

CA BEST PRACTICES

CA IT Process Automation Manager Best Practices

Troubleshooting and Diagnostics
Guidelines

DRAFT DOCUMENT – [FEEDBACK](#) WELCOME!

LEGAL NOTICE

This publication is based on current information and resource allocations as of its date of publication and is subject to change or withdrawal by CA at any time without notice. The information in this publication could include typographical errors or technical inaccuracies. CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice.

Any reference in this publication to non-CA products and non-CA websites are provided for convenience only and shall not serve as CA's endorsement of such products or websites. Your use of such products, websites, and any information regarding such products or any materials provided with such products or at such websites shall be at your own risk.

Notwithstanding anything in this publication to the contrary, this publication shall not (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this publication remain at CA's sole discretion.

The information in this publication is based upon CA's experiences with the referenced software products in a variety of development and customer environments. Past performance of the software products in such development and customer environments is not indicative of the future performance of such software products in identical, similar or different environments. CA does not warrant that the software products will operate as specifically set forth in this publication. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product.

Certain information in this publication may outline CA's general product direction. All information in this publication is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "AS IS" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if CA is expressly advised of the possibility of such damages.

COPYRIGHT LICENSE AND NOTICE:

This publication may contain sample application programming code and/or language which illustrate programming techniques on various operating systems. Notwithstanding anything to the contrary contained in this publication, such sample code does not constitute licensed products or software under any CA license or services agreement. You may copy, modify and use this sample code for the purposes of performing the installation methods and routines described in this document. These samples have not been tested. CA does not make, and you may not rely on, any promise, express or implied, of reliability, serviceability or function of the sample code.

Copyright © 2010 CA. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. Microsoft product screen shots reprinted with permission from Microsoft Corporation.

TITLE AND PUBLICATION DATE:

CA IT Process Automation Manager Best Practices: Troubleshooting and Diagnostics Guidelines
Publication Date: April 2010

ACKNOWLEDGEMENTS

Principal Authors and Technical Editors

George Curran
Alex Moscoso
Terry Pisauro

The principal authors and CA would like to thank the following contributors:

Ashish Pokharel
Daniel Zilberman
CA Services
Development
Marketing
QA
Support
SWAT
Technical Sales
Technical Information

CA PRODUCT REFERENCES

This document references the following CA products:

- CA IT Process Automation Manager™ (CA IT PAM)
- CA Embedded Entitlements Manager (CA EEM)
- CA Enterprise Log Manager

FEEDBACK

Please email us at impcdfedback@ca.com to share your feedback on this publication. Please include the title of this publication in the subject of your email response. For technical assistance with a CA product, please contact CA Technical Support at <http://ca.com/support>. For assistance with support specific to Japanese operating systems, please contact CA at <http://www.casupport.jp>.

Contents

Chapter 1: Introduction	7
Chapter 2: Process Issues	9
Troubleshooting Tools.....	9
Using Process Watch to Determine Why a Process Failed.....	10
Using the Default Process Watch to Identify a Failed Process Instance	10
Using a Custom Process Watch to identify a Failed Process Instance.....	12
How Can You Be Notified if a Process Failed?	15
Determining if a Process Definition Has Changed?	16
Inspecting User Inputs and Other Parameters Values for a Process instance	18
Determining which step in a Process Instance Failed	19
How do you View the Failed Step Graphically.....	19
How do you View Text Logs for a Failed Process Instance?	20
Chapter 3: Component Issues	23
What External Factors Can Impact CA IT PAM Components Behavior?	23
Determining if a CA IT PAM Component is Running.....	23
Determining if a CA IT PAM Component Configuration has Changed?	23
Using a LogViewer Object to Identify Component Faults.....	25
Locating and Checking CA IT PAM Logs Directly for Component Faults	27
Modifying the Logging Configuration.....	28
Modifying the Logging Configuration for an Orchestrator	28
Modifying Logging Configuration for an Agent.....	31
Troubleshooting Common Areas.....	33
Cannot Start "CA IT PAM Orchestrator Service"	33
Cannot Get to Administrator UI.....	33
Cannot Login to Administrator UI	34
Cannot Launch IT PAM Client.....	34
Cannot See Agents in the Orchestrator	34
Cannot Start the Windows Agent	35
Cannot Start *NIX Agents	35
Agent Not Communicating with Orchestrator	36
Database Errors.....	36
Chapter 4: Before You Call Support	39
Appendix A: Tracing Process Operations through Orchestrator and Agent Logs	41
Appendix B: Configuration and Log Files	47
Orchestrator Configuration Files.....	47
Orchestrator Domain.xml	47
Orchestrator OasisConfig.properties	48
Orchestrator c2osvcw.conf and c2osvcd.sh.....	48
Orchestrator log4j.xml.....	49
Orchestrator Log Files.....	49
Agent Configuration Files	50
Agent Domain.xml	50



Agent OasisConfig.properties.....	50
Agent c2oAgtsvcw.conf and c2oagtd.sh	50
Log4j.xml	51
Agent Log Files	51



Chapter 1: Introduction

CA IT Process Automation Manager (CA IT PAM) provides a centralized and structured approach to operations management by enabling you to define, build, orchestrate, manage, and report on automated processes spanning across different teams and roles in your organization. By automating routine administrative tasks, CA IT PAM improves operational efficiency and incident response handling, and ensures best practice and regulatory controls compliance.

This document is one in a series of papers providing best practices for making the most of your CA IT PAM implementation. The focus of this paper is on troubleshooting. It includes details on CA IT PAM log files, configuration options for debugging, and what to do before you call support. It also includes information on common install\usage errors to help you troubleshoot some basic issues on your own. Additional best practices documents include topics such as Maintaining Availability, Security Considerations and Performance Optimization.

Chapter 2: Process Issues

The primary root cause behind problems impacting an individual Process definition or instance of a Process generally falls into one of two categories:

- Changes to Process definitions that may not have been adequately tested prior to implementation
- Unexpected user input or data that cannot be properly handled

Examples include:

- **Misconfigured module parameters.** Parameters can be configured at Domain, Environment, Orchestrator and Agent levels. If these parameters are not configured properly the process will be unable to run. For example, if the process has a start script operator running a script and the user does not have permission to run a script or if the password for that user was changed at the Agent module configuration, the process will be impacted.
- **Problems with the integrated application.** If the application with which the process is to interact is not working as expected, the process will be impacted as well. For example, an operator that is configured to create a Service Desk ticket using web services fails because the Service Desk web service itself is down or a CA SPECTRUM operator used to create alarms fails because the CA IT PAM host was not added as trusted host of CA SPECTRUM client.
- **OS level configurations errors.** Examples of this include a command which was available in the `%path%` of the system in test machine but is not available in the production machine. As a result an operator which has a script to run this command fails. Other examples include a perl script failing on a Windows machine because the file type association is not matched to (perl.exe) and a UNIX `#!/usr/bin/` command that does not point to the right command in the script.
- **Operation raised an exception rather than error.** Typically, when an operator fails, it should follow the "failed" path. In some cases the operators can end up with an exception (for example, the Touchpoint where the operator is running is down, or there is some unhandled issue in the operator, pre-execution or post-execution code or an expression used as an input parameter caused an exception). In these cases, if the "failed or x" path is not followed the process will wait on that operator if the exception handling is not done in process.
- **Swim Lane/Exception Handling issues.** Keep in mind that there may be a problem with parts of process that implement lane change handling and/or exception handling.

Troubleshooting Tools

CA IT PAM includes a number of features to monitor and troubleshoot the Processes it automates. It also provides audit trails to trace and record activity for configuration objects (Domain, Environments, Agents, and Orchestrators), and Library objects (folders and automation objects). You can use the CA IT PAM Client to view audit trail records for any of these objects through the audit trail browser. The information you can view will depend on your user role:



- A member of the Domain administrator role may view the audit trail for the Domain
- A member of the Environment configuration administrator group may view the audit trail for an Environment
- an Environment user with read permission may view the audit trail for an object

Using Process Watch to Determine Why a Process Failed

Process Watch objects are special IT PAM objects that enable you to view the state of all or a subset of IT PAM Processes. Two types of Process Watch objects are available:

- Default Process Watch objects, which, as the name implies, are added, by default, for each Library
- Custom Process Watch objects which can be defined to monitor a subset of Processes

Both can be used to identify and access tools to troubleshoot failed Process instances.

Using the Default Process Watch to Identify a Failed Process Instance

The Default Process Watch can be accessed from either the CA IT PAM web-based Management Console or the CA IT PAM Client. While both function similarly we will focus here on the version exposed via the CA IT PAM Client version since that is the more appropriate and recommended interface for troubleshooting.

To open the Default Process Watch do the following:

1. Open the "Library Browser" in the IT PAM Client (File -> Open Library Browser).

2. Click the "Default Process Watch" icon on the "Library Browser" toolbar



The content displayed in the Process Watch is grouped into the following categories:

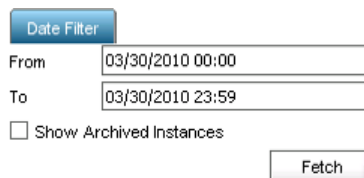
- Abnormally Ended Instances
- Active Agendas
- Active Operators
- Active Systems
- All Instances
- Ended Instances
- Normally Ended Instances

- Queued Instances
- Running Instances
- Suspended Instances
- User Interactions
- Waiting Instances


Depending on which error handling techniques were used in the design of the process (see *CA IT Process Automation Manager Best Practices, Recommendations for Process Design and Monitoring* for details), Processes that have failed or that have encountered an expected state will typically appear under one of the following categories:

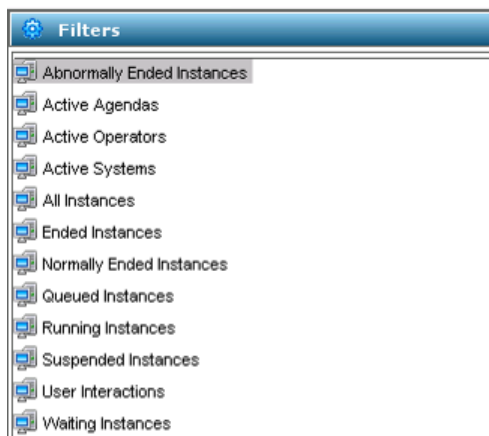
- Abnormally Ended Instances
- Suspended Instances
- Waiting Instances

You can apply a date range filter to further restrict or expand the include content.



