times. For example, you may need to create multiple report nodes in Process Scheduler. In this scenario, you can add the report node

plug-in multiple times with values as needed so that multiple report nodes will be created when you run the configuration program.

Select a check box to include a plug-in in the product configuration.



Dependency Check

Dependency check indicates the plug-in or plug-ins on which the current plug-in is dependent upon. On this page, you can only review the dependent plug-ins. Dependent plug-ins are defined in the `dependant_plugins` method when you create a plug-in.



Properties

Click the Properties icon to edit the property values. Properties and their values are defined in the `getProperties` method when you create a plug-in.

Edit Template Variables

Click the Edit Template Variables link to define properties and their values that will be used at run time. The template variables are common to all the groups within a template.

Editing Configuration Plug-in Properties

Use the Configuration Properties page to edit plug-in properties. You define properties and their values for a plug-in when you create the plug-in. When you create a template or run a configuration process, you may edit the property values.

Access the Configuration Properties page using the following navigation path:

PeopleTools, Automated Config Manager, ACM Templates, Template Definition and click the Properties icon for a configuration plug-in that you want to edit.

Image: Configuration Properties page

This example illustrates the fields and controls on the Configuration Properties page. You can find definitions for the fields and controls later on this page.

Configuration Properties			Find View All	First	1-15 of 20	Last
Property	Value	Prompt User?				
★ env.pia_webserver_host	@host@@domain@	<input type="checkbox"/> ?				
★ env.pia_webserver_port	@httpport@	<input type="checkbox"/> ?				
env.pia_webserver_ssl_port	@sslport@	<input type="checkbox"/> ?				
★ env.pia_site_name	@piasite@	<input type="checkbox"/> ?				
★ env.gateway_host	@host@@domain@	<input type="checkbox"/> ?				
★ env.gateway_port	@httpport@	<input type="checkbox"/> ?				
★ env.gateway_ssl_port	@sslport@	<input type="checkbox"/> ?				
★ env.use_ssl_gateway	false	<input type="checkbox"/> ?				
★ env.use_ssl_webserver	false	<input type="checkbox"/> ?				
★ env.default_user_id	@userid@	<input type="checkbox"/> ?				
★ env.default_local_node_pass		<input checked="" type="checkbox"/> ?				
env.anonymous_default_user_id	@userid@	<input type="checkbox"/> ?				
★ env.default_local_node	@localnode@	<input type="checkbox"/> ?				
env.wsd_external_user_id	@userid@	<input type="checkbox"/> ?				
env.wsd_external_pass		<input type="checkbox"/> ?				

Note: Property values enclosed within @ indicates that these values are supplied by template variables at run time. To edit a template variable, click the Edit Template Variables link on the Configuration Template Definition page.



Required Property icon

Indicates a property that is required and a value must be defined.

Prompt User

Select the Prompt User check box if you want a user to enter a value at run time.

Generally, a prompt for a user to enter a value is required for user name or password.



Help icon

Hover over the help icon to view a description of a property.

Current Values in DB

Click the Current Values in DB button to review the current values in the database for the properties.

Initially, the values for properties are defined when creating a plug-in. Subsequently, if you modify the values, the modified values are displayed when you click the Current Values in DB button.

Defining Template Variables

You assign a value to a template variable which is in-turn used to assign values for properties in cases where values are recurring in multiple places in a template or template file.

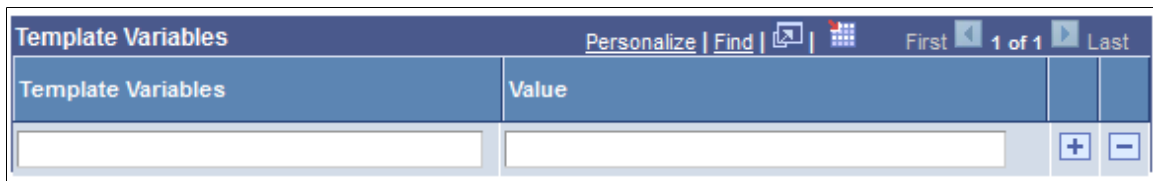
Use the Template Variables dialog box to assign values to a template variable.

