

CA BEST PRACTICES

CA IT Process Automation Manager Best Practices

Securability 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 © 2009 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 – Securability Guidelines

Publication Date: April 2010

Last Update: April 26, 2010

ACKNOWLEDGEMENTS

Principal Authors and Technical Editors

Terry Pisauro
Alex Moscoso

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

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: Security Considerations** **9**
 - Access Privileges Required by IT PAM Services 9
 - Communications and Firewalls 10
 - Port Numbers 11
 - Firewall Considerations 14
 - Resolving Port Conflicts 18
 - Using CA EEM to Manage Access to CA IT PAM..... 22
 - Securing and Isolating Environments and Objects 26
 - Securing the JBoss Server..... 31



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 “securability” – which refers to the ability of a product (CA IT PAM, in this case) to protect information, services, and related systems from unauthorized access, use, disclosure, disruption, modification, and destruction. This document includes recommendations for implementing CA IT PAM in a secure environment – such as port settings, firewall considerations, user ID and privilege requirements for OS Lockdown as well as securing access to CA IT PAM through CA Embedded Entitlements Manager (CA EEM).

Last Updated: April 26, 2010



Chapter 2: Security Considerations

This chapter focuses on issues regarding the securability of CA IT PAM. This includes:

- Access Privileges Required by IT PAM Services
- Communications and Firewalls
- Securing and Isolating Environments and Objects
- Using CA EEM to manage access to CA IT PAM

Access Privileges Required by IT PAM Services

Depending on the components you install, CA IT PAM runs the following services.

- IT PAM Orchestrator
- IT PAM Agent

On all supported Windows operating systems, CA IT PAM Orchestrators and Agents may optionally run as Microsoft Windows services or in a Command Shell (`cmd.exe`). If you choose to run CA IT PAM as a Windows Service, these services appear on the list of installed services in the Services console.

CA IT PAM Orchestrator and Agent services should be run under the Local System account. This makes it easier to deploy CA IT PAM by avoiding the complexities associated with handling of user IDs and passwords. The Local System account also has the proper permissions to run operations on the host, which include:

- Act as part of the operating system (`SeTcbPrivilege`)
- Create a token object (`SeCreateTokenPrivilege`)
- Logon as a Service (`SeServiceLogonRight`)
- Logon as a batch job (`SeBatchLogonRight`)
- Replace process level tokens (`SeAssignPrimaryTokenPrivilege`)

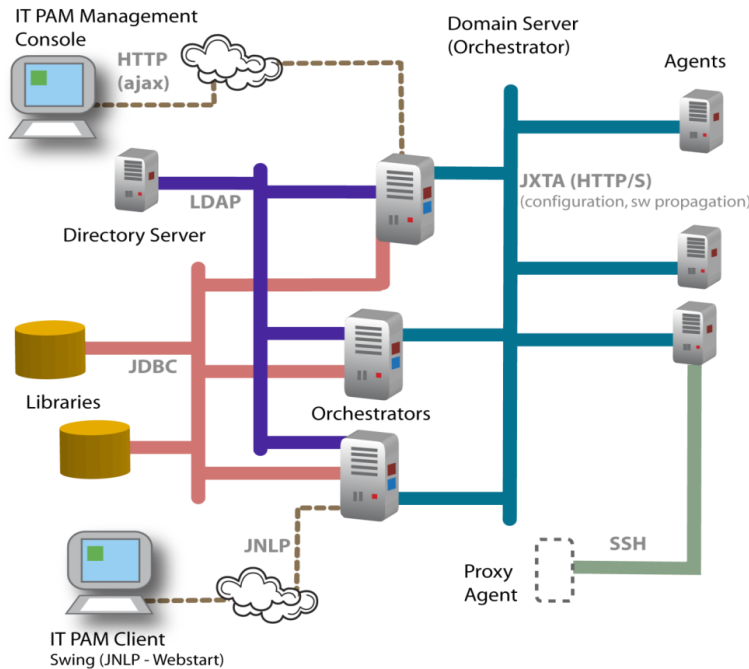
A "Proxy Touchpoint" is a Touchpoint associated with an Agent that is configured to connect to and execute actions on other remote hosts using SSH. Proxy Touchpoints are configured to an existing Touchpoint.

The prerequisites to configure Proxy Touchpoints are as follows:

- A UNIX or Windows Agent is running on a host computer and is configured as a Touchpoint.
- The remote host must have an SSH server running and the host computer of the Touchpoint must be able to run the SSH client to connect to the SSH server running on the remote host.
- When the target for a proxy Touchpoint is a UNIX computer, the Bourne shell must be installed on the target computer. If it is not, either install the Bourne shell or link it from the Bash shell.
- The login account for a Proxy Touchpoint must have write access on the remote computer.
- Before you install a Proxy Touchpoint, you must create a trust relationship for the target computer and test SSH connectivity from the computer running Proxy Touchpoint to the target computer.

Communications and Firewalls

CA IT PAM Orchestrators and Agents communicate using JXTA, an open source peer-to-peer communication protocol, defined as a set of XML messages, that allow any device connected to a network to collaborate independently of the underlying network topology. This communication is organized into JXTA Communication Groups which map to various system elements in CA IT PAM:



Each CA IT PAM Domain represents a JXTA communication group, which is identified by a UUID that is automatically generated in the `domain.xml` file (i.e., `DomainID`). Each Environment that is added to the Domain is also defined as a JXTA Communication group. A new UUID is generated in the `domain.xml` file (i.e., `EnvironmentID`) to identify a new Environment when it is added.

Orchestrators and Agents communicate bi-directionally to perform process operations. The Domain Orchestrator will also perform mirroring, heartbeat and configuration updates with the Agents and with other Orchestrators. Orchestrators communicate bi-directionally with each other to perform process operations.

The CA IT PAM Client and Management Consoles initiate communication with Orchestrators to perform configuration, to create and modify automation objects, to start and monitor processes and to respond to interaction requests. Although both communicate with an Orchestrator through its HTTP port, the Management Console uses only HTTP (Ajax – JavaScript and HTTP) while the Client also uses JNLP (which specifies how to launch the Java Web Start application) to initiate a connection and update its files. Multiple clients and consoles can be initiated from the same desktop systems and can point to the same or different Orchestrators – even in different domains.

For Proxy Agents or Touchpoints, a particular Agent is configured to communicate with the remote host through SSH (v2). The remote host must be running an SSH Server and, for some operations, must have a write-accessible file system.

Communication between Agents on UNIX or Windows boxes is based on TCP/IP. Data transmitted between CA IT PAM components is encrypted.