For example, to identify the instance of a Process "Data Transfer – Simple" that failed (reached an "Abnormal Stop") on March 26 you would do the following:

1. Open the "Library Browser" for the Library where "Data Transfer – Simple" is defined in the IT PAM Client (File -> Open Library Browser).
2. Click the "Default Process Watch" icon on the "Library Browser" toolbar 
3. Select "Abnormally Ended Instances" from the "Filters" palette:



4. Edit the "Date Filter" to set the appropriate date range for content to be included then click Fetch to reduce the number of entries displayed if necessary:

Date Filter

From

To

Show Archived Instances

Process instances meeting the search criteria (in this example, instances that failed on March 26) would be displayed in the right hand pane of the Process Watch panel.

Using a Custom Process Watch to identify a Failed Process Instance

Since a typical CA IT PAM deployment includes a number of concurrent use cases that are accomplished through several Processes, Datasets, Resources and other IT PAM automation objects, using the Default Process Watch might quickly result in a sensory overload. An alternative to this is to create a Custom Process Watch that includes only the Processes, Datasets and Resources associated with a particular use case.

For example, suppose you have defined a number of Processes that are used to automate options exposed by an "Employee Self-Service" initiative. To define a Custom Process Watch for this do the following:

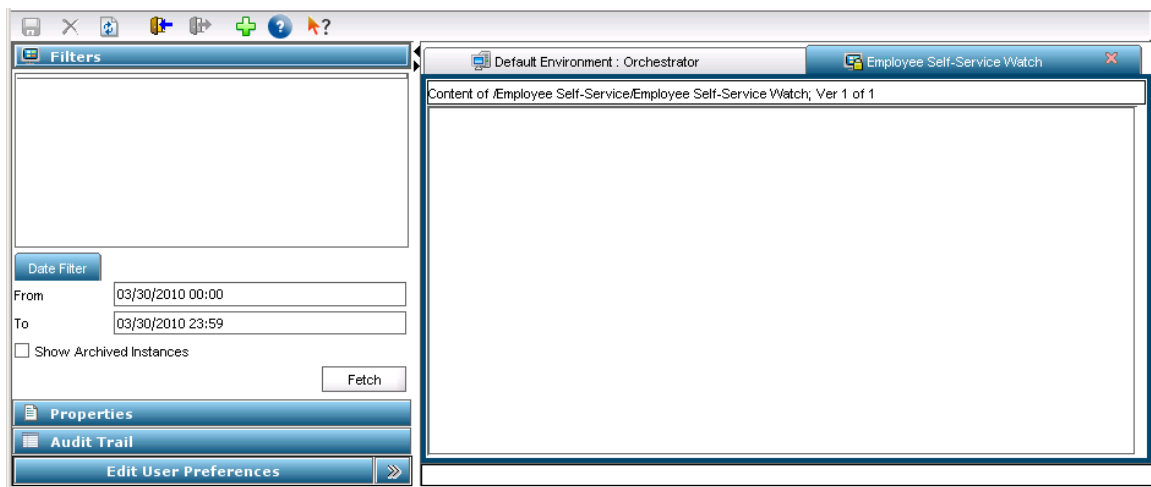
1. Open the "Library Browser" for the Library in which the automation objects are defined in the IT PAM Client (File -> Open Library Browser).
2. Navigate to and select an appropriate Library folder (for example, assume that all the automation objects involved in our scenario have been created in the "Employee Self-Service" folder).
3. Right-click on the Library folder and select "New Object" then "Process Watch" from the in-context menus that appear.

4. Rename the new Process Watch object that appears in the right-hand pane appropriately (for the example, "Employee Self-Service Watch").


To specify the automation objects the new Process Watch should monitor you will need to open the appropriate editor panel:

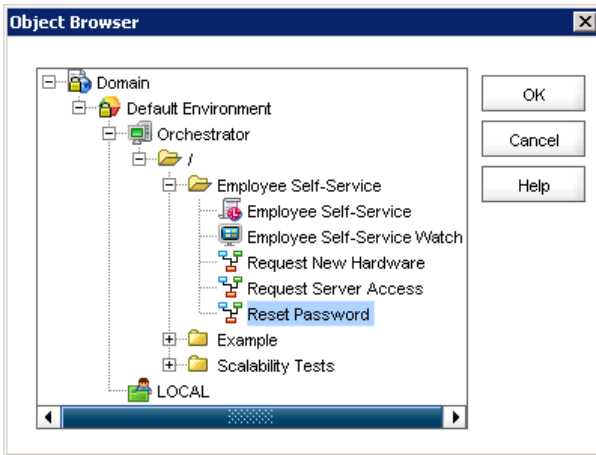
1. Right-click on the new Process Watch object then select "Edit" from the in-context menu that appears.
2. Select the version to load in the editor panel (if the new Process Watch will only have a "Version1" then click the "Edit" button on the "Versions" dialog box that appears).

The editor panel for the new Process Watch appears. For example:

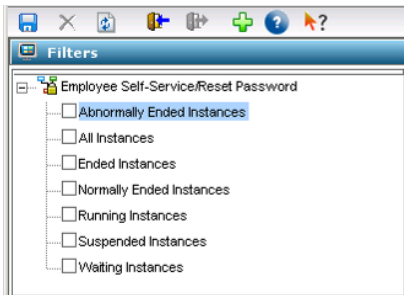


To add a Process to be monitored:

1. Click the add object icon on the toolbar 
2. Navigate to and select the Process in the "Object Browser" dialog that appears (for example the "Reset Password" Process). Then click OK.



- Expand the newly added Process node that appears in the "Filters" palette in the left-hand pane.

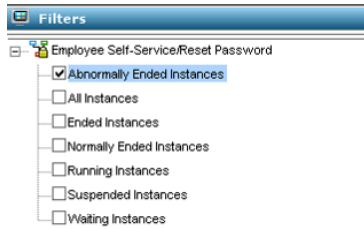


- Choose the states for the Process that should be included, by default, when the Custom Process Watcher is opened (the options displayed will be specific to the object type selected).

Note: The Default Process Watcher will display all states for all automation objects within the Library that it is associated with while a Custom Process Watcher objects will include only the automation objects specified in its definition.

To identify a failed Process instance using a Custom Process Watcher do the following:

- Open the "Library Browser" for the Library where Custom Process Watcher is defined in the IT PAM Client (File -> Open Library Browser).
- Navigate to and double-click the Custom Process Watcher.
- If not specified when the Custom Process Watcher was defined, expand the node or nodes in the "Filters" palette and select the states that should be included. For example, assume you are looking for failed instances ("Abnormally Ended Instances") of the "Reset Password" Process).



4. Optionally, edit the "Date Filter" to set the appropriate date range for content to be included and click Fetch to reduce the number of entries displayed.

Process instances meeting the specified search criteria (in our example, instances that failed on March 26) would be displayed in the right hand pane of the Process Watch panel.

How Can You Be Notified if a Process Failed?

The best practice for generating proactive notifications when a Process fails or behaves unexpectedly is to define one or more Processes that open Service Desk incidents or that generate alerts to responsible parties. Once defined, the notification Process or Processes can be invoked from other Processes, as needed, using the IT PAM "Run Process" Operation (see the Workflow Module section in the *CA IT PAM Programming Reference Guide* for more details).

Opening a service desk incident would be the recommended method for notifying responsible parties when a Process instance requires attention. Most service desks have predefined notification and escalation procedures - eliminating the need to reproduce and manage the information separately. CA IT PAM provides Operators to interact with CA Service Desk and CA Service Management (see *CA IT Process Automation Manager CA Service Desk Connector Guide*). Another option is to use the CA IT PAM SOAP Module Operators to communicate trigger incident creation, notification and escalation for service desks with exposed web services (see *CA IT Process Automation Management Connector Overview*).

If you prefer not to integrate with an existing service desk, you can use the CA IT PAM Operators supplied with the Alert Module to alert operators, administrators, and others about errors or incidents, or to simply inform people that a process or task is complete. CA IT PAM supports sound alerts, e-mail alerts, and telephone or pager alerts. You can use any or all of these depending on your particular requirements. For a condition that requires immediate operator input at a console, you can use a sound alert. For urgent error conditions, you may need to use a TAPI Alert to reach an administrator by phone or pager. For less urgent errors or reports, you can use a MAPI Alert to send e-mail messages. Different alerts can be used for different users. For example, you might have a group for managers that receive monthly reports, and an administrator who is notified after a process succeeds or fails. The notification Process objects would need to not only trigger alerts but also handle escalation and perform other tasks related to those alerts.

Note: It is better to use email as a general mode of alerts (or in addition to other modes), because message delivery is fairly well guaranteed. Telephony alerts depend on modems and external networks and hardware, which may fail to deliver a critical message.

Determining if a Process Definition Has Changed?