Access the Template Variables dialog box using the following navigation path:

PeopleTools, Automated Config Manager, ACM Templates, Template Definition and click the Edit Template Variables link.

Image: Template Variables dialog box

This example illustrates the fields and controls on the Template Variables dialog box.



Note the following when you assign values for template variables:

Template Variable Value	Description
@webhist	Prefix a value with @webhist if the value will be supplied from web history. For example: @webhist.WEBSERVERNAME These values are valid only if a user logs on to PIA before running the configuration program using Automated Configuration Manager.
%	Prefix a value with % if the variable is a PeopleSoft system variable that will be supplied at run time. For example: %ToolsRelease

Importing a Template Using PIA

Use the Configuration Template Import page to import a template file.

Access the Configuration Template Import page by using the following navigation path:

PeopleTools, Automated Config Manager, ACM Templates, Template Export/Import

Image: Configuration Template Import page

This example illustrates the fields and controls on the Configuration Template Import page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web interface for importing a configuration template. At the top, there are two tabs: 'Template Export' and 'Template Import', with 'Template Import' being the active tab. Below the tabs, the main heading is 'Configuration Template Import' in a large blue font, followed by the subtitle 'Upload template file to ACM repository' in a smaller black font. The form contains three input fields: a single-line text box for '*Template Name', a single-line text box for '*Template Description', and a multi-line text area for 'Long Description'. The text area has a small icon in the bottom right corner.

Note: When a template is imported into an environment (from a text file), it is assumed that all passwords contained in the template file are encrypted, so the `decrypt_password` property in the template file must be set to `True`.

Adding a Template to a Permission List

After you create a template, you must add the template to a permission list. PeopleTools provides the PTPT4800 specifically for the purpose of automated configuration.

Use the ACM Templates page to add a template to the PTPT4800 permission list.

Access the ACM Templates page by using the following navigation path:

PeopleTools, Security, Permission & Roles, Permission Lists and click the ACM Templates tab.

Image: Permission Lists - ACM Templates page

This example illustrates the fields and controls on the Permission Lists - ACM Templates page.

Permission List: PTPT4800
Description: ACM Administrator

ACM Templates				
Personalize Find [Icons] First 1-5 of 5 Last				
	Template Name			
1	CLUSTER_TEMPLATE	[Search]	[+]	[-]
2	IB_TEMPLATE	[Search]	[+]	[-]
3	PRCS_SCHEDULER_TEMPLATE	[Search]	[+]	[-]
4	SES_TEMPLATE	[Search]	[+]	[-]
5	UTILITY_TEMPLATE	[Search]	[+]	[-]

Add a template to the permission list.

Note: If a template is not added to the PTPT4800 permission list, you cannot use the template for a configuration run.

Chapter 4

Working with Automated Configuration Using the Command Line

Understanding Automated Configuration Using the Command Line

PeopleTools provides an automated configuration management framework that enables you to store your environment configuration settings in a persistent state in the form of a template file. After setting up the basic infrastructure of a PeopleSoft environment, including database, application server, Process Scheduler server, and PIA domain, you run an Application Engine program, PTEM_CONFIG. This Application Engine program reads your environment properties from the stored template file, and inserts the stored values into the database, saving you from updating the settings manually each and every time you create or refresh an environment.

The settings stored in the template file are those that you typically enter on a configuration page in PeopleTools and save to the database. For example, the value for your Integration Gateway URL, which you would normally add manually on the Gateways page, can be entered in your template file once, and then retrieved by the automated configuration Application Engine program and inserted into the database each time you refresh that environment.

The PTEM_CONFIG Application Engine program, fires configuration plug-ins that take the properties specified for a particular plug-in, as specified in the template file, and configures the features associated with that plug-in. You enable configuration plug-ins by referencing them in your properties file and specifying the settings for that configuration plug-in.

The automated configuration management framework enables you to create the template file. The file can be created either manually or through PIA.

