Cisco AON Administration Guide

AON Release 2.4
March 2007

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCVP, the Cisco Logo, and the Cisco Square Bridge logo are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networking Academy, Network Registrar, *Packet*, PIX, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0612R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

AON Administration Guide
© 2005 – 2007 Cisco Systems, Inc. All rights reserved.

CONTENTS

Preface

Cisco Application-Oriented Network (AON) is a technology foundation for a class of Cisco products that embed intelligence into the network for the support of distributed application deployment. AON complements existing networking technologies by allowing increased visibility of the information passing through the network. This facilitates efforts to accomplish the following:

- Integration of disparate applications by enabling the routing of customer-specified information and message types to the appropriate destination, in the format(s) needed.
- Enforcement of security policies for information access and exchange.
- Optimization of application traffic flows, both in terms of network bandwidth and processing overheads.
- Better management of information flow, including monitoring and metering of information flow for both business and infrastructure purposes.

AON works primarily at the message-level rather than at the packet level, allowing developers and system administrators to work with the content and context of information flow. Typically, an AON node occupies the terminus of a TCP connection so that it can inspect and work with the entire message, including the "payload" and all headers. AON can work with popular application-level protocols such as HTTP, JMS, and other de facto standards.

AON Devices and Components

An application-oriented network consists of the following devices and components:

- Management Tools
- Nodes
- Other Entities

Management Tools

AON Management Console (AMC)

AMC is the software package that centralizes management of the application-oriented network. This includes:

- Configuring, managing, and monitoring AON nodes
- Deploying global and node-level properties

- Managing security, including certificates, keypairs, and users

AON Development Studio (ADS)

ADS is the tool for developers to create message-level logic using a graphical user interface (GUI). ADS provides a set of preconfigured functions, called Bladelets, that are used to construct Policy Execution Plans (PEP). Additionally, ADS includes functionality that enables developers to upload custom Bladelets to perform business functions unique to different environments.

Nodes

AON Appliance

This is the AON form factor available as rack-mountable appliance. Typically this node is used in a data center.

AON Services Modules on Catalyst 6500 Series Switches

This is the AON form factor available as a single-slot services module for the Catalyst 6500 Series Switches. Typically this node is used in a data center.

AON Network Modules on Cisco 2600, Cisco 2800, Cisco 3700, and Cisco 3800 Series Routers

This is the AON form factor available as a single-slot network module for several different Cisco modular access routers. Typically this type of node is used in a branch office. See *Release Notes for Cisco Application-Oriented Networking* for a detailed list of supported router platforms.

Other Entities

Depending on the configuration of your network and the needs of your business, your application-oriented network may include any of the following:

Database

When a database policy is configured, AON can store specified data in a Sybase or Oracle database.

LDAP server

Can be used to perform user authentication for both the AMC application and on individual messages traversing the application-oriented network.

Java Messaging Service (JMS)

AON devices can be configured to exchange messages between clients and JMS queues.

AON Features

Project Management

AON resources are partitioned in projects. A project contains all of the resources, such as policy execution plans (PEPs), message types, and property sets, created by a team. Nodes and other global resources are shared among multiple projects, yet changes to those resources within a given project do not affect other projects. A single node can simultaneously process message traffic from multiple projects.

Explicit and Transparent Interception

An AON node resides in the network as an inline application-aware device. The device acts as an intelligent intermediary gateway that can either be explicitly addressed by applications or as a passthrough proxy that is transparent to applications.

Access Methods and Adapters

AON understands various application access methods and provides adapters that can natively interface with commonly used protocols. The key protocols that AON supports include:

- HTTP v1.0/ v1.1 and HTTPS
- JMS
- MQ (through native adapter)

The AON software development kit (SDK) enables development of adapters for custom protocols.

Protocol Translation

AON nodes can act as protocol gateways between multiple applications that use differing protocols—as an example, a node can receive an application message through JMS and send the message information to another application as an HTTP post.

Transformation

AON supports both XML and non-XML transformation through an open transformation architecture. AON can function as an XSLT based transformation engine. You can add your own Java transformation engine to execute custom transformations.

Security

AON provides a series of intelligent services which enable message-level access and control to meet application security needs within the network. These security services include authentication, authorization, nonrepudiation, data integrity, data confidentiality, and centralized key management.

Service Virtualization

AON can act as proxy to create an abstraction layer for endpoint applications and apply policies across all of these services—in a centralized configuration, with distributed enforcement in the network. Service virtualization functionality supports execution of content-based routing, workload balancing, and message distribution operations.