When a Process that had been functioning normally suddenly begins to behave unexpectedly or to fail it is reasonable to suspect that the unexpected behavior may be the result of a change made to the Process definition. CA IT PAM records user activities related to the addition, deletion and modification of automation objects. The history of these activities can be displayed in an Audit Trail.

To view the Audit Trail for a Process definition, do the following:

1. Open the "Library Browser" for the Library in which the Process is defined in the CA IT PAM Client (File -> Open Library Browser).
2. Navigate to the folder containing the Process definition in the left-hand pane.
3. Right-click on the Process definition in the right-hand pane and select "Edit" from the in-context menu that appears.

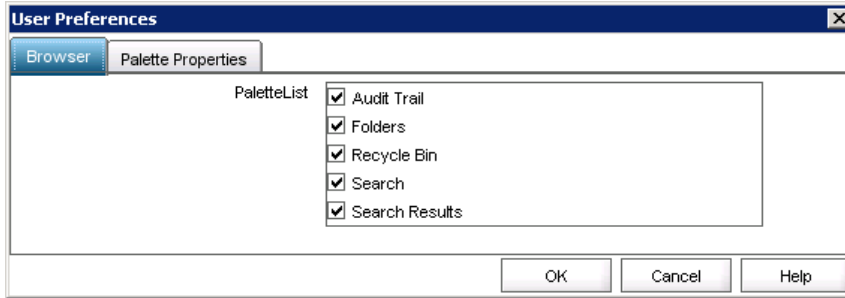
The Process editing panel is displayed for the selected Process. If the "Audit Trail" palette is not displayed in the left-pane do the following:

1. Click the double-arrow button (">>") adjacent to the "Edit User Preferences" bar at the bottom left of the panel.
2. Click the "Audit Trail" entry from the in-context menu that appears.



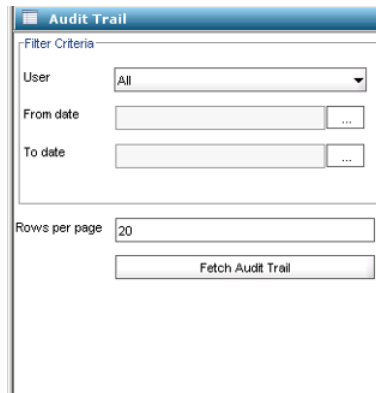
If the "Audit Trail" palette is not available do the following:

1. Click the "Edit User Preferences" bar.
2. Enable (check) the "Audit Trail" entry in the "Palette List" on the "Browser" tab of the "User Preferences" dialog that appears.



- (Optionally) increase the value for "Number of visible palettes" on the "Palette Properties" tab sufficiently to make the "Audit Trail" palette visible by default.

To view the Audit Trail, simply click the header bar for the "Audit Trail" palette. The history of changes recorded for the object will be displayed in the right-hand pane. The content can be filtered by editing the "Filter Criteria" in "Audit Trail" palette displayed in the left-hand pane.



Assume, for example, that you receive reports that instances of the "Reset Password" Process that had been functioning properly with exception suddenly began to fail starting on the 31st of the month. Following is the Audit Trail for the Process definition:

/Employee Self-Service/Reset Password					1 to 5 out of 5 records
Date/Time	User	Action Type	Versions	Description	
03/31/10 7:55 AM	itpamadmin	Check-in with new versi...	2	The object was successfully checked in with Version:2 .	
03/31/10 7:55 AM	itpamadmin	Current version set.	2	Version:2 was successfully made the current version.	

Here you can see that the "itpamadmin" user checked in a new version of the Process definition on or about the time that the issue(s) were noted. Obviously, one approach to correcting the problem would be to determine what changes were made and, in the short term, revert to the previous version until a more stable updated version can be made available.

For more information see "Auditing User Actions" in the *CA IT PAM Administration Guide*. For details about using the CA IT PAM Client to create and debug automation objects, see the *CA IT PAM Development Guide* (for r2.1) or the *CA IT PAM User Guide* (r2.2).

Inspecting User Inputs and Other Parameters Values for a Process instance

In some cases the root cause of a Process failure may be the result of unexpected user input or other parameter values passed when an instance of a Process is initiated. This is especially true when the failures are somewhat intermittent (in other words, when not every instance launched fails).

Assume, for example, that you have been notified that an attempt to use a Process "Data Transfer – Simple" has failed. Since the failure was reported for a single instance of the Process you might reasonably suspect that value of one of the user inputs was invalid.

To view the parameter values for a specific instance of a Process, do the following:

1. Open the appropriate Process Watch (see Using Process Watch to Determine Why a Process Failed).
2. Right-click on the instance of the Process that failed then select "Open" from the in-context menu that appears.

The Process editing panel will open.

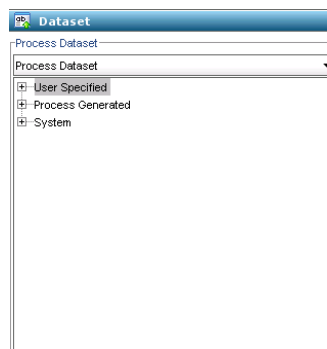
3. To inspect parameter values, click the "Dataset" tab at the bottom of the right-hand pane.



To view the parameter values for a specific *Operator* open the properties dialog box for that Operator and click the Operator Dataset button in the Execution Setting palette of Operator Properties.

Note: For details on the scope and organization of Dataset objects see the *CA IT Process Automation Manager Quick Start Guide* as well as the *CA IT PAM Development Guide* (for r2.1) or the *CA IT PAM User Guide* (r2.2) as well as the *CA IT Process Automation Management Best Practices: Recommendations for Process Design and Monitoring*.

You can navigate through the Pages of the Dataset using the controls exposed in the left-hand pane under the "Dataset" palette bar.



When the Process was designed, parameters that require user input were organized under a Page named "User Specified". When the Page is selected in the left-hand pane the corresponding parameter names and values will be displayed in the right-hand pane.

strSourceSelectQuery	...
strSourceServer	lod0272.ca.com
strSourcePort	1433
strSourceUser	sam
strSourcePassword	*****
strTargetServer	lod0272.ca.com
strTargetPort	1433
strTargetUser	sa
strTargetPassword	*****
strTargetTableName	dbContact.dbo.tblManager

In this case you might suspect that the value specified for the parameter "strSourceUser" might be incorrect.

Important: The example was purposely simplified to illustrate the concept. In reality, Process definitions should include user input validation steps and error handling for specific violations.

For details about using the CA IT PAM Client to create and debug automation objects, see the *CA IT PAM Development Guide* (for r2.1) or the *CA IT PAM User Guide* (r2.2). For more information on best practices for defining Process see the *CA IT Process Automation Management Best Practices: Recommendations for Process Design and Monitoring*.

Determining which step in a Process Instance Failed

Identifying the root cause of any application or procedure typically involves tracing through the steps executed until the action that causes the fault is located. To assist with this, CA IT PAM provides both graphic and text based tools to trace the path taken through the Operators defined to the Process for each instance of the process executed.

How do you View the Failed Step Graphically

To trace the path for a failed Process instance graphically do the following:

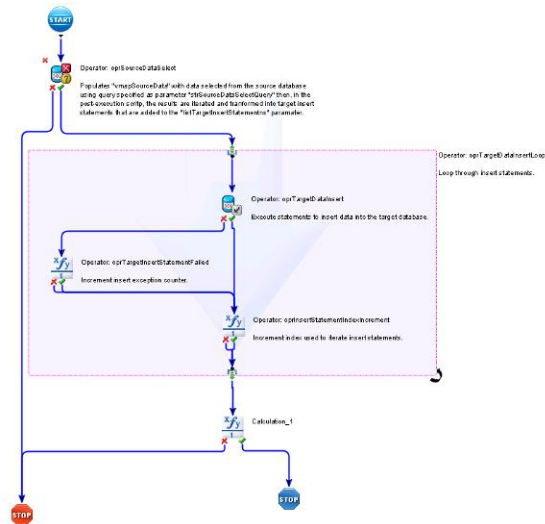
1. Open the appropriate Process Watch (see Using Process Watch to Determine Why a Process Failed).
2. Right-click on the instance of the Process that failed then select "Open" from the in-context menu that appears.

The Process editing panel will open.

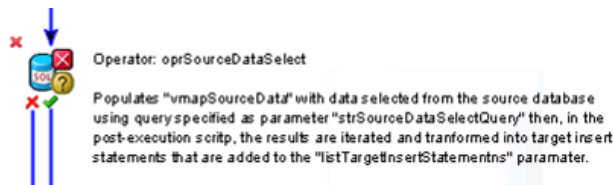
3. To follow the path taken, click the "Main Editor" tab at the bottom of the right-hand pane.



The Process definition is displayed.



Operations that failed are flagged accordingly.



Once you have identified the failed step or steps you can double-click on Operator icon to view the instance specific properties, parameter values (see Inspecting User Inputs and Other Parameters Values for a Process instance) or detailed text logs (see How do you View Text Logs for a Failed Process Instance?).

How do you View Text Logs for a Failed Process Instance?

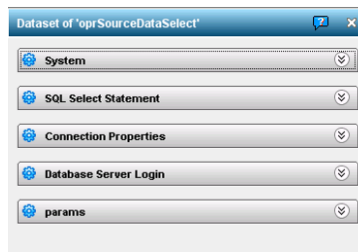
To view logs for a failed Process instance do the following:

1. Open the appropriate Process Watch (see Using Process Watch to Determine Why a Process Failed).
2. Right-click on the instance of the Process that failed then select "Open" from the in-context menu that appears.
3. Click the "Logs" tab at the bottom of the right-hand pane.