Understanding the Template File

You store your environment settings in a template file that you create either manually or through PIA. You save this file to a location on your system, which is identified by the PS_FILEDIR environment variable. You enable the desired configuration plug-ins by including a reference to that plug-in in the template file, followed by the expected configuration properties for that plug-in. When you run the automated configuration Application Engine program, the program engages the specified configuration plug-ins and inserts the configuration values you've included into the database.

To manually create a template file, see [Creating a Template File Manually](#).

To create a template file through PIA, see [Creating a Template File in PIA](#).

Creating a Template File Manually

This section discusses:

- Working with the structure and general settings of a template file.
- Specifying general settings.
- Specifying template settings.
- Enabling configuration plug-ins.

Working with the Structure and General Settings of a Template File

This section discusses:

- Naming the template file.
- Setting the structure of the template file.

Naming the Template File

Using the text editor for your operating system, create a new file and save it using the following convention:

<file_name>.<file_extension>

The file extension can be .txt, .properties, or any extension that is readable in ASCII format.

Note: You need to specify the template name using the PTEM_PROP environment variable prior to running the PTEM_CONFIG Application Engine program.

Setting the Structure of the Template File

The content of the properties should be structured in the following order:

1. General settings.
2. Template settings.
3. Individual configuration plug-in settings.

To enable a plug-in, you include a reference to that plug-in, such as:

```
plugin.IB.1=PTEM_CONFIG:PTIBRenameNode
```

where

- IB - refers to the category of the plug-in.
- 1- represents the sequence number in which plug-ins are configured.

- `PTEM_CONFIG:PTIBRenameNode` - represents the plug-in name, which consists of application package name (`PTEM_CONFIG`) and class name (`PTIBRenameNode`).

Then, you include the properties for that plug-in immediately following the plug-in reference. The following example illustrates a sample template file:

```
configure=true
verify=true
decrypt_password=true

template.domain=@webhist.SESSIONCOOKIEDOM
template.host=@webhist.WEBSERVERNAME
template.httpport=@webhist.PORTALHTTPPORT
template.jslport=@webhist.jolt_port
template.localnode=%LocalNode
template.piasite=@webhist.ps.discovery.siteName
template.ses_port=7777
template.seshost=seshost
template.sslport=@webhist.PORTALHTTPSPORT
template.tools_release=%ToolsRelease
template.userid=%UserId

#Group: Integration Broker
#Description: Integration Broker Configurations

##### Rename the default local node#####

plugin.IB.1=PTEM_CONFIG:PTIBRenameNode

#Default local node
env.default_local_node=@localnode@

#Purge application server messages
env.app_msg_purge_all_dms=true
```

Note: All other configuration plug-ins you want to enable can be included in the file as needed.

Specifying General Settings

The general settings appear at the top of the template file. These settings apply to the overall configuration program run.

<i>Property</i>	<i>Description</i>
<code>configure</code>	Enables you to set the entire template file as active by setting it to true. If set to false, the <code>PTEM_CONFIG</code> Application Engine program ignores the plug-ins and settings stored within the template file.
<code>verify</code>	If set to true, the automated configuration management framework verifies the settings. The parameters provided in the template file are validated against the corresponding values inserted into the database to ensure they are identical.
<code>encrypt_password</code>	If set to true, any sensitive data stored in the template file will be encrypted, and it will generate a template file where the sensitive data is encrypted.
<code>decrypt_password</code>	Set to true if the template file contains any encrypted sensitive data.

Property	Description
generate_property_file	Set to true if the template file consists only plug-in names and the properties for each plug-in is to be generated. If set to True, the property generates a new file (new_config_file.txt) with the plug-in properties in the same path where the template file is stored.

Specifying Template Settings

Template settings are a group of settings that apply to almost all configuration plug-ins and can be shared across multiple property settings. For example, HTTP port is a template setting, and rather than specifying it for each and every configuration value where it is needed (PIA 'URL, URI URL, Integration Gateway URL, and so on), you can specify it once in the template settings, and it will be replicated as needed throughout the template file, where referenced. This avoids unnecessary repetition and errors.