Schema Validation

AON provides the ability to validate XML documents against schemas you create.

Reliable Messaging

AON provides a reliable delivery semantic across all supported protocols. Based on the level of support required, AON can ensure exactly once delivery, at least once delivery, or at most once delivery.

Optimization Services

AON has the capability to cache or compress messages to allow for optimization of message traffic, thus enabling reduced application response time and the conservation of network bandwidth.

External Data Access

AON enables access to or notification of other applications in parallel to the handling of the main message flow. External access is currently available using HTTP and Java Database Connectivity (JDBC).

Message Logging

AON can capture application messages for logging either synchronously, for auditing purposes, or asynchronously.

Related Documentation

The AON documentation set includes the following guides:

- Release Notes for Cisco Application-Oriented Networking
- *AON Installation and Upgrade Guide*—covers tasks related to installing and upgrading software on AON devices.
- *AON Development Studio User Guide*—covers the AON Development Studio, bladelets, and PEP creation.
- *AON Programming Guide*—covers the development of custom bladelets, custom adapters, and other features related to extending AON functionality.

Obtaining Documentation

Cisco documentation and additional literature are available on Cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this URL:

http://www.cisco.com/univercd/home/home.htm

You can access the Cisco website at this URL:

http://www.cisco.com

You can access international Cisco websites at this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation DVD

Cisco documentation and additional literature are available in a Documentation DVD package, which may have shipped with your product. The Documentation DVD is updated regularly and may be more current than printed documentation. The Documentation DVD package is available as a single unit.

Registered Cisco.com users (Cisco direct customers) can order a Cisco Documentation DVD (product number DOC-DOCDVD=) from the Ordering tool or Cisco Marketplace.

Cisco Ordering tool:

http://www.cisco.com/en/US/partner/ordering/

Cisco Marketplace:

http://www.cisco.com/go/marketplace/

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Ordering tool:

  http://www.cisco.com/en/US/partner/ordering/

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, USA) at 408 526-7208 or, elsewhere in North America, by calling 1 800 553-NETS (6387).

Documentation Feedback

You can send comments about technical documentation by using the embedded feedback form next to the document on Cisco.com or by writing to the following address:

Cisco Systems, Inc.
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Cisco Product Security Overview

Cisco provides a free online Security Vulnerability Policy portal at this URL:

http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html

From this site, you can perform these tasks:

- Report security vulnerabilities in Cisco products.
- Obtain assistance with security incidents that involve Cisco products.
- Register to receive security information from Cisco.

A current list of security advisories and notices for Cisco products is available at this URL:

http://www.cisco.com/go/psirt

If you prefer to see advisories and notices as they are updated in real time, you can access a Product Security Incident Response Team Really Simple Syndication (PSIRT RSS) feed from this URL:

http://www.cisco.com/en/US/products/products_psirt_rss_feed.html

Reporting Security Problems in Cisco Products

Cisco is committed to delivering secure products. We test our products internally before we release them, and we strive to correct all vulnerabilities quickly. If you think that you might have identified a vulnerability in a Cisco product, contact PSIRT:

- Emergencies—security-alert@cisco.com
- Nonemergencies—psirt@cisco.com

Tip We encourage you to use Pretty Good Privacy (PGP) or a compatible product to encrypt any sensitive information that you send to Cisco. PSIRT can work from encrypted information that is compatible with PGP versions 2.*x* through 8.*x*.

Never use a revoked or an expired encryption key. The correct public key to use in your correspondence with PSIRT is the one that has the most recent creation date in this public key server list:

http://pgp.mit.edu:11371/pks/lookup?search=psirt%40cisco.com&op=index&exact=on

In an emergency, you can also reach PSIRT by telephone:

- 1 877 228-7302
- 1 408 525-6532

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, Cisco Technical Support provides 24-hour-a-day, award-winning technical assistance. The Cisco Technical Support Website on Cisco.com features extensive online support resources. In addition, Cisco Technical Assistance Center (TAC) engineers provide telephone support. If you do not hold a valid Cisco service contract, contact your reseller.

Cisco Technical Support Website

The Cisco Technical Support Website provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The website is available 24 hours a day, 365 days a year, at this URL:

http://www.cisco.com/techsupport

Access to all tools on the Cisco Technical Support Website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register at this URL:

http://tools.cisco.com/RPF/register/register.do

Note Use the Cisco Product Identification (CPI) tool to locate your product serial number before submitting a web or phone request for service. You can access the CPI tool from the Cisco Technical Support Website by clicking the **Tools & Resources** link under Documentation & Tools. Choose **Cisco Product Identification Tool** from the Alphabetical Index drop-down list, or click the **Cisco Product Identification Tool** link under Alerts & RMAs. The CPI tool offers three search options: by product ID or model name; by tree view; or for certain products, by copying and pasting **show** command output.