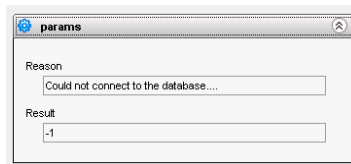
The logs for the failed Process instance will be displayed in the right-hand pane.

Time	Event Description	Category
03/30/2010 ...	'Data Transfer - Simple_76072' instance has been created	Process
03/30/2010 ...	'Data Transfer - Simple_76072' is in 'Queued' State	Process
03/30/2010 ...	Process started at '03/30/2010 08:38:04'	Process
03/30/2010 ...	'oprSourceDataSelect' is enabled following 'oprStart'	Operator
03/30/2010 ...	'oprStart' is 'Completed' on 'Current Server'	Operator
03/30/2010 ...	'oprSourceDataSelect' service request	Operator
03/30/2010 ...	'oprSourceDataSelect' is 'Running' on 'Current Server'	Operator
03/30/2010 ...	'oprSourceDataSelect' is Failed	Operator
03/30/2010 ...	Executing 'oprSourceDataSelect' post-execution ' //	Operator
03/30/2010 ...	'oprAbnormalStop' is enabled following 'oprSourceDataSelect'	Operator
03/30/2010 ...	'oprAbnormalStop' is 'Failed' on 'Current Server'	Operator
03/30/2010 ...	Process is 'Failed'	Process

In this example, note that the Operator "oprSourceDataSelect" failed. To get more details, double-click the "Failed" hyperlink to open the Operator's Dataset properties panel.



Expand the individual panels to inspect properties and parameters specific to the Operator for that instance of the Process. In the example, if you expand the "params" palette...



...you see that the reason for the Operation failures was "Could not connect to the database...". Since other instances of the Process complete successfully, you might reasonably expect that an incorrect parameter was provided for one of the properties required to connect to the database. To confirm, expand the "Database Server Login" palette.



In this example user name specified ("sam") is incorrect.

For more information see "Auditing User Actions" in the *CA IT PAM Administration Guide*. For details about using the CA IT PAM Client to create and debug automation objects, see the *CA IT PAM Development Guide* (for r2.1) or the *CA IT PAM User Guide* (r2.2).

Chapter 3: Component Issues

In the previous sections we focused on detecting and troubleshooting issues with individual Process defined by users to the CA IT PAM Library. When the issues appear to be more widespread or severe (for example, if all Processes fail or behave unexpectedly on system(s) where CA IT PAM components have been deployed) it is not unreasonable to suspect a component configuration issue as the root cause. At this point it may be necessary to contact CA support for assistance.

What External Factors Can Impact CA IT PAM Components Behavior?

As with any distributed application, changes to the network configuration or name resolution could adversely impact the ability of CA IT PAM components to function properly. User permissions may also impact the behavior. When you suspect a problem with one of the CA IT PAM components, you should follow the standard techniques for troubleshooting Agent/Server applications:

- Verify both Agent and Orchestrator services/daemons are running (changes to user accounts or permissions may prevent startup).
- Verify bi-directional name resolution (IP address changes or changes to DNS and/or "Hosts" files could impact the ability for components to communicate).
- Verify required communications ports have not been blocked by firewalls.

Refer to the *CA IT PAM Best Practices: Securability Guidelines* for detailed information on required permissions and communications ports. Additional details on administering your CA IT PAM implementation can be found in the *CA IT PAM Administration Guide*.

Determining if a CA IT PAM Component is Running

If you are experiencing a problem with CA IT PAM one of the first places to look is the Configuration Browser. Here you can quickly see if any of the components are down. For example, a red Touchpoint icon would indicate that that particular Touchpoint is down. If it is a Touchpoint group or a Touchpoint mapped to multiple agents and only part of the group is down, the icon would appear yellow. If an operator is assigned to be executed on a Touchpoint that is currently down, the processes would show "SYSTEM_ERROR" with an error message of "Unable to post message".

Determining if a CA IT PAM Component Configuration has Changed?

As with any application or system, when a component that had been functioning normally suddenly behaves unexpectedly the most likely root cause is a planned or unplanned configuration change. CA IT PAM maintains an Audit Trail for infrastructure component objects (such as Domains, Environments, Orchestrators, Touchpoints, and Agents) to assist in tracking such changes.

To view the Audit Trail for an infrastructure component do the following:



1. Open the "Configuration Browser" in the CA IT PAM Client (File -> Open Configuration Browser).
2. If not shown, add the "Audit Trail" palette to the left-hand pane. To do this:
 - a. Click the "Edit User Preferences" bar at the bottom of the left-hand pane.
 - b. On the "Config Browser" tab of the "User Preferences" dialog that appears enable "Audit Trail" in "Palette List" section.
 - c. On the "Palette Properties" tab, increase the "Number of visible palettes" to 4.

The "Audit Trail" palette should now appear in the left-hand pane.

3. In the "Browser" click on the "Domain" object.
4. Click on the "Audit Trail" palette.

Changes recorded for the selected object that meets the "Filter Criteria" specified in the "Audit Trail" palette will be displayed in the right hand pane.

Domain/					1 to 14 out of 14 records
Date/Time	Object Name	User	Action Type	Description	
03/17/10 8:24 AM	Domain	ftpadmin	Deleted	Agent 'LOD0272.ca.com' was successfully	
03/17/10 8:22 AM	LOD0272.ca.com_0	ftpadmin	Added	Agent Reference was assigned to Touchpoint	
03/17/10 8:22 AM	Default Environment	ftpadmin	Locked	Environment was locked successfully.	
03/11/10 5:33 PM	LOD0272.ca.com	ftpadmin	Unlocked	Server was unlocked successfully.	
03/11/10 5:31 PM	LOD0272.ca.com	ftpadmin	Locked	Server was locked successfully.	
03/11/10 5:31 PM	LOD0272.ca.com	ftpadmin	Unlocked	Server was unlocked successfully.	
03/11/10 5:31 PM	LOD0272.ca.com	ftpadmin	Locked	Server was locked successfully.	
03/10/10 2:40 PM	Domain	ftpadmin	Locked	Domain was locked successfully.	
03/08/10 7:57 AM	Domain	ftpadmin	Unlocked	Domain was unlocked successfully.	
03/08/10 7:57 AM	Domain	ftpadmin	Locked	Domain was locked successfully.	
03/05/10 4:54 PM	Domain	ftpadmin	Unlocked	Domain was unlocked successfully.	
03/05/10 4:54 PM	Domain	ftpadmin	Locked	Domain was locked successfully.	
03/03/10 4:20 PM	Default Environment	ftpadmin	Unlocked	Environment was unlocked successfully.	
03/03/10 3:58 PM	Default Environment	ftpadmin	Locked	Environment was locked successfully.	

For example, assume that, after troubleshooting a reported issue, you discover that all or a significant number of operations that had been working began to fail early on the morning of 3/17. A check of the Processes involved indicates they had not been modified but you have also noted that the "Execution Settings" for the failing operations have the Touchpoint "LOCAL" specified as the "Target". To determine if the Touchpoint configuration has changed do the following:

1. Open the Configuration Browser in the CA IT PAM Client.
2. Click on the "Browser" palette in the left-hand pane.
3. Navigate to and select the "LOCAL" Touchpoint object.
4. Click on the "Audit Trail" palette in the left-hand pane.
5. Adjust the "Filter Criteria" in left-hand pane under "Audit Trail" as necessary to view configuration changes that may have been implemented on or about the time the impacted Processes had started failing.

For example:



Domain/Default Environment/LOCAL					1 to 4 out of 4 records
Date/Time	Object Name	User	Action Type	Description	
03/17/10 8:24 AM	LOCAL	itpamadmin	Deleted	The agent 'LOD0272.ca.com' was removed from Touchpoint	
03/17/10 8:22 AM	LOCAL	itpamadmin	Added	The agent 'LOD0272.ca.com_0' was assigned to the Touchpoint.	
03/03/10 4:20 PM	LOCAL	itpamadmin	Added	The agent 'LOD0272.ca.com' was assigned to the Touchpoint.	
03/03/10 4:20 PM	LOCAL	itpamadmin	Added	The agent Touchpoint was created.	

Based on the entries shown it would appear that the user "itpamadmin" deleted and added an agent from/to the "LOCAL" Touchpoint and that the time and date of the configuration change appears to coincide with the reported change in behavior. While it is not possible to determine the exact reason, the search for a root cause has been narrowed to some difference between the original Agent that was deleted and the new Agent that was added. The next steps to resolution would likely include identifying why the configuration change was necessary and, if possible, reverting to the previous configuration.

Using a LogViewer Object to Identify Component Faults

CA IT PAM log messages can provide clues to the execution outcome and reasons for failure. They include hyperlinks to the parameters and results of past invocation of operations and can be helpful for debugging. LogViewer objects monitor log messages generated by Touchpoints and their modules. You can add filters to view actions executed by a specified server, on a specified Agent, and by a module. LogViewer objects also provide a dynamic or historical view. By default, a filter displays log messages as they occur but you can optionally choose to view log messages occurring within a specified date-time interval.

To define a new LogViewer object, do the following:

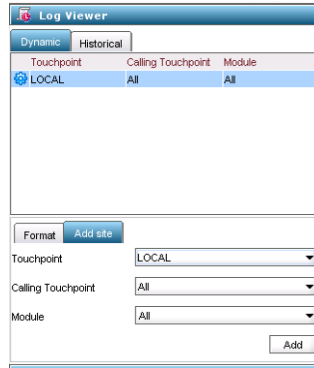
1. Open the "Library Browser" for the Library in which you intend to create the LogViewer in the CA IT PAM Client (File -> Open Library Browser).
2. Navigate to and select an appropriate folder in "Folders" palette in the left-hand pane then right-click on the folder and choose "New Object" then "LogViewer" from the in-context menus that appear.
3. Rename the new LogViewer object that appears in the right-hand pane appropriately.