You can use %SystemVariable to default the value to a system variable; and use

%ENV_VARIABLE_NAME% to set.

You can also retrieve stored values in the web profile web history, by using

@webhist.<property_name>.

The following example illustrates the template settings section of a sample template file:

```
template.domain=@webhist.SESSIONCOOKIEDOM
template.host=@webhist.WEBSERVERNAME
template.httpport=@webhist.PORTALHTTPPORT
template.jslport=@webhist.jolt_port
template.localnode=%LocalNode
template.piasite=@webhist.ps.discovery.siteName
template.ses_port=7777
template.seshost=seshost
template.sslport=@webhist.PORTALHTTPSPORT
template.tools_release=%ToolsRelease
template.userid=%UserId
```

Property	Description
template.domain	Name of the PIA domain, such as 'peoplesoft'.
template.host	Server host for the web server installation.
template.httpport	The HTTP port on which the PIA domain listens.
template.jslport	Jolt port.
template.localnode	The current node used by the system. %Dbname or %LocalNode can be used if applicable.
template.piasite	The name of the PeopleSoft site.
template.ses_port	The port on which Oracle SES listens for requests, such as 7777 or 7778.
template.seshost	Server host where Oracle SES is installed.

<i>Property</i>	<i>Description</i>
template.sslport	The HTTPS port on which the PIA domain listens.
template.tools_release	PeopleTools release.
template.userid	User ID for accessing and configuring the system.

Enabling Configuration Plug-ins

You can include the configuration plug-ins in this section within your template file.

The following example illustrates the PTIBRenameNode configuration plug-in. In this example, the plug-in is not enabled because the plug-in is commented (#). To enable the plug-in, delete the (#) hash sign preceding the plug-in name. For example, plugin.IB.1=PTEM_CONFIG:PTIBRenameNode.

```
##### Rename the default local node#####
#plugin.IB.1=PTEM_CONFIG:PTIBRenameNode

#Default local node
env.default_local_node=@localnode@

#Purge application server messages
env.app_msg_purge_all_dms=true
```

Note: All other configuration plug-ins you want to enable can be included in the file as needed.

In PIA, you can enable a plug-in in a template by selecting the check box corresponding to the plug-in on the Configuration Template Definition page.

See [Creating a Template](#).

Creating a Template File in PIA

The automated configuration management framework enables you to create a template file by exporting a template file in PIA. The template file is saved with a .txt extension and can be used as an input file when configuring through command line.

Exporting a Template

Use the Configuration Template Export page to create a text file of the settings and configuration plug-ins of the selected template.

Access the Configuration Template Export page using the following navigation path:

PeopleTools, Automated Config Manager, ACM Templates, Template Export/Import

Image: Configuration Template Export page

This example illustrates the fields and controls on the Configuration Template Export page. You can find definitions for the fields and controls later on this page.

Configuration Template Export

Download the template as text file

Configuration Template Export	
Template Name	Download
CLUSTER_TEMPLATE	Download
IB_TEMPLATE	Download
OTHER_UTILITY_TEMPLATE	Download
PRCS_SCHEDULER_TEMPLATE	Download
SES_TEMPLATE	Download

Download

Click to create and save the template file as a text file.

Note: If the template contains any passwords, the passwords are encrypted.

Running the PSRUNACM Script

In addition to the Automated Configuration Manager that is executed from PIA, the automated configuration management framework allows you to configure the products for your environment from the command line using the PSRUNACM script.

The PSRUNACM.bat and the PSRUNACM.sh files are available in the Utility folder of PS_HOME.

This section discusses:

- Running PSRUNACM in Windows.
- Running PSRUNACM in UNIX.

Running PSRUNACM in Windows

To run PSRUNACM in Windows:

1. Set the PS_HOME and PS_CFG_HOME environment variables.
2. Set the PS_FILEDIR environment variable, if required.

The template file that is used as input to configure the environment can be present either in database or as a text file.