Port Numbers

The following table details the port numbers that are used by CA IT PAM components:

Port #	CA IT PAM Component	Configurable at installation?	Notes
7003*	Agent	Yes	<ul style="list-style-type: none"> • TCP port • JXTA communications • Bi-directional
22	Proxy Agent	Yes (at SSH installation)	<ul style="list-style-type: none"> • TCP port • Privileged port • Outbound SSH connection required by CA IT PAM Proxy Agents • Bi-directional <p>Outbound SSH connections to the SSH host required for its operations.</p>
7001*	Orchestrator	Yes	<ul style="list-style-type: none"> • TCP port • JXTA communications • Bi-directional • OasisConfig.properties parameter: <code>oasis.jxta.port</code>
162*	Orchestrator	Yes	<ul style="list-style-type: none"> • UDP port • Incoming SNMP traps • OasisConfig.properties parameter: <code>oasis.snmptrigger.service.port</code>
1098*	Orchestrator	Yes	<ul style="list-style-type: none"> • TCP port • Java Remote Method Invocation (RMI) • Bi-directional • OasisConfig.properties parameter: <code>jboss.rmi.port</code> <p>Used for discovering information on RMI services.</p>



1099*	Orchestrator	Yes	<ul style="list-style-type: none"> • TCP port • Java Naming and Directory Interface (JNDI) • Bi-directional • <code>OasisConfig.properties</code> parameter: <code>jboss.jndi.port</code> <p>Used for looking up port and other information for application services offered by the Orchestrator. Provides a common remote interface to various services (e.g., LDAP, NDS, DNS, and NIS).</p>
8083*	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • RMI Web Service port • Bi-directional • <code>OasisConfig.properties</code> parameter: <code>jboss.rmi.classloader.webservice.port</code> <p>Used for handling web service calls to load various RMI objects (classes) from various nodes of the Orchestrator within a cluster.</p>
4444*	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • RMI Server port • Bi-directional • <code>OasisConfig.properties</code> parameter: <code>jboss.rmi.object.port</code> <p>RMI server socket listening port. This is the port RMI clients connect to when communicating through a single server to handle incoming RMI client calls.</p>
4446*	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • Pooled Invoker port • Pools connections from client to server • Bi-directional • <code>OasisConfig.properties</code> parameter: <code>jboss.pooledinvoker.serverbind.port</code> <p>Pooled invoker server bind port. Used by the Orchestrator as a multiplexer for custom socket connections by using standard RMI service implemented through the MBean interface. Pooled invoker service pools client socket connections to the server unlike standard JRMP interface that creates a socket connection for each request.</p>
1100*	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • High Availability Java Naming and Directory Interface (HA-JNDI) • Bi-directional • For High Availability (clustering) • <code>OasisConfig.properties</code> parameter: <code>jboss.ha.jndi.port</code> <p>Port on which the HA-JNDI stub is made available. Used for looking up port and other information for Application Services offered by the Orchestrator. Provides a common remote interface to various services (e.g., LDAP, NDS, DNS, and NIS).</p>
1101	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • Highly Available Java Remote Method Invocation (HA-RMI) • For High Availability (clustering) • Bi-directional • <code>OasisConfig.properties</code> parameter: <code>jboss.ha.jndi.rmi.port</code> <p>RMI port to be used by the HA-JNDI service once bound. Used for discovering information on RMI services, when the Orchestrator is clustered.</p>



4447*	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • HA-RMI Server port • For High Availability (clustering) • Bi-directional • OasisConfig.properties parameter: jboss.ha.rmi.object.port <p>RMI object port used by JRMPInvokerHA class. When the Orchestrator is clustered, this port is used by RMI clients when communicating through a single server to handle incoming calls from RMI clients.</p>
4445*	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • Highly Available Pooled Invoker port • Bi-directional • Pools connections from client to server • For High Availability (clustering) • OasisConfig.properties parameter: jboss.ha.pooledinvoker.serverbind.port <p>Pooled invoker HA server bind port. When clustered, used by the Orchestrator as a multiplexer for custom socket connections, via standard RMI service implemented by the MBean interface.</p>
1102*	Orchestrator	No	<ul style="list-style-type: none"> • UDP port • JNDI Autodiscovery Service • Multicast group port • For High Availability (clustering) • Bi-directional • OasisConfig.properties parameter: jboss.mcast.jndi.autodiscovery.port <p>Multicast group port used to auto-discover other JNDI ports within an IT PAM Orchestrator cluster. Used when the Orchestrator is clustered.</p>
8080*	Orchestrator	Yes	<ul style="list-style-type: none"> • TCP port • Incoming Web Services port • Bi-directional • OasisConfig.properties parameter: tomcat.connector.http.port <p>Axis web service port. Used to receive incoming web service calls from external clients\application services.</p>
8443*	Orchestrator	Yes	<ul style="list-style-type: none"> • TCP port • Incoming Web Services port • Secure port (SSL) • Bi-directional • OasisConfig.properties parameter: tomcat.secure.port <p>Port for Connector component that supports the HTTP/1.1 protocol. It enables Catalina to function as a stand-alone web server, in addition to its ability to execute servlets and JSP pages. Used for receiving incoming web service calls from external clients or application services. Data is encrypted using SSL.</p>
8009*	Orchestrator	Yes	<ul style="list-style-type: none"> • TCP port • AJP port • Orchestrator cluster • Bi-directional • OasisConfig.properties parameter: tomcat.connector.ajp.port • worker.properties parameter (Load Balancer): worker.nodename.port <p>Used for managing sessions with an external load balancer. Uses Apache JServ protocol (AJP).</p>



8093*	Orchestrator	No	<ul style="list-style-type: none"> • TCP port • Unified Invocation Layer (UIL) Service port • Bi-directional • OasisConfig.properties parameter: <code>jboss.uil.serverbind.port</code> <p>Port used by UIL service clients to establish a connection to JBoss Message Queues (JBossMQ) server</p>
80	Apache HTTP Server/Load Balancer	Yes (8080 when manually started – but not recommended since 8080 is by Apache Axis)	<ul style="list-style-type: none"> • TCP port • Bi-directional • Apache HTTP Server/Apache Tomcat Connector (AJP protocol) <p>Used to access IT PAM (Management Console and Client) when Orchestrators are clustered.</p>
3303*	Database (MySQL)	Yes (in MySQL)	<ul style="list-style-type: none"> • TCP port • Bi-directional • OasisConfig.properties parameter: <code>oasis.database.dbport</code>
1433*	Database (MS-SQL)	Yes (in MS-SQL)	<ul style="list-style-type: none"> • TCP port • Bi-directional • OasisConfig.properties parameter: <code>oasis.database.dbportt.</code>
1521*	Database (Oracle)	Yes (in Oracle)	<ul style="list-style-type: none"> • TCP port • Bi-directional • OasisConfig.properties parameter: <code>oasis.database.dbport.</code>
389*	LDAP – EEM	--	<ul style="list-style-type: none"> • TCP port • Bi-directional <p>Standard LDAP protocol port used by EEM client libraries to connect to EEM for IT PAM security data. See CA EEM documentation for details.</p>
636*	LDAP – EEM	--	<ul style="list-style-type: none"> • TCP port • Bi-directional • Secure port (SSL) <p>Optionally used by EEM client libraries to connect to EEM for IT PAM security data, if SSL connectivity has been configured. Data is encrypted using SSL.</p>
5250	LDAP – EEM		<ul style="list-style-type: none"> • TCP port • Bi-directional <p>Used by EEM Server for UI management.</p>

*Each Orchestrator must have its own port if multiple Orchestrators are configured on the same host.