To edit the default properties of a LogViewer object do the following:

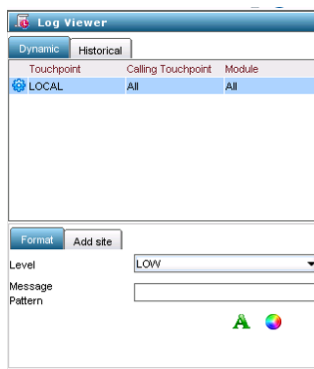
1. Right-click on the LogViewer object and select "Edit" from the in-context menu that appears.
2. Select the version of the object you intend to use as the starting point for editing from the "Versions" dialog that appears then click the "Edit" button.

You can now modify the filtering for the LogViewer using the tools in the "LogViewer" palette in the left-hand pane. To include event messages associated with specific sites select the "Add Site" tab near the bottom of the palette.





To limit the messages displayed by severity or by a message pattern select the “Format” tab.



To view messages using a LogViewer do the following:

1. Open the “Library Browser” for the Library in which you intend to create the LogViewer in the CA IT PAM Client (File -> Open Library Browser).
2. Navigate to the folder where the LogViewer is defined and double-click on the LogViewer object in the right-hand pane.

By default, the Dynamic perspective will be displayed (message relating to events that recorded at or after the time the current session is opened) with previously defined default filter properties applied. Filter properties may be modified as needed for the current session.

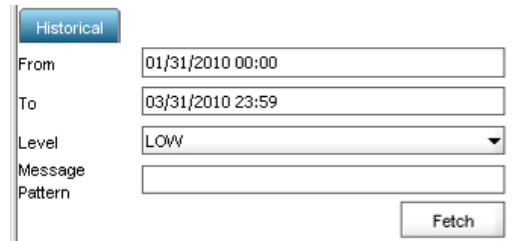
Note: To save these changes after LogViewer is closed you must check out, save and check in modifications to the LogViewer definition using the respective items on that appear on the toolbar.



The Historical retrospective can be accessed by clicking the “Historical” tab in LogViewer palette.



The ability to apply a date range filter is added when using the Historical perspective.



Adjust filters properties as required then click the "Fetch" button to retrieve the event messages matching the criteria specified.

Messages displayed in the LogViewer can be exported by clicking the "Export" icon on the toolbar.



For more information on LogViewer objects see the *CA IT PAM Development Guide* (for r2.1) or the *CA IT PAM User Guide* (r2.2).

Locating and Checking CA IT PAM Logs Directly for Component Faults

By default, CA IT PAM Orchestrator and Agent logging is configured to record messages with a severity level of warning ("WARN") and above to avoid excessive use of disk space and possible performance impact. Logs size and retention is set as follows by default:

- Orchestrators – Log files are created in the <CA IT PAM Installation Folder>\server\c2o\log directory and are limited to 50000KB (50MB) in size before rolling over; the current and two previous files will be retained.
- Agents – Log files are created in the <CA IT PAM Installation Folder>\Agent\itpamagent\log and are limited to 5000KB (5MB) in size before rolling over; the current and two previous files will be retained.

Modifying the default logging configuration is not recommended unless instructed to do so by CA support.

Modifying the Logging Configuration

Under some circumstances the default logging configuration may not capture sufficient information to identify the root cause of an issue. Typically you will be instructed to increase the level of detail (decrease the severity threshold for messages recorded). Because the number of messages recorded will increase significantly you will also need to increase the number of files retained.

Modifying the Logging Configuration for an Orchestrator

The Orchestrator logging configuration file is:

```
<CA IT PAM Orchestrator Folder>\server\conf\log4j.xml
```

Important: It is recommended that you only modify this file when directed by CA support and that you make and retain a back up copy before you edit the content.

Locate the following block in the file:

```
<!-- A size based file rolling appender for C20 and JXTA Logs-->
<appender name="C2OFILE" class="org.jboss.logging.appender.RollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="\${jboss.server.home.dir}/log/c2o.log"/>
  <param name="Threshold" value="WARN"/>
  <param name="Append" value="true"/>
  <param name="MaxFileSize" value="50000KB"/>
  <param name="MaxBackupIndex" value="3"/>

  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c] [%15.15t] %m%n"/>
  </layout>
</appender>
```

Change the "value" for the "Threshold" parameter from "WARN" to "DEBUG". Change the "value" for the "MaxBackupIndex" from "3" to "20". The update block should read as follows:

```
<!-- A size based file rolling appender for C20 and JXTA Logs-->
<appender name="C2OFILE" class="org.jboss.logging.appender.RollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="\${jboss.server.home.dir}/log/c2o.log"/>
  <param name="Threshold" value="DEBUG"/>
  <param name="Append" value="true"/>
  <param name="MaxFileSize" value="50000KB"/>
  <param name="MaxBackupIndex" value="20"/>

  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c] [%15.15t] %m%n"/>
  </layout>
</appender>
```

Important: By increasing the value for "MaxBackupIndex" to "20" there is the potential for 1GB total disk space of log files (20 files of 50MB each) to be created and retained. Verify that sufficient disk space will be available or discuss alternatives with CA support before committing the change.

Next locate the following block:

```
-->

<category name="com.optinuity.c2o.workflowengine.WorkflowManager">
    <priority value="INFO" />
</category>

<category name="com.optinuity.c2o.server.mdb.WorkflowResponseListener">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.server.mdb.RequestListener">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.archiver.ArchiverManager">
    <priority value="INFO" />
</category>

<category name="com.optinuity.c2o.workflowengine.C2oWorkFlowEngine">
    <priority value="INFO" />
</category>

<category name="com.optinuity.c2o.workflowengine.FlowManagerCache">
    <priority value="INFO" />
</category>
```

Change the values for "WorkflowManager", "C2oWorkFlowEngine" and "FlowManagerCache" from "INFO" to "DEBUG" (see highlighted items below):

```
-->

<category name="com.optinuity.c2o.workflowengine.WorkflowManager">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.server.mdb.WorkflowResponseListener">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.server.mdb.RequestListener">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.archiver.ArchiverManager">
    <priority value="INFO" />
</category>

<category name="com.optinuity.c2o.workflowengine.C2oWorkFlowEngine">
    <priority value="DEBUG" />
</category>
```



```
<category name="com.optinuity.c2o.workflowengine.FlowManagerCache">
    <priority value="DEBUG" />
</category>
```

Add the following lines following the "category" element for "FlowManagerCache":

```
<category name="com.optinuity.c2o.config.NodeManager">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject">
    <priority value="DEBUG" />
</category>

<category name="net.jxta.impl.util.pipe.reliable">
    <priority value="DEBUG" />
</category>
```

The updated section should look similar to the excerpt below:

```
-->

<category name="com.optinuity.c2o.workflowengine.WorkflowManager">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.server.mdb.WorkflowResponseListener">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.server.mdb.RequestListener">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.archiver.ArchiverManager">
    <priority value="INFO" />
</category>

<category name="com.optinuity.c2o.workflowengine.C2oWorkFlowEngine">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.workflowengine.FlowManagerCache">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.config.NodeManager">
    <priority value="DEBUG" />
</category>

<category name="com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject">
    <priority value="DEBUG" />
```

```
</category>

<category name="net.jxta.impl.util.pipe.reliable">
    <priority value="DEBUG" />
</category>
```

After completing the changes as directed, save the file. Note that because the logging configuration file is only read during Orchestrator startup, you must restart the Orchestrator service/daemon before the changes take effect.

Modifying Logging Configuration for an Agent

The Agent logging configuration file is:

```
<CA IT PAM Orchestrator Folder>\c2orepository\c2oagentresources\properties\log4j.xml
```

Important: It is recommended that you only modify the file when directed by CA support and that you make and retain a back up copy before you edit the content.

Locate the following block in the file:

```
<!-- A time/date based rolling appender -->
<appender name="FILE" class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="${user.dir}/log/c2o.log"/>
    <param name="Append" value="false"/>
    <param name="Threshold" value="INFO"/>
    <param name="MaxFileSize" value="5000KB"/>
    <param name="MaxBackupIndex" value="3"/>
<layout class="org.apache.log4j.PatternLayout">
    <!-- The default pattern: Date Priority [Category] Message\n -->
    <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>

    <!-- The full pattern: Date MS Priority [Category] (Thread:NDC) Message\n
    <param name="ConversionPattern" value="%d %-5r %-5p [%c] (%t:%x) %m%n"/>
    -->
</layout>
</appender>
```

Change the "value" for the "Threshold" parameter from "INFO" to "DEBUG". Change the "value" for the "MaxBackupIndex" from "3" to "20". The update block should read as follows:

```
<!-- A time/date based rolling appender -->
<appender name="FILE" class="org.apache.log4j.RollingFileAppender">
    <param name="File" value="${user.dir}/log/c2o.log"/>
    <param name="Append" value="false"/>
    <param name="Threshold" value="DEBUG"/>
    <param name="MaxFileSize" value="5000KB"/>
    <param name="MaxBackupIndex" value="20"/>
<layout class="org.apache.log4j.PatternLayout">
    <!-- The default pattern: Date Priority [Category] Message\n -->
    <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>

    <!-- The full pattern: Date MS Priority [Category] (Thread:NDC) Message\n
    <param name="ConversionPattern" value="%d %-5r %-5p [%c] (%t:%x) %m%n"/>
    -->
</layout>
</appender>
```



```
-->
</layout>
</appender>
```

Important: By increasing the value for "MaxBackupIndex" to "20" there is the potential for total of 100MB of log files (20 files of 5MB each) to be created and retained. Verify that sufficient disk space will be available or discuss alternatives with CA support before committing the change.