- If the template file is present in database, do not set the PS_FILEDIR environment variable.
- A template file (text file) can exist in a remote path or you can copy the template file (text file) into the same directory as the psrunACM script.

Note: If the template file (text file) exists in a remote path, you must set the PS_FILEDIR environment variable to specify the remote path of the template file. If the template file exists in a remote path and if the PS_FILEDIR variable is not set, the directory from where the psrunACM script is run is considered as the location of the template file. Hence, you need to copy the template file to the same directory as the psrunACM script.

- If you copy the template file into the same directory as the psrunACM script, do not set the PS_FILEDIR environment variable.

3. Run the PSRUNACM.bat file.

- If the template file is present in database, run the following command by replacing the values within <> with appropriate values.

```
psrunACM.bat <SERVER> <DBTYPE> <DBNAME> <USERID> <PASSWORD> <TEMPLATE_NAME> <OPTION>
```

where

SERVER	The host machine name in which the script is executed.
DBTYPE	The type of database on which the configuration is run, for example, ORACLE.
DBNAME	The name of the hosted database.
USERID	Operator ID.
PASSWORD	Operator password.
TEMPLATE_NAME	The name of the template file, which is used as input for the configuration run.
OPTION	The option parameter is used to decide whether the template should be executed from database, imported to database, or exported from database.

Valid values:

- EXP - The template mentioned in TEMPLATE_NAME is exported from the database mentioned in DBNAME to the local machine.
- IMP - The template mentioned in TEMPLATE_NAME is imported to the database mentioned in DBNAME from the current path from where the psrunACM.bat script is invoked.

- EXEC - The template mentioned in TEMPLATE_NAME is used as input to configure the environment.
- If the template file is available as a text file, run the following command by replacing the values within <> with appropriate values.

Before you run the command, ensure that the PS_FILEDIR environment variable is set if the template file exists in a remote path.

```
psrunACM.bat <SERVER> <DBTYPE> <DBNAME> <USERID> <PASSWORD> <TEMPLATE_NAME>
E>
```

Note: If the input is a text file, the OPTION parameter is not required.

Running PSRUNACM in UNIX

To run PSRUNACM in UNIX:

1. Set the PS_HOME and PS_CFG_HOME environment variables.
2. Set the PS_FILEDIR environment variable, if required.

The template file that is used as input to configure the environment can be present either in database or as a text file.

- If the template file is present in database, do not set the PS_FILEDIR environment variable.
- A template file (text file) can exist in a remote path or you can copy the template file (text file) into the same directory as the psrunACM script.

Note: If the template file (text file) exists in a remote path, you must set the PS_FILEDIR environment variable to specify the remote path of the template file. If the template file exists in a remote path and if the PS_FILEDIR variable is not set, the directory from where the psrunACM script is run is considered as the location of the template file. Hence, you need to copy the template file to the same directory as the psrunACM script.

- If you copy the template file into the same directory as the psrunACM script, do not set the PS_FILEDIR environment variable.
3. Run the PSRUNACM.sh script.
 - If the template file is present in database, run the following command by replacing the values within <> with appropriate values.

```
psrunACM.sh <SERVER> <DBTYPE> <DBNAME> <USERID> <PASSWORD> <TEMPLATE_NAME>
> <DOMAIN> <OPTION>
```

where

SERVER

The host machine name in which the script is executed.

DBTYPE	The type of database on which the configuration is run, for example, ORACLE.
DBNAME	The name of the hosted database.
USERID	Operator ID.
PASSWORD	Operator password.
TEMPLATE_NAME	The name of the template file, which is used as input for the configuration run.
DOMAIN	The application server domain name.
OPTION	The option parameter is used to decide whether the template should be executed from database, imported to database, or exported from database.

Valid values:

- EXP - The template mentioned in TEMPLATE_NAME is exported from the database mentioned in DBNAME to the local machine.
- IMP - The template mentioned in TEMPLATE_NAME is imported to the database mentioned in DBNAME from the current path from where the psrunACM.sh script is invoked.
- EXEC - The template mentioned in TEMPLATE_NAME is used as input to configure the environment.