Firewall Considerations

CA IT PAM Orchestrators communicate with each other over JXTA (7001) and Web Service (8080) ports. The best way to observe this behavior is to trace Orchestrator to Orchestrator communication. For example:



```

C:\>netstat -a | findstr 7001
TCP    LOD0270:7001          LOD0270-0             LISTENING
TCP    LOD0270:7001          LODUM1221.ca.com:1673 ESTABLISHED
TCP    LOD0270:7001          LODUM1221.ca.com:1679 ESTABLISHED
TCP    LOD0270:7001          LOD0271.ca.com:2978   ESTABLISHED
TCP    LOD0270:7001          LOD0271.ca.com:3156   ESTABLISHED
TCP    LOD0270:7001          LOD0271.ca.com:3161   ESTABLISHED
TCP    LOD0270:7001          LOD0275.ca.com:1665   ESTABLISHED
TCP    LOD0270:7001          LOD0275.ca.com:1675   ESTABLISHED
TCP    LOD0270:7001          LOD0275.ca.com:2245   ESTABLISHED
TCP    LOD0270:7001          LOD0275.ca.com:4201   ESTABLISHED
TCP    LOD0270:7001          ITPAMS3.ca.com:3260   ESTABLISHED
TCP    LOD0270:7001          ITPAMS3.ca.com:3822   ESTABLISHED
TCP    LOD0270:7001          ITPAMS3.ca.com:4793   ESTABLISHED
TCP    LOD0270:7001          ITPAMS4.ca.com:1702   ESTABLISHED
TCP    LOD0270:7001          ITPAMS4.ca.com:1793   ESTABLISHED
TCP    LOD0270:7001          ITPAMS4.ca.com:4611   ESTABLISHED
C:\>_

```

In this example you can see that CA IT PAM Primary Domain Orchestrator ("LOD0270") communicates with other Orchestrators in the same Domain – such as the CA IT PAM Secondary Domain Orchestrator ("LOD0271"), CA IT PAM Primary Orchestrator ("LOD0275") and CA IT PAM Cluster Node for the Primary Orchestrator ("ITPAMS4") – by sending traffic from 7001 back to the source ports opened by those Orchestrators. The ports opened by the Orchestrators are 2978, 3156, 3161, 1665, 1675, 2245, 4201, 1702, 1793, and 4611 respectively. Orchestrators (clients) open a source port and send traffic to a destination port, which is 7001 by default.

Consequently, firewalls must be configured to allow bi-directional communication between Orchestrators, that is, to allow traffic from ANY to 7001 and from 7001 to ANY, where ANY is a random port greater than 1024. By default, when an application requests a socket from the system for an outbound call, a port between the values of 1024 and 5000 is supplied. For example, if we look at sockets in ESTABLISHED state on the CA IT PAM Secondary Domain Orchestrator (LOD0271), we see the outbound connection to the CA IT PAM Primary Domain Orchestrator (LOD0270 – destination port 7001) as well as inbound connections from other CA IT PAM Orchestrators, such as the Primary Orchestrator (LOD0275).

```

C:\>netstat -a | findstr 7001
TCP    LOD0271:4634      LOD0271.ca.com:7001  ESTABLISHED
TCP    LOD0271:4635      LOD0271.ca.com:7001  ESTABLISHED
TCP    LOD0271:4637      LOD0270.ca.com:7001  ESTABLISHED
TCP    LOD0271:4641      LOD0270.ca.com:7001  ESTABLISHED
TCP    LOD0271:4832      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4842      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4847      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4850      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4852      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4853      LOD0271.ca.com:7001  TIME_WAIT
TCP    LOD0271:4854      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4856      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4857      LOD0271.ca.com:7001  TIME_WAIT
TCP    LOD0271:4861      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4865      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4867      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4872      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4873      LOD0270.ca.com:7001  TIME_WAIT
TCP    LOD0271:4874      LOD0270.ca.com:7001  ESTABLISHED
TCP    LOD0271:7001      LOD0271:0           LISTENING
TCP    LOD0271:7001      LOD0271.ca.com:4634  ESTABLISHED
TCP    LOD0271:7001      LOD0271.ca.com:4635  ESTABLISHED
TCP    LOD0271:7001      LOD0275.ca.com:3823  ESTABLISHED
TCP    LOD0271:7001      LOD0275.ca.com:3825  ESTABLISHED
TCP    LOD0271:7001      LOD0275.ca.com:3828  ESTABLISHED
TCP    LOD0271:7001      LOD0275.ca.com:3830  ESTABLISHED
TCP    LOD0271:7001      ITPAMS3.ca.com:1440  ESTABLISHED
TCP    LOD0271:7001      ITPAMS3.ca.com:1443  ESTABLISHED
  
```

Note: It is normal to have sockets in the TIME_WAIT state (clients enter this state after an active close). RFC793 states that sockets can stay in TIME_WAIT state for a period of twice the Maximum Segment Lifetime (MSL). Since MSL is specified to be 2 minutes, a socket could be in a TIME_WAIT state for twice that – as long as 4 minutes.

As previously noted we have to consider another port as well – 8080 (Web Service – 8443 when using SSL). Firewalls must be configured to allow bi-directional communication between Orchestrators over the Web Service port. In other words, to allow traffic from ANY to 8080 and from 8080 to ANY, where ANY is a random port greater than 1024.

All other ports are used exclusively within a cluster – required by JBoss for intra-cluster management.

Note: We do not recommend splitting clusters between WAN links/firewalls. The cluster configuration (as it is today) uses multicast and, if multicast communication is not open end-to-end between cluster nodes, they are effectively not part of the same cluster, leading to unpredictable CA IT PAM behavior.

CA IT PAM Orchestrators and Agents communicate with each other using a pair of JXTA ports. As listed above, the default port setting for Orchestrators is 7001 while the default setting for Agents is 7003.

The best way to observe this behavior is to trace Orchestrator-Agent communication.




```
C:\>netstat -a | findstr 7001
TCP    LOD0270:1231          LOD0270.ca.com:7001    ESTABLISHED
TCP    LOD0270:7001          LOD0270:0               LISTENING
TCP    LOD0270:7001          LODUM1221.ca.com:3736  ESTABLISHED
TCP    LOD0270:7001          LODUM1221.ca.com:3742  ESTABLISHED
TCP    LOD0270:7001          LODUM1221.ca.com:3743  ESTABLISHED
TCP    LOD0270:7001          LOD0270.ca.com:1231    ESTABLISHED
TCP    LOD0270:7001          ITPAMS3.ca.com:2447    ESTABLISHED
TCP    LOD0270:7001          ITPAMS3.ca.com:2451    ESTABLISHED
TCP    LOD0270:7001          ITPAMS3.ca.com:2456    ESTABLISHED
C:\>_
```

In this example you can see that the CA IT PAM Primary Domain Orchestrator (LOD0270) communicates with Agents in the same Domain – LODVM1221 and ITPAMS3 – by sending traffic from 7001 back to the source ports opened by the Agents (ports 3736, 3742, 3743, 2447, 2451, and 2456 respectively). Agents open a source port and send traffic to a destination port, which is 7001 by default.

```
C:\>netstat -a | findstr 7003
TCP    LODUM1221:7003        LODUM1221:0             LISTENING
TCP    LODUM1221:7003        LOD0275.ca.com:2745    ESTABLISHED
C:\>_
```

Likewise, Agents communicate with Orchestrators in the same Domain by sending traffic from 7003 back to the ports that the Orchestrators established. The port opened by the CA IT PAM Primary Orchestrator (LOD0275) is 2745.



Because of this, firewalls must be configured to allow bi-directional communication between Orchestrators and Agents, that is, to allow traffic from ANY to 7001 and from 7001 to ANY as well as from ANY to 7003 and from 7003 to ANY, where ANY is a random port greater than 1024. For example, if we look at sockets in ESTABLISHED state on the CA IT PAM Primary Domain Orchestrator (LOD0270 – above), we see the inbound connection from Agents (LODVM1221 and ITPAMS3) to destination port 7001. The same is true for the socket in ESTABLISHED state on the LODVM1221 Agent where we see an inbound connection from the CA IT PAM Primary Orchestrator (LOD0275) to destination port 7003.

As with Orchestrator-Orchestrator communications, firewalls must be configured to allow bi-directional communication between Orchestrators and Agents over the Web Service port, that is, to allow traffic from ANY to 8080 (8443 when using SSL) and from 8080 (8443 when using SSL) to ANY, where ANY is a random port greater than 1024.