Next locate the following block:

```
-->

<!-- ===== -->
<!-- Setup the Root category -->
<!-- ===== -->
```

Insert the lines as shown below:

```
-->
<category name="org.exolab.castor.xml">
  <priority value="WARN" />
</category>

<category name="com.optinuity.c2o.config.EnvironmentManager">
  <priority value="INFO" />
</category>

<category name="com.optinuity.c2o.util.SimpleCastorXMLUtil">
  <priority value="WARN" />
</category>

<category name="net.jxta.impl.util.pipe.reliable">
  <priority value="DEBUG" />
</category>

<!-- ===== -->
<!-- Setup the Root category -->
<!-- ===== -->
```

After completing the changes as directed, save the file. Note that, because the logging configuration file is only read at startup, you must recycle the Agent service/daemon before the changes can take effect.

Troubleshooting Common Areas

Following are some quick checks that you can perform to troubleshoot common problems.

Cannot Start "CA IT PAM Orchestrator Service"

If you are unable to start the CA IT PAM Orchestrator Service, do the following:

- Check the c2o.log and boot.log files. These files are located in the \server\c2o\log folder.
- Check Java:
 - > Verify JDK 1.6 or above is installed.
 - > Check the Java path configured in the \server\c2o\bin\c2osvcw.conf file. For example:
wrapper.wrapper.java.command=C:\Program Files\Java\jdk1.6.0_15\bin\java
- Validate Hostname and DNS Lookups
- Start Orchestrator service by running \server\c2o\bin\c2osvc.bat in DOS command mode rather than starting it from "Services" console to see output directly in Window
- Utilize the \server\c2o\bin\c2osvcw.conf
 - > Change wrapper.console.loglevel= from "ERROR" to "INFO"
 - > Change wrapper.syslog.loglevel= from "ERROR" to "INFO"

Cannot Get to Administrator UI

If you are unable to access the CA IT PAMN Management Console check the following:

- If Service is Started, check the following log files:
 - > \server\c2o\log\boot.log
 - > \server\c2o\log\c2o.log
- Check if Apache is up and running. To do this, launch the following URL:

<http://<hostname>:<port>>

Note that there is no "itpam" after the URL. If the Web Server has started correctly, you should see an Apache page.

- Validate Hostname and DNS Lookups using the following commands:

```
ping <itpam_host address>
ping <itpam_host name>
nslookup <itpam_host name>
```



Cannot Login to Administrator UI

If you are unable to login to the CA IT PAM Management Console, try the following:

- If a Login page appears:
 - > Verify Authentication type (EEM / AD / LDAP)
 - > Validate that you can login to any of those authentication provider applications directly using the same user ID and password
 - > If you are using CA EEM confirm that you can login under the application "ITPAM" or any other application name specified during the installation
 - > If you are using Active Directory, try to login using the domain name
 - > Check the logs to see if there is any lack of communication to the authentication servers
- For CA EEM:
 - > Review *CA IT PAM Release Notes* guide for information on how to create the certificate using safex.
 - > ITPAM_EEM.xml can be modified for Application Name and Password if needed
 - > Verify direct login to CA EEM with user if needed
 - > CA IT PAM objects in EEM have changed between r2.1 and r2.2. Review the *Release Notes* for more information.

Cannot Launch IT PAM Client

If you are unable to launch the CA IT PAM Client (JNLP), check the following:

- If you get an "Authentication Error" see TEC500532
- Once you have logged in, verify if the JNLP has started to load. If it has, check the error message that appears under the details. Typical problems include hostname and lookups. See TEC500533 for information on the Configure option.
- Verify Correct / Supported Version of Java
- Check the logs for signs that there is any lack of communication to the authentication servers
- Download the JNLP locally and view the file to verify hostname and port are valid (Check FQDN vs Host) .JGo.jar is First Jar that gets downloaded

Cannot "See" Agents in the Configuration Browser

If you cannot see installed and running CA IT PAM Agents from the Configuration Browser, check the following:

- > Is the agent able to communicate to the Server (ports opened in case of firewalls etc)?
- > Within the ITPAM Client:
 - > Does the Orchestrator have the Agent set to Inactive?
 - > Do you have access rights to view the Agents?
 - > Are you looking for Agents via Touchpoint before they are added to it?
 - > Is the client using INET Standard Hostnames ?

On agent folder, check the **./config/Oasisconfig.properties**, change value of **oasis.jxta.host** to correct host name and then delete the **.system** folder

You may need to reinstall the agent. If so, keep in mind that Agents do not have versions to them; they download the latest Domain.xml from the Orchestrator.

Cannot Start the Windows Agent

If you cannot start the CA IT PAM Windows Agent, check the following:

- Is the agent installed?
- Is there a JRE on the server ? If so, is it a supported version?
- Is the Java path within the config file valid? Check the \ITPAM\itpamagent\c2osvcw.conf file. For example:

```
wrapper.java.command=C:\Program Files\Java\jdk1.6.0_15\bin\java
```

- Is the Hostname to the Orchestrator valid in the c2osvcw.conf file?
- Enable Debugging in c2osvcw.conf by replacing the following:
 - Change *wrapper.console.loglevel=* from "ERROR" to "INFO"
 - Change *wrapper.syslog.loglevel=* from "ERROR" to "INFO"
- Run c2oagtsvc Agent service in DOS command window **instead** of starting it from Services MMC

Cannot Start *NIX Agents

If you cannot start the CA IT PAM Agent on Unix or Linux, check the following:

- Is the agent installed?



- Is there a JRE on the server ? If so, is it a supported version?
- Is the Java path within the config file valid in the \usr\local\ITPAMAgent\c2oagtd.sh file? Verify 'JAVA_HOME' or it can be commented out
- Is the Hostname of the Orchestrator valid in that file?
- Enable Debugging in c2osagt.d daemon file by REMOVING this part of the "nohup" (stands for no hang up):

```
>/dev/null 2>/dev/null
```

```
leave echo $! > c2oagt.pid
```

Start the **c2oagtd.sh** and wait a few seconds. The will output to file: nohup.out

Agent Not Communicating with Orchestrator

If the Agent (either Win32 or Unix/Linux) is not communicating with the Orchestrator, check the following:

- Confirm the Orchestrator hostname value is valid in the following files:
 - > On Windows: c2oagtsvcw.conf
 - > On Unix/Linux: c2oagtd.sh
- Check in the Orchestrator LOG file that it received a "heartbeat message" from that Agent. For example:

```
[com.optinuity.c2o.c2oserver.ServerManager] Received heartbeat message from
CMDBVM01.forward.inc - 09a772f1-f842-4c50-a378-f6d89dac6dd9 on 192.168.21.31
```

- Does the Orchestrator Hostname resolve correctly?
- Can Orchestrator Hostname / IP be pinged?
- Is the Orchestrator Service Started and Running?

Database Errors

To avoid database related errors, keep the following in mind:

For MySQL database:

- The following startup variables must be set in MySQL before installing an Orchestrator that uses the MySQL database server:
 - > Max. packet size >=16 M
 - > Lock Wait Timeout >=60 seconds

- Note that databases required by CA IT PAM can be created during the Orchestrator installation

For Microsoft SQL Server:

- It is recommended to install MS SQL Server with **mixed authentication mode** and specify an account with SQL Server authentication during the Orchestrator installation

For Oracle:

- Databases must be created first (manually) to host the following:
 - > General Library (recommended at 1.5GB+)
 - > Queues (recommended at 150MB+)
 - > Reporting (recommended at 1GB+) Note that one Reporting database can be shared by multiple Orchestrators
- Ensure Connection pool (max connections = 10) and other settings are OK

Database information is located in **oasisconfig.properties** file.

Chapter 4: Before You Call Support

If you are unable to identify the root cause and/or could not determine how to correct the Process or component configurations to restore normal function you may need to contact CA support for assistance. To facilitate the resolution of any issue, collect as much of the following information as you can:

- Determine when the problem first occurred
- Determine what changes were made to the CA IT PAM configuration or the environment in general on or about the time the problem first occurred.
- Determine if the problem can be reproduced; if so, document the steps to reproduce the problem.
- Gather the relevant Orchestrator and Agent logs for the time period of the problem:
 - > All log files in the <CA IT PAM Installation Folder>\server\c2o\log directory from each affected Orchestrator.
 - > All log files in the <CA IT PAM Installation Folder>\Agent\itpamagent\log directory.
- Export one or more of the Processes that can be used to reproduce the behavior to a CA IT PAM XML export file (refer to the *CA IT Process Automation Manager User Guide* for more information on exporting objects).