- If the template file is available as a text file, run the following command by replacing the values within <> with appropriate values.

Before you run the command, ensure that the PS_FILEDIR environment variable is set if the template file exists in a remote path.

```
psrunACM.sh <SERVER> <DBTYPE> <DBNAME> <USERID> <PASSWORD> <TEMPLATE_NAME=>
> <DOMAIN>
```

Note: When the input is a text file, the OPTION parameter is not required.

Running the PTEM_CONFIG Application Engine Program

This section provides an overview and discusses:

- Ensuring sufficient security permissions.
- Setting the environment variables for the command line session.

- Running the PTEM_CONFIG Application Engine program.
- Verifying your configuration settings.

Understanding the PTEM_CONFIG Program Run

After you have created the template file and entered all the required property values for your environment for the configuration plug-ins you have referenced, you load the values into the database by running the PTEM_CONFIG Application Engine program. The PTEM_CONFIG Application Engine program, which you run from the command line, determines the location of your environment template file by the PS_FILEDIR environment variable setting.

The following instructions assume you are familiar with running Application Engine programs. For more information on running Application Engine programs from the command line, see "Using the Command Line to Invoke Application Engine Programs" (PeopleTools 8.54: Application Engine).

Ensuring Sufficient Security Permissions

Make sure the user ID running the Application Engine program has the appropriate security access. To run the PTEM_CONFIG Application Engine program, the user you submit on the command line must be associated with the PeopleSoft Administrator or the ACM Administrator role in PeopleTools Security.

Likewise, the user needs to have any required security for any of the products being automatically configured. For example, if you are configuring the Search Framework, the user ID needs to be associated with the required roles for that product, such as Search Administrator, Search Server, and so on.

See [Security Permissions Required for Automated Configuration Management](#).

Setting the Environment Variables for the Command Line Session

Open the command line utility for your operating system and set the following environment variables.

- Set the PS_HOME environment variable to reflect the path of PS_HOME directory.

For example:

```
D:\>set PS_HOME=D:\PT8.54
```

- Set the PS_FILEDIR environment variable to reflect the location of the template file for the current command line session.

For example:

```
D:\>set PS_FILEDIR=D:\PTSYS_AUTOCONFIG
```

In this case, the template file is located in D:\ptsys_autoconfig.

- Specify the name of the template file for Application Engine program input using the PTEM_PROP environment variable.

For example:

```
D:\>set PTEM_PROP=IB_TEMPLATE.txt
```

- Specify whether to execute, import, or export a template in database using the PTACM_OPTION environment variable.

Valid values:

- Execute - This option will override the template file given in PTEM_PROP and take the template from database (template name will be provided in the PTACM_TEMPLATE environment variable).
- Import - this option will import a template file given in PTEM_PROP to template provided in PTACM_TEMPLATE.
- Export - this option will export the template in PTACM_TEMPLATE to template file given in PTEM_PROP.

For example:

```
D:\>set PTACM_OPTION=execute
```

- If the PTACM_OPTION is set, specify the template name using the PTACM_TEMPLATE environment variable.

Running the PTEM_CONFIG Application Engine Program

In the same command line session where you set PS_FILEDIR to point to your environment variables template file, launch the PTEM_CONFIG Application Engine program.

Change directories to PS_HOME\bin\client\winx86 and submit the following command line parameters to the Application Engine executable (psae.exe) replacing the values in <> with appropriate values.

For example:

```
psae.exe -CT <DB type> -CS <server name> -CD <DB name> -CO <user> -CP <password> ->
R <runid> -AI PTEM_CONFIG -I <instance number>
```

Parameter	Description
-CT	Database type (connection type).
-CS	Server name (used only for specific database types).
-CD	Database name.
-CO	User ID connecting to the database.
-CP	Password associated with the user ID.
-R	Run control ID.
-AI	Application Engine program name (Application ID)
-I	Instance number.

See the Application Engine documentation for more information on other optional command line parameters.