You should also make sure to configure your firewalls for bi-directional communication between the Orchestrators and their associated database servers and directory servers. Default port numbers are listed in the table.

Note: When using local firewalls on Orchestrator or Agent host machines, make sure that CA IT PAM executables can listen and connect bi-directionally through the firewall on each host. Some host-based firewall programs (such as Windows Firewall) allow exceptions for executables.

The CA IT PAM Management Console and Client communicate with Orchestrators using Web Services port 8080 or Apache HTTP Server/Load Balancer port 80 when Orchestrators are clustered.

Resolving Port Conflicts

When you install Orchestrators or Agents you can typically accept the default network port settings that the installer provides. Typical reasons for selecting ports other than the default ones include:

- The default ports are being used by other applications on the host.
- The default ports are being used by other Orchestrators or Agents already installed on the host.
- There is a restriction that prevents you from configuring the firewall to allow traffic on the default ports.

If you want to use ports other than the default ports, select values between 5000 and 9999, as these typically avoid most conflicts with the operating system and other applications. Well-known (privileged) port numbers should not be used as there might be conflicts with the operating system or other applications.

Port settings for Orchestrators and Agents are configured during installation. For a previously installed Orchestrator, you can modify these ports by editing the `OasisConfig.properties` file.

Last Updated: April 26, 2010

As noted earlier, Orchestrator port settings include the following parameters in the file:

```
oasis.jxta.port=7001
tomcat.connector.http.port=8080
jboss.jndi.port=1099
jboss.rmi.port=1098
oasis.snmptrigger.service.port=162
tomcat.secure.port=8443
```

CA IT PAM Domain
General Properties

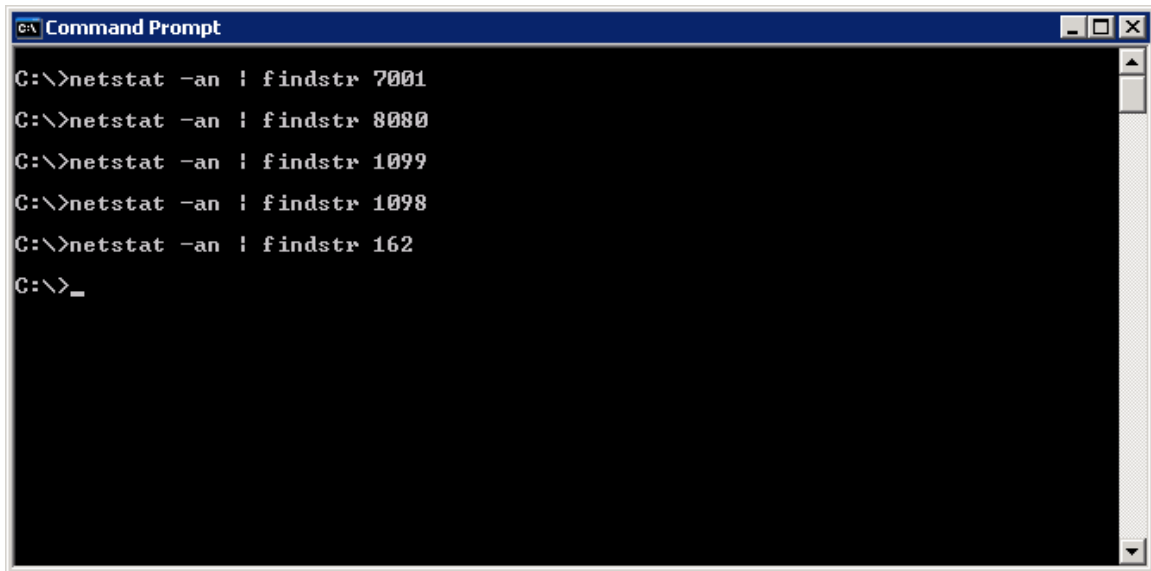
Server Host: ITPAMS1.ca.com
Display Name: ITPAMS1.ca.com
Server Port: 7001
Http Port: 8080
JNDI Port: 1099
RMI Port: 1098
SNMP Port: 162
Https Port: 8443

Support Secure Communication
 Install as Windows Service

< Back Next > Cancel

The best way to ensure that these ports are not already in use is by tracing them.





```
C:\>netstat -an | findstr 7001
C:\>netstat -an | findstr 8080
C:\>netstat -an | findstr 1099
C:\>netstat -an | findstr 1098
C:\>netstat -an | findstr 162
C:\>_
```

However, as noted earlier, there are a number of other ports that are not exposed at installation to consider. This includes:

- jboss.rmi.classloader.webservice.port
- jboss.rmi.object.port
- jboss.pooledinvoker.serverbind.port
- jboss.ha.jndi.port
- jboss.ha.jndi.rmi.port
- jboss.ha.rmi.object.port
- jboss.ha.pooledinvoker.serverbind.port
- jboss.mcast.jndi.autodiscovery.port
- tomcat.secure.port (when using SSL)
- tomcat.connector.ajp.port (worker.properties parameter (Load Balancer): worker.nodename.port)
- jboss.uil.serverbind.port

For this reason it is always a good idea to check the `c2o.log` file for port conflicts. For example:

```
2010-03-18 01:33:29,062 WARN [org.jboss.system.ServiceController] [ main]
Problem starting service jboss:service=WebService
java.lang.Exception: Port 8083 already in use.

2010-03-18 01:33:29,671 WARN [org.jboss.system.ServiceController] [ main]
Problem starting service jboss:service=invoker,type=jrmp
java.rmi.server.ExportException: Port already in use: 4444; nested exception is:
    java.net.BindException: Address already in use: JVM_Bind

2010-03-18 01:33:29,796 WARN [org.jboss.system.ServiceController] [ main]
Problem starting service jboss:service=invoker,type=pooled
java.lang.Exception: Port 4446 is already in use

2010-03-18 01:33:39,953 WARN [org.jboss.system.ServiceController] [ main]
Problem starting service jboss:service=HAJNDI
java.rmi.server.ExportException: Port already in use: 1101; nested exception is:
    java.net.BindException: Address already in use: JVM_Bind
```



Last Updated: April 26, 2010

```
2010-03-18 01:33:40,093 WARN [org.jboss.system.ServiceController] [          main]
Problem starting service jboss:service=invoker,type=jrmpha
java.rmi.server.ExportException: Port already in use: 4447; nested exception is:
    java.net.BindException: Address already in use: JVM_Bind

2010-03-18 01:33:40,125 WARN [org.jboss.system.ServiceController] [          main]
Problem starting service jboss:service=invoker,type=pooledha
java.lang.Exception: Port 4445 is already in use

2010-03-18 01:34:27,796 INFO [org.apache.coyote.http11.Http11Protocol] [
main] Starting Coyote HTTP/1.1 on http-0.0.0.0-9080
2010-03-18 01:34:27,937 INFO [org.apache.jk.common.ChannelSocket] [          main]
Port busy 8009 java.net.BindException: Address already in use: JVM_Bind
```

In this example we can see that there are conflicts on the following ports:

8083 - jboss.rmi.classloader.webservice.port

4444 - jboss.rmi.object.port

4446 - jboss.pooledinvoker.serverbind.port

1101 - jboss.ha.jndi.rmi.port

4447 - jboss.ha.rmi.object.port

4445 - jboss.ha.pooledinvoker.serverbind.port

8009 - tomcat.connector.ajp.port (and worker.nodename.port in the worker.properties file - Load Balancer - if the Orchestrator is clustered)