Search results show an illustration of your product with the serial number label location highlighted. Locate the serial number label on your product and record the information before placing a service call.

Submitting a Service Request

Using the online TAC Service Request Tool is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool provides recommended solutions. If your issue is not resolved using the recommended resources, your service request is assigned to a Cisco TAC engineer. The TAC Service Request Tool is located at this URL:

http://www.cisco.com/techsupport/servicerequest

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)
EMEA: +32 2 704 55 55
USA: 1 800 553-2447

For a complete list of Cisco TAC contacts, go to this URL:

http://www.cisco.com/techsupport/contacts

Definitions of Service Request Severity

To ensure that all service requests are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down,” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- Cisco Marketplace provides a variety of Cisco books, reference guides, and logo merchandise. Visit Cisco Marketplace, the company store, at this URL:

  http://www.cisco.com/go/marketplace/

- *Cisco Press* publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to Cisco Press at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources. You can access Packet magazine at this URL:

  http://www.cisco.com/packet

- *iQ Magazine* is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions. You can access iQ Magazine at this URL:

  http://www.cisco.com/go/iqmagazine

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

  http://www.cisco.com/ipj

- World-class networking training is available from Cisco. You can view current offerings at this URL:

http://www.cisco.com/en/US/learning/index.html

CHAPTER 1

Project Management

AON Version 2.4 introduces the concept of projects to partition the work performed by different development teams. A project contains all of the resources, such as policy execution plans (PEPs), message types, and property sets, created by a team. Nodes and other global resources are shared among multiple projects, yet changes to those resources within a given project do not affect other projects. A single node can simultaneously process message traffic for multiple projects.

AON users are assigned permissions to access AON resources based on the roles assigned to them when they are created in AMC. Some users may be given permission to modify any resource or entity in the AON environment, while others may be limited to modifying a subset of resources contained in the project to which they are assigned.

AON now supports the concept of multi-tier environments, in which projects can be developed and thoroughly tested before being deployed to a production environment. Development nodes are managed by one AMC, while production nodes are managed by a different AMC. Projects are exported from the development tier and imported into the production tier.

This AON release also includes the Programmatic Management Interface (PMI) feature, which provides a scriptable interface to AMC. PMI enables users to develop custom applications that can be used to automate the task of deploying changes to AON nodes.

This chapter includes the following topics:

- Projects, page 1-1
- Role-Based Access Control, page 1-6
- Multiple AON Environments, page 1-7
- Programmatic Management Interface, page 1-10

Projects

Projects are the containers that hold all of the resources, including nodes, PEPs, message types, and other resources, that are assigned to a development team. A team working in a project is isolated from all other projects. Users can create and deploy resources to the AON network without regard for what other teams are doing with their own projects. Those changes, once deployed, have no bearing on how nodes used by other projects process messages for those projects.

Note When you log on to AMC, you must open a project before you can view the following items:

- Properties tab
- Deploy tab
- Monitor tab
- Licensing and Extensions on the Admin tab

These items are hidden until you open a project.

AMC maintains deployment requests (DR) for each project. A project has a DR for each node associated with that project. A project also has a global DR whose scope is all of the nodes associated with that project. Each PEP, property set, or message type created in a project is either bound to a single node or bound globally across all nodes in the project. When a global project resource is changed, it causes the resource to be deployed to all nodes associated with the project.

System Project

The system project contains resources that are shared by all application projects. For example, a property set created in a system project by a system administrator can be referenced by PEPs belonging to multiple projects. A user assigned to the system project, assuming the user has appropriate role-based permissions, can create and edit shared resources in the system project. It is also the responsibility of the system administrator to deploy shared resources to the appropriate nodes. These shared resources can be viewed and referenced by other projects as read-only resources.

Note PEPs and Message Types are not allowed in the System project. You cannot access the System project from ADS.

Project Resources

Since nodes are shared among projects, the resources configured on a node must not conflict with resources from other projects. When you create a new project, you also specify a prefix to be associated with that project. AMC then appends the project prefix to resources that are created by the project. This ensures that each resource has a unique name across different projects.

Resources in a project are either assigned to a particular node, or they have a global scope, meaning they are deployed to every node that is assigned to the project. Resources that are created within a project are bound to that project and are not accessible from any other project.