After the PTEM_CONFIG program processes all the configuration properties for the configuration plug-ins contained in the environment variables template file, you should see a message indicating the successful program run.

Verifying Your Configuration Settings

Depending on the configuration plug-ins you included, test the appropriate settings in your environment to confirm that the values you entered in the environment variables template file are correct.

For example, if you included Integration Broker configuration plug-ins, make sure the elements related to Integration Broker are configured correctly.

Chapter 5

Working with Sensitive Data

Understanding Password Encryption in Template File

Sensitive data, like passwords, can be encrypted so that these properties are not exposed if you share the template file among multiple environments and multiple development or testing teams.

The following properties are currently identified as being sensitive data:

- userpass
- SES_administrator_password
- env.default_local_node_pass
- env.gateway_password
- env.ib_appserver_domain_password
- env.gateway_keystore_password
- env.ses_proxy_password

To encrypt sensitive data, run the PTEM_CONFIG Application Engine program with the property `encrypt_password` set to `True` in the template file. This will generate a new template file named *<original template file name>_encrypted* in the location of the original template file.

The encrypted template file can be used as input to run the PTEM_CONFIG Application Engine program by providing the property `decrypt_password=true` in the template file, which internally decrypts the sensitive properties for configuring the environment.

Migrating Configuration Data

Migrating Configuration Data Using ADS

PeopleSoft provides ADS definitions to migrate configuration data from one environment to another as a project file.

For information on using Data Set Designer and other functionality of data sets, see [Data Migration Workbench](#).

Tables for Configuration Data Migration

The following tables store configuration data that can be used for migrating data.

ADS Definition	Description
PTACM_PLUGIN	Table for configuration plug-ins. This table contains the plug-in application class name, its description and category.
PTACM_TEMPLATE	Table for configuration templates. Templates contain plug-ins and their properties for configuration.
PTACM_TEMP_VAR	Table for configuration template variables. This table contains the template variable, which can be used to assign the values to the plug-in properties.

ADS Definition for Template

Use the PTACM_TEMPLATE ADS definition to export templates and related plug-ins.

The PTACM_TEMPLATE ADS definition uses the following tables.

Table	Description
PTACM_TEMPLATE	Stores template data.
PTACM_TEMP_VAR	Stores template variable data.
PTACM_ADS_RELN	Stores data pertaining to relationship between templates and plug-ins.

ADS Definition for Plug-in

Use the PTACM_PLUGIN ADS definition to export only plug-ins.

Exporting Configuration Data

In addition to the ADS definitions, you can also export configuration data using the export functionality available within Automated Configuration Manager. The exported template file can be used as input for command line when you run a product configuration on another system.

See [Creating a Template File in PIA](#).