To resolve this, shut down the Orchestrator and edit the OasisConfig.properties to fix those conflicting ports:

```
jboss.rmi.classloader.webservice.port=9083
```

```
jboss.rmi.object.port=9444
```

```
jboss.pooledinvoker.serverbind.port=9446
```

```
jboss.ha.jndi.rmi.port=9101
```

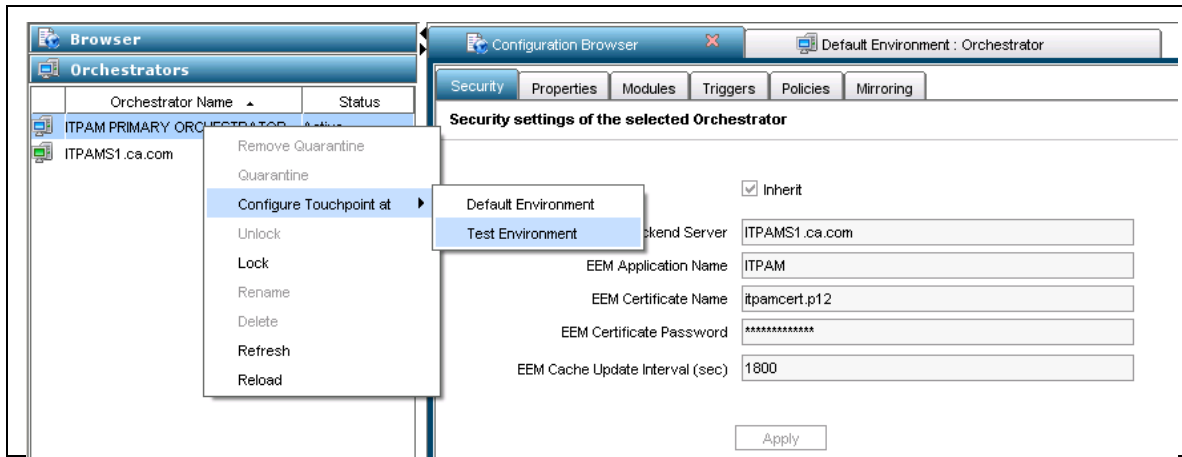
```
jboss.ha.rmi.object.port=9447
```

```
jboss.ha.pooledinvoker.serverbind.port=9445
```

```
tomcat.connector.ajp.port=9009
```

Once the Orchestrator re-starts, launch the CA IT PAM Client, and wait for the Orchestrator to appear under the Orchestrators palette (in the Configuration Browser). As soon as the Orchestrator successfully completes initialization and mirroring it is ready to be added to an Environment.





Using CA EEM to Manage Access to CA IT PAM

CA Embedded Entitlements Manager (CA EEM) manages User, Group and ACL policies for CA IT PAM, providing an additional level of stability, robustness and functionality and enabling access to be managed in a centralized manner. Since CA EEM allows direct interfaces to existing LDAP Directory services you can continue to adhere to existing LDAP Directory Service Administration best-practices that may be in effect in your current environment.

When LDAP is used in CA IT PAM, the configured settings determine security management. For example, if you must set monitoring permission on a Process, you must access that particular process and change the permission in the security browser.

When CA EEM is used, it determines the permissions instead of CA IT PAM. As a result, policies are created in CA EEM and the permissions are defined in those policies. CA IT PAM reads those policies and defines the permissions.

CA IT PAM v2.2 supports CA EEM v8.4.100.0 or later. CA EEM is installed as an independent prerequisite prior to CA IT PAM installation. Although CA EEM can be installed on the same hardware as CA IT PAM, it is recommended that it be installed on a separate server for production environments.

When CA EEM is used it:

- provides a single interface to connect to any LDAP Directory Service
- hides external LDAP Directory Service configurations from CA IT PAM
- manages ACLs on CA IT PAM Objects using Policies and thus eliminating the need for CA IT PAM to manage ACLs as part of its Library
- supports multi-tenancy (for v2.2)
- supports token-based single sign-on
- supports certificate based authentication and authorization for CA IT PAM web services
- supports LDAP Directory Services failover management through CA EEM
- provides additional reporting
- provides an additional level of stability, robustness and functionality to CA IT PAM



LDAP Data can be managed using CA EEM from one of the following external locations:

- Internal built-in LDAP Repository
- Microsoft ActiveDirectory
- SUN ONE Directory
- CA Identity Manager
- Novell eDirectory
- custom mapped directory

Failover configuration and management is detailed in the CA EEM documentation that is provided with the installed software. CA EEM Failover is managed externally, outside the realm of CA IT PAM, and is not documented here.

As note earlier, CA EEM must be installed and configured prior to CA IT PAM installation. When the CA IT PAM resource definitions are loaded into CA EEM successfully, a certificate file will be created in the current location. The default attributes for this file are

- File name: `itpamcert.p12`
- Password: `itpamcertpass`

This should be modified by editing the `itpam_eem.xml` file before loading resource definitions. For example:

```
<Register certfile="itpamcert.p12" password="itpamcertpass">
<ApplicationInstance name="ITPAM" label="ITPAM" />
</Register>
```

To load resource definitions in CA EEM do the following:

1. Copy the `itpam_eem.xml` file to the `...\CA\SharedComponents\iTechnology` folder.
2. Open a Command Prompt window and change to the `...\CA\SharedComponents\iTechnology` folder.
3. Run the `safex` command to load the resources and produce the certificate file for CA IT PAM. For example,

```
safex -h <hostname> -u <user> -p <password> -f <fully qualified path to itpam_eem.xml>
```

where:

- > `<hostname>` is the CA EEM hostname
 - > `<user>` is the CA EEM Administrator – `EiamAdmin`
 - > `<password>` is the password for the CA EEM Administrator, provided at installation time
4. Copy this certificate file to CA IT PAM hosts as it is required by the installer when installing Orchestrators.

Note: The `itpam_eem.xml` file should be deleted after resource definitions are loaded into CA EEM and the certificate file has been created.

CA IT PAM requires a minimum of 2 user groups (or **roles**):



Last Updated: April 26, 2010

- A group that is designated as Administrators: ITPAMAdmins
This group MUST be a direct member of the ITPAMUsers group.
- A group that is designated as users: ITPAMUsers

Internal LDAP Repository

By default, when CA EEM is configured to use its *internal* LDAP Repository, CA IT PAM creates these two user groups under the CA IT PAM application. ITPAMAdmins is the predefined CA IT PAM Administrators group and it consists of a single member – itpamadmin (user). ITPAMUsers is the predefined CA IT PAM Users group which has the following members:

- itpamuser (user)
- ITPAMAdmins (sub-group)

The default password for the user is the same as the user ID (all lowercase letters). However, it should be changed in CA EEM. To do this launch the CA EEM UI and log into CA EEM as EiamAdmin (ITPAM Application).



CA Embedded Entitlements Manager

Application: ITPAM

User Name: EiamAdmin

Password:

Remember my settings

Log In

ca

Copyright © 2008 CA. All rights reserved.

Then, click on the Manage Identities tab and search for the CA IT PAM user whose password you want to change.

Check the Reset Password checkbox in the Authentication group and enter the New Password (password must be confirmed). Finally, click on the Save button.

Administrators must be a member of both the Administrator (ITPAMAdmins) and User (ITPAMUsers) groups. Users, on the other hand, need only be added to the user group.