A newly installed AMC creates predefined resources in the system project. The predefined resources include:

- Adapters (aonp, http, jms, pmode)
- Property sets (see Table 1-1)

Table 1-1 Pre-Defined Property Sets

Property Set Category	Predefined Property Sets
Adapter listener domain	aonp, http, https
Adaptive load balancer	default

Table 1-1 Pre-Defined Property Sets (continued)

Property Set Category	Predefined Property Sets
QoS mapping	• bulk data transfer • default • mission critical • network management • transactional data
Fastpath	http listener port
Caching	caching
Delivery semantics	default, DS1
Global send properties	default, DC1
Node send properties	default, DC1, DS1
Node capabilities	default
Bladelet monitoring	default
SSL configuration	default SSL policy
Content lookup	default, default1
Content validation	default, default1

Extensions

Extensions are uploaded into and deployed from a particular project. Any property set categories included in the extension are accessible only to the project that uploaded the extension. Each project must upload its own unique extensions. An extension uploaded by an application project cannot be shared among other projects. However, extensions uploaded into the system project can be shared.

Extensions are subject to the following limitations:

- Adapters and adapter extensions are only supported in the project-global scope of the system project. They cannot be uploaded from a user project.
- Schemas, transforms, transform parsers, and JMS bindings are only supported in the project-global scope of a project.
- There can be only one instance of an extension package in AMC. If two different projects attempt to upload the same extension, the second upload attempt fails.

Upgrading from a Previous AON Release

If you upgrade a previous release to AON 2.4, the AMC installer creates an additional project. The default name is “USER_PROJECT,” however, you can specify a different name during the upgrade process. This project contains all PEPs and message types that were configured in AMC before the upgrade. All nodes and users that were configured in AMC before the upgrade are also assigned to this project. PEPs and message types in this project have a prefix, called “PREFIX” by default but can be set to something else during the upgrade process. Users configured in AMC before upgrade will have the same permission in this version of AMC.

Note that adapter and adapter extension packages can only be associated with system project, these resources will be assigned to system project after upgrade.

The sections that follow describe various aspects of working with projects, including:

- Creating a New Project, page 1-4
- Viewing a Project, page 1-4
- Opening a Project, page 1-5
- Editing a Project, page 1-5
- Assigning Users to a Project, page 1-5
- Assigning Nodes to a Project, page 1-5
- Deleting a Project, page 1-6

Creating a New Project

How to Get There

Go to **Project**, then click the **New** button. Complete the following fields:

- **Project name**—The project name must start with an alphanumeric character and may contain only letters, numbers, hyphens, and underscores. If you insert spaces in your project name, AMC removes them.
- **Project prefix**—The project prefix is appended to the names of resources created by that project. It is used to identify the project that created a given resource.
- **Project description**

After you click the **Next** button, subsequent pages provide you the opportunity to assign nodes and users to the project.

See the following additional topics:

- Opening a Project, page 1-5
- Editing a Project, page 1-5
- Assigning Users to a Project, page 1-5
- Assigning Nodes to a Project, page 1-5

Viewing a Project

How to Get There

Go to **Project**, then select a project and click the **View** button

The Project Details page displays the basic details of the selected project, including:

- **Project name**
- **Project prefix**—the prefix is used to identify resources assigned to that project.
- **Project description**
- **Assigned nodes**—nodes for which project members are able to create PEPs, message types, and other resources.
- **Assigned users**—users that have permission to modify project resources.

Opening a Project

How to Get There

Go to **Project**, then select a project and click the **Open** button. Doing so loads the Project Details page. From there you can click any of AMC's tabs to begin configuring the project.

Editing a Project

How to Get There

Go to **Project**, then select a project and click the **Edit** button. Doing so enables you to change the project description. You cannot change the project name or prefix.

Assigning Users to a Project

How to Get There

Go to **Project**, then select a project and click the **Assign Users** button.

To assign users, do the following:

1. Select a user from the **Available Users** column. Control+click to select multiple users.
2. Click the right arrow button to move the users to the **Assigned Users** column.
3. If necessary, select users from the **Assigned Users** column and use the left arrow button to remove them from the project.
4. You can also double-click a user to immediately move it from one column to the other.
5. Click the **Save** button to save your changes.

Users must be assigned to a project before they can modify it, and only assigned users can manipulate resources in that project. External users cannot be assigned to projects. A user can be assigned to more than one project, however, that user can access only one project at a time. Users with security, network, and system administrator roles are not shown, as they are assigned to all projects by default.