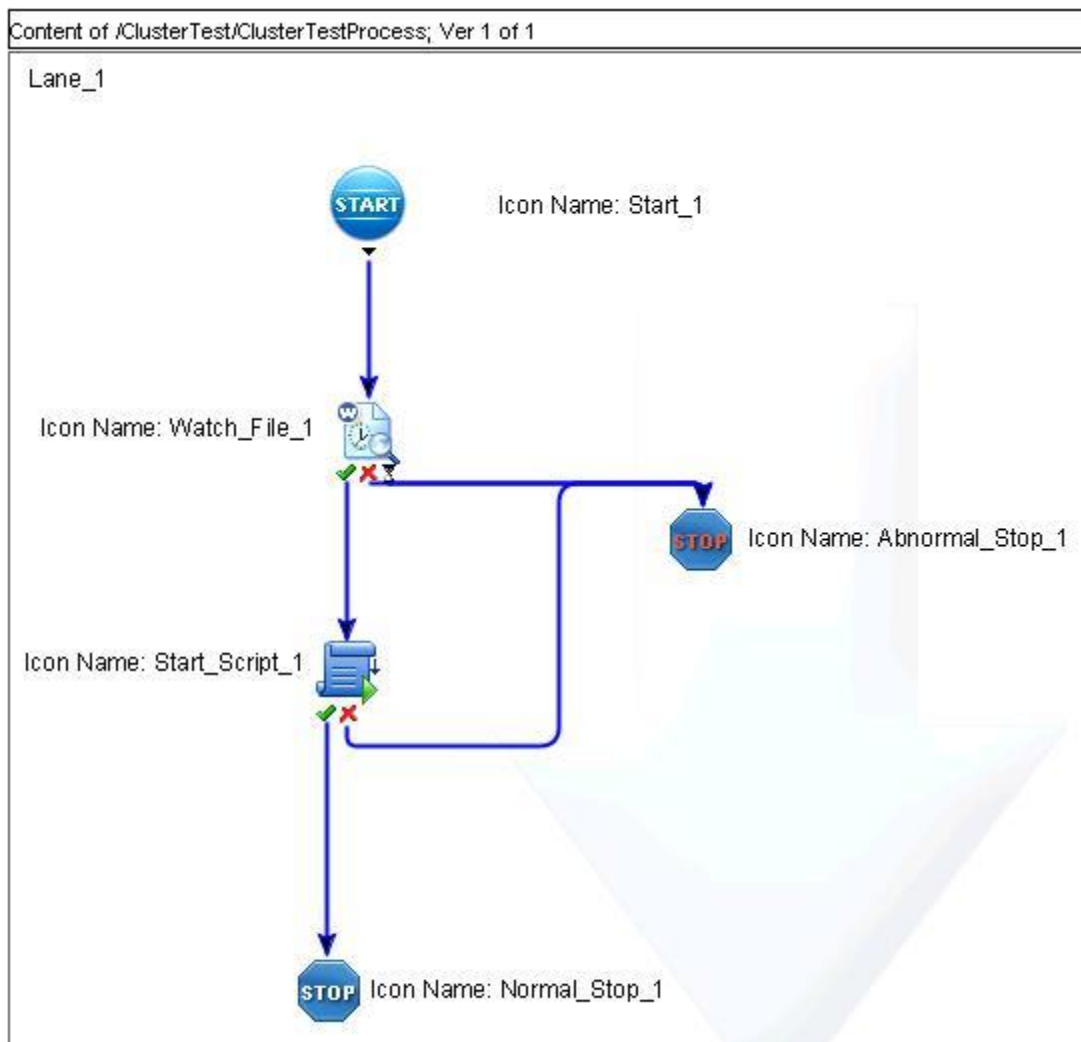
CA support representatives will provide detailed instructions on uploading the necessary files.



Appendix A: Tracing Process Operations through Orchestrator and Agent Logs

In most cases, the techniques described in the previous sections should be sufficient for troubleshooting CA IT PAM issues. The following example is provided for the advanced user interested in tracing an operation of a Process as it transitions from one internal CA IT PAM sub-component to another. Typically, this type of tracing is performed by CA support engineers when a component level defect is suspected - end users are not expected to perform such analysis.

Consider the following sample process:



Open <CA IT PAM Installation Folder>\server\c2o\log\c2o.log. The first entry we are interested in is the one that returns the Runtime Object ID (ROID) specific to the Process being instantiated (in other words, "ClusterTestProcess"):

```
2010-03-03 15:25:02,015 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager] [
TP-Processor11] [ITPAM Core Performance] WorkflowManager: Time taken in
createRunTimeInstance = 15ms.; for Doc: ms.; for Doc:
/ClusterTest//ClusterTestProcess; for ROID: 42
```

The ROID for process `ClusterTestProcess` in folder `ClusterTest` is "42". This ROID will be used throughout this example.

Note: Because the log level for the C2OFILE appender log is set to DEBUG, there will be many other events logged. If the log level had been set otherwise, such as to WARN, all DEBUG events would have been filtered out.

Now that we know what the process ROID is, let's find its UUID:

```
2010-03-03 15:25:02,062 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager] [
TP-Processor11] Special Response to dequeue flow. ROID is 42 UUID is b4b8a461-19fc-
4383-bbc3-fbca12defccc
```

As we can see in the sample process diagram, the workflow implements a single entry point defined by the Start Operator `Start_1`. Let's find its ROID:

```
2010-03-03 15:25:02,109 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager] [
onPool Worker-1] Going to Execute the next enabled icon. ROID is 43 Instance is
Start_1
```

```
2010-03-03 15:25:02,109 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Execute begins instance Start_1 ROID 43
```

...

```
2010-03-03 15:25:02,109 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Completed Operator ROID is 43 Name StartCommonServicesOperation
```

Now that the Start Operator `Start_1` completed its execution, we need to track the next one, in other words, `Watch_File_1`:

```
2010-03-03 15:25:02,109 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager] [
onPool Worker-1] Looking for next set Source is Start_1 ROID is 43
```

So, since the Source for the next set was `Start_1`, we keep an eye on ROID 43 until we find the next one, in other words, ROID 44, and confirm that it is, in fact, the correct ROID for the `Watch_File_1` Operator:

Last Updated: April 21, 2010

```
2010-03-03 15:25:02,109 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Retrieving Runtime instance for ROID 44
```

```
2010-03-03 15:25:02,109 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Retrieved Runtime instance for ROID 44
```

...

```
2010-03-03 15:25:02,109 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Next Destination is Watch_File_1 ROID is 44
```

...

```
2010-03-03 15:25:02,109 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Going to Execute the next enabled icon. ROID is 44 Instance is
Watch_File_1
```

```
2010-03-03 15:25:02,109 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Execute begins instance Watch_File_1 ROID 44
```

...

```
2010-03-03 15:25:02,218 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Queueing outbound request : e7d248d8-ad6b-4749-8313-bc8acce81f98
```

Notice that the Watch File Operator `Watch_File_1` runs on a Touchpoint with CA IT PAM Agent(s) mapped to it. So, once the `Watch_File_1` execution begins, an outbound request is generated and sent to the CA IT PAM Agent(s) mapped to that Touchpoint. Therefore, it is really important to keep track of that outbound request UUID because that is what must be tracked on the Agent side to make sure that the request has been received.

Here we see that the CA IT PAM Orchestrator did send the outbound request:

```
2010-03-03 15:25:04,781 DEBUG [com.optinuity.c2o.server.mdb.RequestListener] [onPool
Worker-2] [ITPAM Core Performance] RequestListener: Time taken in deliverRequest =
1110ms.; for UUID: e7d248d8-ad6b-4749-8313-bc8acce81f98
```

In this example, the CA IT PAM Agent(s) responded to outbound request:

```
2010-03-03 15:26:01,296 INFO [com.optinuity.c2o.config.EnvironmentManager] [
PM: 1] Received service response 1 for request: e7d248d8-ad6b-4749-8313-bc8acce81f98
- queue queue/ResponseQueue
```

...

```
2010-03-03 15:26:01,328 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Calling Icon to handle Response, as Response seem OK. Instance is
Watch_File_1; ROID is 44
```



Once the `Watch_File_1` Operator criterion is satisfied, we track the next one. i.e., `Start_Script_1`:

```
2010-03-03 15:26:01,359 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Looking for next set Source is Watch_File_1 ROID is 44
```

Since the Source for the next set is `Watch_File_1`, we keep an eye on ROID 44 until we find the next one, in other words ROID 46, and confirm that it is, in fact, the correct ROID for the `Start_Script_1` Operator:

```
2010-03-03 15:26:01,359 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Next Destination is Start_Script_1 ROID is 46
```

...

```
2010-03-03 15:26:01,359 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Going to Execute the next enabled icon. ROID is 46 Instance is
Start_Script_1
```

```
2010-03-03 15:26:01,359 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Execute begins instance Start_Script_1 ROID 46
```

...

```
2010-03-03 15:26:01,375 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Queueing outbound request : 19f19453-5eee-4e39-8155-85b8bd25c3ad
```

Notice that the Watch File Operator `Start_Script_1` also runs on a Touchpoint with CA IT PAM Agent(s) mapped to it. So, once the `Start_Script_1` execution begins, an outbound request is generated and sent to the CA IT PAM Agent(s) mapped to that Touchpoint. As a result, it is really important to keep track of that outbound request UUID as well because that is what must be tracked on the Agent side to make sure that the request has been received.

Here we see that the CA IT PAM Orchestrator did send the outbound request:

```
2010-03-03 15:26:01,484 DEBUG [com.optinuity.c2o.server.mdb.RequestListener] [onPool
Worker-2] [ITPAM Core Performance] RequestListener: Time taken in deliverRequest =
31ms.; for UUID: 19f19453-5eee-4e39-8155-85b8bd25c3ad
```

In this example, the CA IT PAM Agent(s) responded to outbound request:

```
2010-03-03 15:26:01,562 INFO [com.optinuity.c2o.config.EnvironmentManager] [
PM: 2] Received service response 2 for request: 19f19453-5eee-4e39-8155-85b8bd25c3ad
- queue queue/ResponseQueue
```

...