Additionally, the other resources, such as Default Policies and available operations on CA ITPAM Objects, are also loaded into CA EEM. These users, groups, and policies can be fine-tuned in CA EEM to manage CA IT PAM User Security and CA IT PAM Objects' Access Rights.

External LDAP Repository

By default, when CA EEM is configured to use an *external* LDAP Repository, CA IT PAM creates the ITPAMAdmins and ITPAMUsers user groups under ITPAM Application, however, the users are not created automatically. CA EEM will automatically load the users from the external LDAP Directory Service as Global Users in CA EEM but it is up to the CA EEM Administrator to link these Global Users in CA EEM into these application groups as required for CA IT PAM access.

Note: If CA EEM is not used, then the following has to be set up in the LDAP Directory Service prior to installing CA IT PAM:

- Create an Organizational Unit (ITPAMRoot)
- Create 2 Roles: ITPAMAdmins and ITPAMUsers
 - ITPAMAdmins is the predefined ITPAM Administrators Group and contains a single member: itpamadmin (user)
 - ITPAMUsers is a predefined ITPAM Users group with the following members: itpamadmin (user) and the ITPAMAdmins (sub-group)
 - The default password for the users (all lowercase): itpamdemo
- Create appropriate access rights to the Organizational Unit to allow read access for all users



Securing and Isolating Environments and Objects

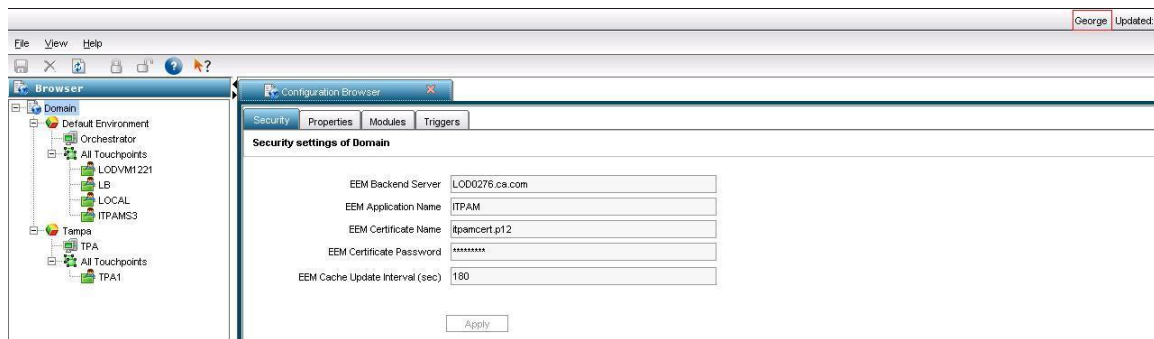
The Environment content administrator has full permissions to all Automation Objects in the Libraries for all the Orchestrators in an Environment. The owner of a folder or Automation Object similarly has permissions to that Automation Object. The owner of a folder or Automation Object and the Environment content administrator can grant permissions to other users or groups defined in the LDAP directory.

The user who creates a folder or Automation Object has initial ownership of the Automation Object. Ownership permissions cannot be removed except by specifying a different owner for an Automation Object. Either the owner of an Automation Object or the Environment content administrator can transfer ownership to a different user.

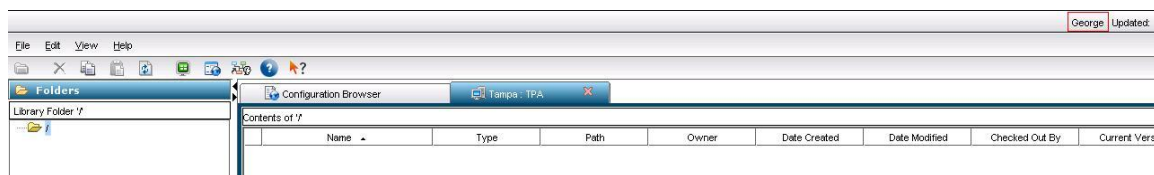
By default, a new folder or Automation Object initially assumes the permissions of its parent folder. The inheritance is then broken only if permissions are explicitly set on the folder or Automation Object. The inheritance can be optionally reformed when permissions are changed on the parent folder.

Permissions on a folder control access to the folder and to subordinate Automation Objects inside the folder. Unless permissions have been explicitly configured on the Automation Object or folder, it will automatically inherit the permissions of its parent folder. Changes to permissions on a folder automatically apply to all subordinate folders and Automation Objects that have not been explicitly assigned permissions. You must be the owner or be an Environment Content Administrator to change permissions for an Automation Object.

For example, the Environment Policy created by default while registering CA IT PAM with CA EEM specifies that the ITPAMUsers Group has the Environment role `ENVIRONMENT_USER`.



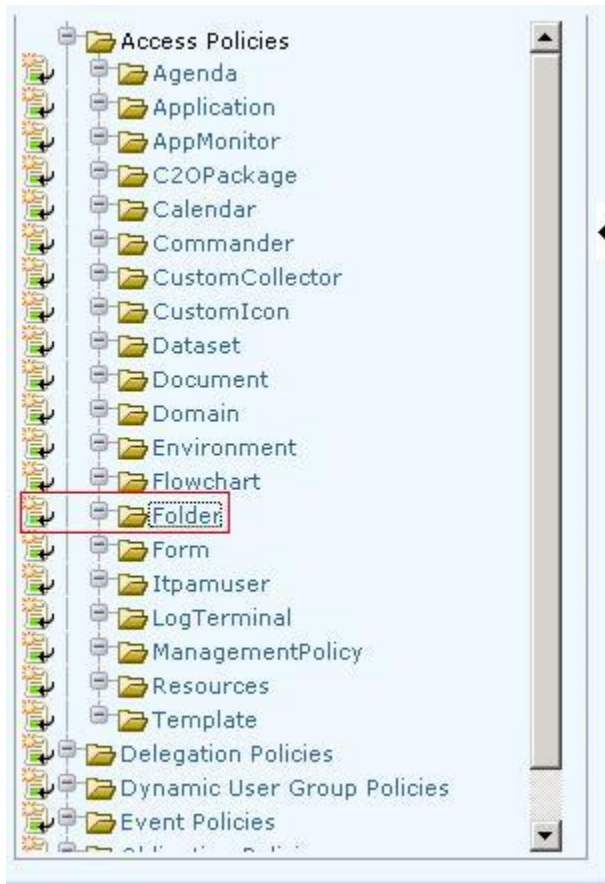
Therefore, CA IT PAM users can list Environments. However, the `ENVIRONMENT_USER` role has no impact on Automation Objects such as Folders. In other words, there are no implicit permissions, and ITPAMUsers members (`ITPAM_USER` role) cannot read CA IT PAM Library items.