Appendix A

Sample Template File

Sample Template File

The following example shows a sample template file:

```
template_name=IB_TEMPLATE

configure=true
verify=true
decrypt_password=true

template.domain=@webhist.SESSIONCOOKIEIDOM
template.host=@webhist.WEBSERVERNAME
template.httpport=@webhist.PORTALHTTPPORT
template.jslport=@webhist.jolt_port
template.localnode=%LocalNode
template.piasite=@webhist.ps.discovery.siteName
template.ses_port=7777
template.seshost=seshost
template.sslport=@webhist.PORTALHTTPSPORT
template.tools_release=%ToolsRelease
template.userid=%UserId

#Group: Integration Broker
#Description: Integration Broker Configurations

##### Rename the default local node#####

#plugin.IB.1=PTEM_CONFIG:PTIBRenameNode

#Default local node
env.default_local_node=@localnode@

#Purge application server messages
env.app_msg_purge_all_dms=true

##### Configure the DB node#####

plugin.IB.2=PTEM_CONFIG:PTIBConfigureDBNode

#PIA web server host
env.pia_webserver_host=@host@@domain@

#PIA web server port
env.pia_webserver_port=@httpport@

#PIA web server SSL port
env.pia_webserver_ssl_port=@sslport@

#PIA site name
env.pia_site_name=@piasite@

#Gateway host
env.gateway_host=@host@@domain@

#Gateway port
env.gateway_port=@httpport@

#Gateway SSL port
```

```

env.gateway_ssl_port=@sslport@

#Flag determines if the configuration is secure(https) / non secure (http); True-> =>
https, false->http
env.use_ssl_gateway=false

#Flag determines if the configuration is secure(https) / non secure (http); True-> =>
https, false->http
env.use_ssl_webserver=false

#Default user name
env.default_user_id=@userid@

#Default local node password
env.default_local_node_pass=

#Anonymous default user name
env.anonymous_default_user_id=@userid@

#Default local node
env.default_local_node=@localnode@

#External user id for wsdl_node
env.wsdl_external_user_id=@userid@

#External user password for wsdl_node
env.wsdl_external_pass=

#Flag to configure WSDL node
env.configure_wsdl_node=false

#Authentication TokenType for WSDL; Possible Values: NONE=none,STSD=SAML Token,USRT=>
=Username Token
env.wsdl_node_tokentype=NONE

#Encryption for WSDL node
env.wsdl_node_tokenencrypted=0

#Digital Signature for WSDL node
env.wsdl_node_tokensigned=0

#Encryption Level for WSDL node; Possible Values: A=All, B=body and H=Header level =>
encryption
env.wsdl_node_ibencryptionlevel=A

##### Configure gateway URL, load connectors, define node in the gateway#####
plugin.IB.3=PTEM_CONFIG:PTIBConfigureGatewayNodes

#Gateway host
env.gateway_host=@host@@domain@

#Gateway port
env.gateway_port=@httpport@

#Gateway SSL port
env.gateway_ssl_port=@sslport@

#Flag determines if the configuration is secure(https) / non secure (http); True-> =>
https, false->http
env.use_ssl_gateway=false

#Default local node
env.default_local_node=@localnode@

#Gateway user name
env.gateway_user=administrator

#Gateway user password
env.gateway_password=

```

```

#Application server host name
env.ib_appserver_host=@host@@domain@

#Jolt port
env.ib_jolt_port=@jslport@

#Node proxy user name
env.ib_node_proxy_userid=@userid@

#Node proxy user password
env.ib_node_proxy_password=

#Tools Release
env.tools_release=@tools_release@

#Application server domain password
env.ib_appserver_domain_password=

#Virtual node
env.ib_virtual_node=

#Set as default node if this is set to true
env.ib_set_as_default_node=true

#Execute MAP File Synchronization
env.ib_synchronize_map_files=Y

##### Set keystore password, proxy host, proxy port, non proxy host in gateway prop⇒
erties file#####

plugin.IB.4=PTEM_CONFIG:PTIBConfigureGatewayProperties

#Gateway host
env.gateway_host=@host@@domain@

#Gateway port
env.gateway_port=@httpport@

#Gateway SSL port
env.gateway_ssl_port=@sslport@

#Flag determines if the configuration is secure(https) / non secure (http); True-> ⇒
https, false->http
env.use_ssl_gateway=false

#Gateway user name
env.gateway_user=administrator

#Gateway user password
env.gateway_password=

#Gateway keystore password
env.gateway_keystore_password=

#Gateway proxy host
env.gateway_proxy_host=

#Gateway proxy port
env.gateway_proxy_port=

#Gateway non proxy hosts
env.gateway_non_proxy_hosts=@host@@domain@|localhost|*.oracle.com

##### Active IB Pub/Sub Domain.#####

#plugin.IB.5=PTEM_CONFIG:PTIBActivateDomain

#Active Domain Retry Count
domain.activate_retry_count=10

#Active Domain Wait Time

```

```
domain.activate_wait_time=10

##### Activate IB Queues.#####

#plugin.IB.6=PTEM_CONFIG:PTIBActivateQueues

#Possible Values: PS_ALL to activate all queues, Else provide comma separated value→
s for queue names
queue.activate_queue_list=PS_ALL

#Active Queue Status; Default Value of queue status is 1
queue.activate_queue_status=1
```