Last Updated: April 21, 2010

```
2010-03-03 15:26:01,593 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Calling Icon to handle Response, as Response seem OK. Instance is
Start_Script_1; ROID is 46
```

Once the Start_Script_1 Operator is finished, we track the next one. i.e., Normal_Stop_1 (assuming that Start_Script_1 completed successfully, of course):

```
2010-03-03 15:26:01,640 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Looking for next set Source is Start_Script_1 ROID is 46
```

So, since the Source for the next set is Start_Script_1, we keep an eye on ROID 46 until we find the next one, in other words, ROID 47, and confirm that it is, in fact, the correct ROID for the Normal_Stop_1 Operator:

```
2010-03-03 15:26:01,640 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Next Destination is Normal_Stop_1 ROID is 47
```

...

```
2010-03-03 15:26:01,640 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Going to Execute the next enabled icon. ROID is 47 Instance is
Normal_Stop_1
```

```
2010-03-03 15:26:01,640 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Execute begins instance Normal_Stop_1 ROID 47
```

Since the Normal Stop Operator Normal_Stop_1 runs on the CA IT PAM Orchestrator, no outbound request is generated and sent to the CA IT PAM Agent(s) and we see the Normal Stop Operator completing:

```
2010-03-03 15:26:01,687 DEBUG
[com.optinuity.c2o.workflowengine.C2OSvcIconInstanceRefObject] [onPool Worker-1]
Completed Operator ROID is 47 Name NormalStopCommonServicesOperation
```

The CA IT PAM Orchestrator attempts to look for the next enabled Icon/Operator after the Normal Stop Operator

```
2010-03-03 15:26:01,687 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager] [onPool
Worker-1] Looking for next set Source is Normal_Stop_1 ROID is 47
```

However, as we can see, there is none:

```
2010-03-03 15:26:01,687 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] [ITPAM Core Performance] setNextEnabledIcons: Time taken in
processing (nothing) = 0ms.; for ROID: 47
```

```
2010-03-03 15:26:01,687 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager]
[onPool Worker-1] Icon is Normal_Stop_1 ROID is 47; Completed and removed from state
running Icon List
```



So, once the Normal Stop Operator is removed from the Running state Icon/Operator List, the CA IT PAM Orchestrator requests the completed flow instance, in other words, ROID 42, to be rolled up:

```
2010-03-03 15:26:11,687 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3] Request to rollup completed flow instance - UUID is d2c29516-bd3f-453c-999f-932b2729800d
```

Once the roll up request is completed, the process ("ClusterTestProcess"), is also completed:

```
2010-03-03 15:26:11,687 DEBUG [com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3] Rolling up svc icon runtime states for completed flow 4
```

Appendix B: Configuration and Log Files

This appendix contains general information regarding several of the CA IT PAM configuration and log files. For additional details, consult the *CA IT PAM Administration Guide*.

Orchestrator Configuration Files

There are four main configuration files for Orchestrators:

- Domain.xml
- OasisConfig.properties
- c2osvcw.conf (Windows) or c2osvcd.sh (Unix/Linux)
- Log4j.xml

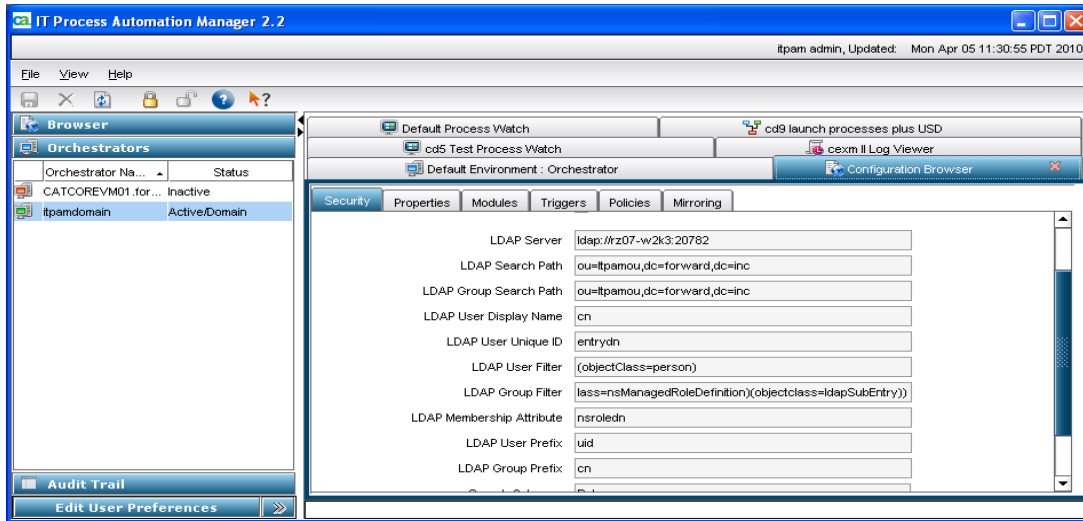
Orchestrator's Domain.xml

The domain.xml file stores system configuration information and is located in the following path:

```
<install path>/server/c2o/.config/
```

The master copy is maintained by the Domain Orchestrator and other Orchestrators receive a copy of Domain.xml from the Domain Orchestrator (via synchronization process). Agents receive partial copies of Domain.xml from the Domain Orchestrator.

The configuration contained in Domain.xml is set using the Configuration Browser in the ITPAM Java Client.



Do not modify the Domain.xml directly!

Orchestrator OasisConfig.properties

The OasisConfig.properties file contains default JBoss setting and other settings that are specified during the install. Some of these settings are pushed to the configuration files for other components used by CA IT PAM at startup. For example, JBoss settings are copied from OasisConfig.properties into the JBoss configuration files.

The OasisConfig.properties files is located in the following path:

```
<install path>/server/c2o/.config/
```

There are a number of settings (typically for ports) in OasisConfig.properties that are set to defaults and are not configured as part of the installer. If you are having issues bringing up an Orchestrator due to port conflicts, you may have to manually modify ports in this file.

Orchestrator c2osvcw.conf and c2osvcd.sh

The c2osvcw.conf file (Windows) and c2osvcd.sh file (Unix/Linux) are located in the following path:

```
<install path>/server/c2o/bin/
```

The c2osvcw.conf file contains JVM and NT Service settings. Some of the settings you may want to manually add or modify include the following:

- Initial Java Heap size (wrapper.java.initmemory=128)
- Maximum Java Heap size (wrapper.java.maxmemory=1024)
- Paths to SSL certificates needed for HTTPS access to an application or system

- `Wrapper.ping.timeout` which specifies the time to wait for a **ping** response from a JVM before considering that JVM is dead and restarting it (`wrapper.ping.timeout= 60`)

- Log file settings: level, size, number of rolled log files. These include:

`wrapper.logfile.loglevel=INFO`

`wrapper.logfile.maxsize=5m`

`wrapper.logfile.maxfiles=5`

See the Performance Tuning section in the *CA IT PAM Best Practices Scalability Guidelines* document for additional details.

Orchestrator log4j.xml

The `log4j.xml` file specifies the logging levels for the Orchestrator and is located in the following path:

```
<install path>/server/c2o/conf/
```

See Logging section earlier in this document for more details.

Orchestrator Log Files

There are four primary log files for Orchestrators:

- **installation.log** – contains the logged messages related to the installation of the CA IT PAM Orchestrator.
- **boot.log** – contains all messages logged to the core components (JBoss, etc.) before the IT PAM process starts executing.
- **c2o.log** – contains all messages logged by all components after the IT PAM process starts executing.
- **c2ow.log** – contains log messages for the NT service wrapper.

The `installation.log` file is located in the following path:

```
<install path>/server/c2o/.install4j/
```

The `boot.log`, `c2o.log` and `c2ow.log` files are located in the following path:

```
<install path>/server/c2o/log/
```

The Orchestrator issues are typically detected in the `c2o.log` and `boot.log` files.

Agent Configuration Files

There are three main configuration files for Agents. They are:

- Domain.xml
- OasisConfig.properties
- C2oagtsvcw.conf (Windows) or c2oagtd.sh (Unix/Linux)
- Log4j.xml

Agent Domain.xml

The Agent Domain.xml stores systems configuration information for the Agent. It is located in the following path:

```
<install path>\itpamagent\.config
```

Agents receive partial copies of the "master" Domain.xml from the Domain Orchestrator. As with the Domain Orchestrator's Domain.xml file – **do not** modify the Agent's Domain.xml file directory. Use the CA IT PAM Client!

Agent OasisConfig.properties

The Agent OasisConfig.properties file contains the JXTA port setting configuration details. It is located in the following path:

```
<install path>\itpamagent\.config
```

Agent c2oagtsvcw.conf and c2oagtd.sh

The Agent service configuration file – c2oagtsvcw.conf for Windows and c2oagtd.sh for Unix/Linux – is located in the following path:

```
<install path>\itpamagent
```

The c2osvcw.conf file contains JVM and NT Service settings. Typical settings that you may want to manually add or modify include:

- Initial Java Heap size (wrapper.java.initmemory=64)
- Maximum Java Heap size (wrapper.java.maxmemory=256)
- Paths to SSL certificates needed for HTTPS access to an application or system
- Time to wait for a ping response from the JVM before considering that the JVM is dead and restarting it (wrapper.ping.timeout=60)
- Wrapper Logging properties

Log4j.xml

The log4j.xml logger configuration file is located in the following path:

```
<install path> \itpamagent\.c2orepository\.c2oagentresources\properties
```

This file specifies the logging levels for the Agent. See the earlier Logging Configuration discussion for more details.

Agent Log Files

There are three main log files for Agents:

- **installation.log** – contains logged messages related to the installation of the ITPAM Agent
- **c2o.log** – the main log file, it contains all messages logged by Agent component
- **c2ow.log** – contains log message for the NT service wrapper

The installation.log file is located in the following path:

```
<install path>/itpamagent/.install4j/
```

The c2o.log and c2ow.log files are located in the following path:

```
<install path> >/itpamagent /log/
```

Agent issues are typically monitored via **c2o.log** file