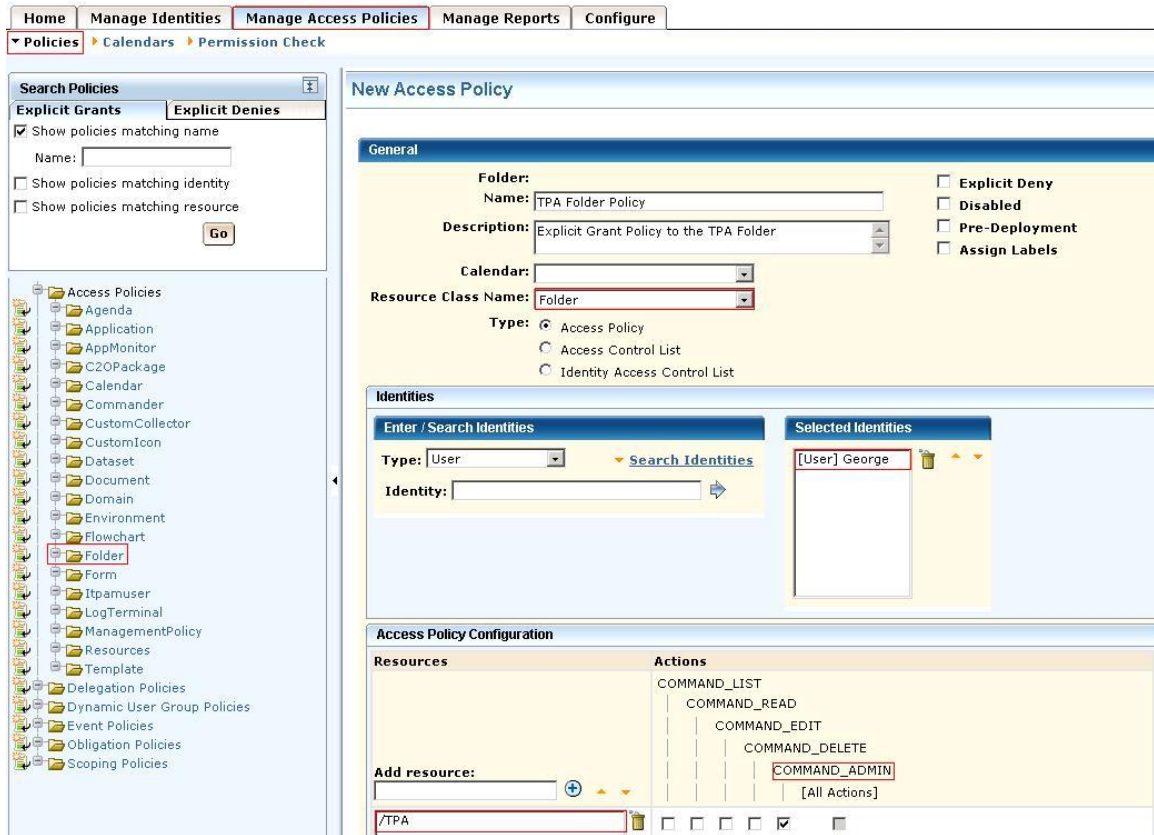
CA EEM Policies can be used to fine-tune access to CA IT PAM Automation Objects, by assigning permissions to them. To do this:



1. Login to the CA IT PAM application in CA EEM (ITPAM Application).
2. Select the Manage Access Policies tab.
3. Click the icon next to the default Resources Classes to create a policy. For example, Folder.

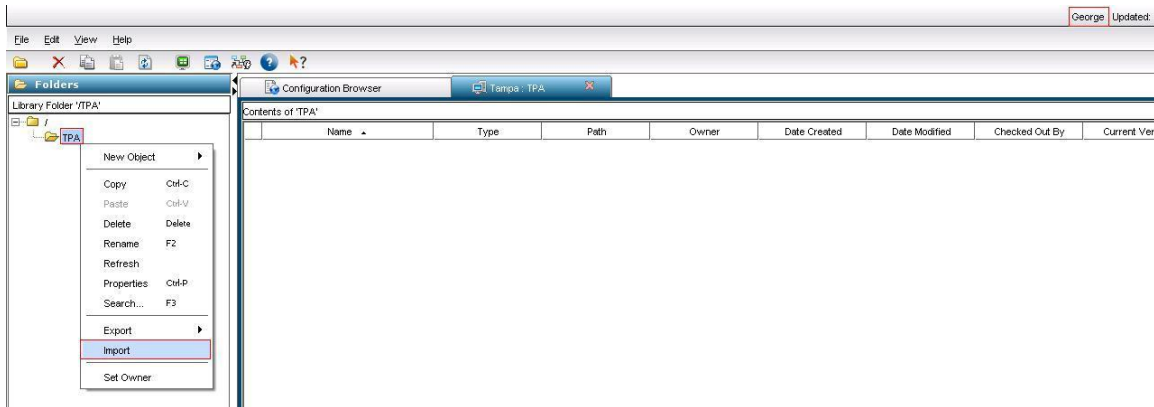


4. Provide a Policy Name and search for the users and groups for which you want to provide permissions. For example, "George"
5. In the Add Resource field, enter the **complete** path of the object. For example, /TPA (/Folder/ProcessName for Process objects).
6. Designate the access permissions. To do this select the check box to provide permissions (List, Read, and so forth) for the Resource (objects) added against the identities (users and groups) added. For example, "Admin" specifies full permissions to a folder or object.
7. Save the policy.

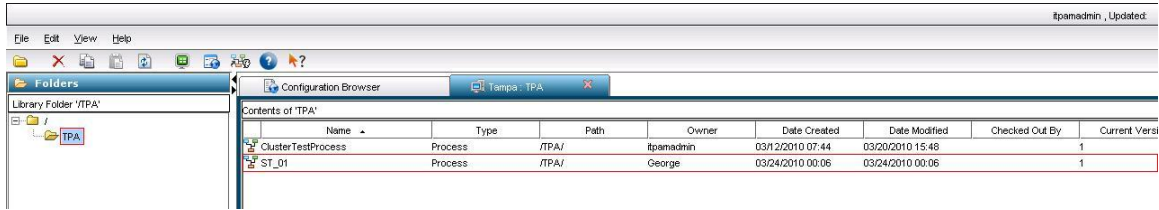


The policies can be applied to specific environments by specifying filters. Environment filters work by filtering based on the environment name. In this case, the policy is valid only with the environment provided in the filter.

CA IT PAM User "George" now has all permissions for Folder "TPA".



This includes the ability to import items into the TPA folder.



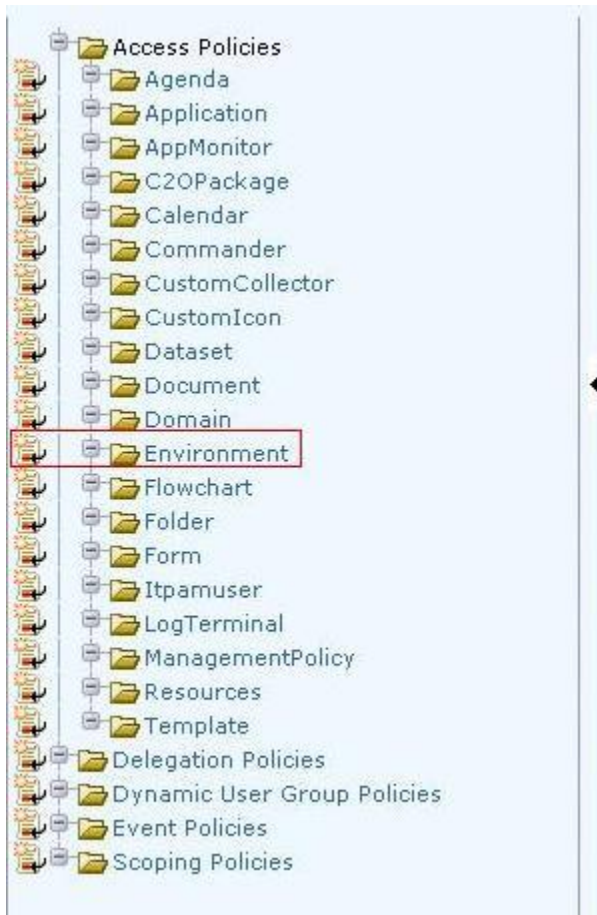
CA EEM policy can also be used to assign roles to users and groups. For example, suppose we want to explicitly deny user "Alex" access to the "Tampa" Environment. To do this:

1. Log in to the CA IT PAM application in CA EEM (ITPAM Application).
2. Select the Manage Access Policies tab.

The CA IT PAM Administrator can create either an Explicit Grant or an Explicit Deny policy. An Explicit Deny policy will be created in this case.

3. Click the icon next to the default Resource Classes that display for assigning roles.

For example, for Environment:



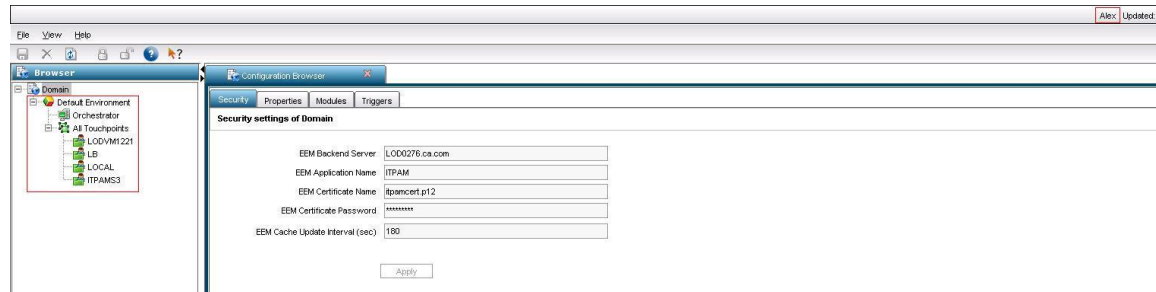
4. Provide a Policy Name and search for the users and groups for which you want to provide roles. In our example "Alex".

The screenshot shows the 'Identity Access Control List Configuration' dialog box. It has a title bar and a main area with the following fields: 'Type' set to 'User', 'Attribute' set to 'User Name', 'Operator' set to 'LIKE', and 'Value' is empty. To the right, there is a list of identities: 'Alex', 'George', 'itpamadmin', and 'itpamuser'. A 'Search' button is at the bottom left, and a blue arrow button is at the bottom right.

5. Select the Environment role (ENVIRONMENT_USER) to be denied for the resource (Tampa Environment) added against user Alex.
6. Save the policy.

The screenshot shows the 'Policy' configuration page in a web application. The navigation bar includes 'Home', 'Manage Identities', 'Manage Access Policies', 'Manage Reports', and 'Configure'. The breadcrumb trail is 'Policies > Calendars > Permission Check'. On the left, there is a 'Search Policies' section with tabs for 'Explicit Grants' and 'Explicit Denies'. Below it is a tree view of policy categories, with 'Environment' highlighted. The main area is titled 'Policy' and contains a 'General' section with fields for 'Folder' (Environment Deny Policy), 'Description' (Explicitly Denies user Alex Access to the Tampa Environment), 'Calendar', and 'Resource Class Name' (Environment). There are checkboxes for 'Explicit Deny', 'Disabled', 'Pre-Deployment', and 'Assign Labels'. Below this is an embedded 'Identity Access Control List Configuration' dialog box, identical to the one in the previous screenshot. Underneath the dialog is a 'Selected Identities' table with columns for 'Identities' and 'Actions'. The table lists 'ENVIRONMENT_CONTENT_ADMIN', 'ENVIRONMENT_CONFIG_ADMIN', and 'ENVIRONMENT_USER'. The 'ENVIRONMENT_USER' row has a checked checkbox in the 'Actions' column. At the bottom, there is a 'Resources' section with 'Tampa' listed.

Notice that, unlike user George, user Alex now has the Environment role ENVIRONMENT_USER for the Default Environment only.



Securing the JBoss Server

Security is a fundamental concern for all organizations, which must be able to restrict who is allowed to access applications and control what operations application users may perform. JBoss includes several admin access points that need to be secured or removed to prevent unauthorized access to administrative functions in a deployment. Following is a list of those admin services:

- The `jmx-console.war` found in the `deploy` directory provides an HTML view into the JMX microkernel. As such, it provides access to arbitrary admin type access such as shutting down the server, stopping services, deploying new services, etc. It can be secured using J2EE role based security. The security setup is based on two pieces:
 - > standard `WEB-INF/web.xml` servlet URI to role specification
 - > `WEB-INF/jboss-web.xml` specification of the JAAS configuration which defines how authentication and role mapping is performed.

The security constraint block in `jmx-console.war/WEB-INF/web.xml` is already uncommented and restricts access to the HTML JMX Console to users with the role `JBossAdmin` (`admin`) by default in CA IT PAM. Likewise, the security-domain block in `jmx-console-war/WEB-INF/jboss-web.xml` is already uncommented and security is enabled by default in CA IT PAM. Users and passwords are set in the following file:

```
<CA IT PAM Installation Folder>\server\c2o\conf\props\jmx-console-users.properties and
roles are defined in <CA IT PAM Installation Folder>\server\c2o\conf\props\jmx-console-
roles.properties
```

Since JMX Console passwords are stored in clear text, CA IT PAM encrypts the admin's password and stores it in `OasisConfig.properties` (`CERTPASSWORD` parameter value) file. `com.optinuity.c2o.security.jaas.JMXConsoleLoginModule` built into CA IT PAM, a stronger authentication mechanism, validates it.

- The `web-console.war` found in the `deploy/management` directory (`console-mgr.sar`) is another Web application view into the JMX microkernel. It uses a combination of an applet and a HTML view and provides the same level of access to admin functionality as the `jmx-console.war`. It can be secured using J2EE role based security. It also has a skeleton setup that allows for enabling security using username/password/role mappings found in the `web-console.war` deployment in the corresponding `WEB-INF/classes/web-console-users.properties` and `WEB-INF/classes/web-console-roles.properties` files.

The security constraint block in `web-console.war/WEB-INF/web.xml` is already uncommented and restricts access to the Web Console to users with the role `JBossAdmin (admin)` by default in CA IT PAM. Likewise, the security-domain block in `web-console-war/WEB-INF/jboss-web.xml` is already uncommented and security is enabled by default in CA IT PAM. Users and passwords are set in `<CA IT PAM Installation Folder>\server\c2o\deploy\management\console-mgr.sar\web-console.sar\WEB-INF\classes\web-console-users.properties` and roles are defined in `<CA IT PAM Installation Folder>\server\c2o\deploy\management\console-mgr.sar\web-console.sar\WEB-INF\classes\web-console-roles.properties`. Since Web Console passwords are stored in clear text, CA IT PAM encrypts the admin's password and stores it in `OasisConfig.properties` (`CERTPASSWORD` parameter value) file.

`org.jboss.security.auth.spi.UsersRolesLoginModule` built into CA IT PAM, a stronger authentication mechanism, validates it.

- The `http-invoker.sar` found in the `deploy` directory is a service that provides RMI/HTTP access for EJBs and the JNDI Naming service. This includes a servlet that processes posts of marshaled `org.jboss.invocation.Invocation` objects that represent invocations that should be dispatched onto the `MBeanServer`. Effectively this allows access to MBeans that support the detached invoker operation via HTTP since one could figure out how to format an appropriate HTTP post. To secure this access point you would need to secure the `JMXInvokerServlet` servlet found in the `http-invoker.sar/invoker.war/WEB-INF/web.xml` descriptor. There is a secure mapping defined for the `/restricted/JMXInvokerServlet` path by default.

For more information, please, refer to <http://www.jboss.org>.