Additionally, users with the following roles are assigned to every project by default:

- System administrator
- Security administrator
- Network administrator

For additional information about users and role-based access control, see the "Managing Local Users" section on page 5-2.

Assigning Nodes to a Project

How to Get There

Go to **Project**, then select a project and click the **Assign Nodes** button.

To assign users, do the following:

1. Select a user from the **Available Nodes** column. Control+click to select multiple nodes.
2. Click the right arrow button to move the users to the **Assigned Nodes** column.

3. If necessary, select nodes from the **Assigned Nodes** column and use the left arrow button to remove them from the project.
4. You can also double-click a node to immediately move it from one column to the other.
5. Click the **Save** button to save your changes.

Deleting a Project

How to Get There

Go to **Project**, then select a project and click the **Delete** button.

Depending on whether the project has resources assigned or has been opened by other users, the following occurs:

- If resources have been configured, AMC asks to confirm the deletion. If you continue, deployment requests are generated. You must deploy these changes to the nodes in order to completely delete the project resources from the AON environment. After the deletion has been deployed, return to the Project List to delete the project
- If users have opened the project, you are asked to confirm the deletion. If you continue, the project is immediately deleted.
- If there are no project resources and no users have opened the project, the project is immediately deleted.

Role-Based Access Control

AON Version 2.4 introduces two new user types and modifies the capabilities of the four existing user types. The two new user types include:

- System administrator—a user that is able work with all resources. This user has no restrictions in the management of AMC, ADS, and nodes.
- Application developer—a user that is able to work only with resources contained in assigned projects.

The existing user types are modified as follows:

- Network administrator—a user that is able to work with network resources. This user can also open and view projects.
- Security administrator—a user that is able to work with security-related resources. This user can also open and view projects.
- Application administrator—a user that is able to create project, create users, and assign users to projects. This user can also manage project-related resources.
- Application designer—a user that is able to upload and register extensions. This user can also open and view projects.

For more information about user types, including the specific AMC pages each user can view and modify, see the "Managing AON Users" section on page 5-1.

Multiple AON Environments

AMC supports the use of multiple AON environments in the lifecycle of a project. For example, development, staging, and production activities can be carried out in three separate AON environments. Project teams can develop and test AON projects in the development tier. They can then request that their projects be promoted, first to the staging tier, then to the production tier.

It is assumed that AON nodes in each tier are managed by different AMC installations. For example, nodes used to test new PEPs in the development tier are not expected to process production traffic.

Promotion of AON resources from one tier to another is accomplished using the following steps:

1. The AMC of the source tier exports all of the resources for a project. The export operation creates a configuration archive. For more information, see Exporting Projects, page 1-8.
2. The archive is modified to adjust any environment-dependent parameters that differ between the source and destination tiers. You can create scripts to automate this task. For more information, see Archive File, page 1-8.
3. The configuration archive is imported into the same project in the AMC of the destination tier. For more information, see Importing Projects, page 1-9.
4. The imported resources are deployed to the nodes of the destination tier.

The export, import, and deploy operations are available through a programmatic interface to support control of promotion from a workflow automation tool. For more information, see Programmatic Management Interface, page 1-10.

Assumptions

This feature is subject to the following assumptions:

- Project resources created for a for a node can only be promoted to nodes running the same version of AON software and running on the same hardware.
- Version control of projects is accomplished by the user using an external version control system. Exported configuration archives, created by AMC, must be versioned and managed by the user.
- Rollback is accomplished by importing an earlier version of a project's configuration archive.
- Shared resources must be deployed before dependent project resources can be deployed.
- The name and prefix assigned to a project cannot be changed after project creation. The same prefix must be used in all tiers to which the project may be promoted.
- To prevent message type conflicts, it is assumed that each application team will apply distinct URI patterns in the message types they create.
- Message type ordering is required only at the Project Level (not needed between Projects due to URI namespace uniqueness).

This feature is subject to the following limitations:

- Project-level log and trace viewing is not supported in this release.
- Deployment request history will not be migrated from previous software releases during the AMC upgrade. Only the latest version of each resource will be migrated.

Exporting Projects

How to Get There

Go to **Admin > Data Migration**.

The Export Project page contains the following elements:

Element	Description
Configuration File Path	Used to specify a location on the AMC server's file system where the configuration file is to be created. For example, `/tmp/proj1-config-mm-dd-yy.zip` If the given path does not start with a /, the path is considered to be relative to the directory from which AMC is run. AMC must have write permissions to the given path. Additionally, you cannot specify a file name that already exists. AMC will not overwrite existing files.
Project List	Displays list of projects available. You must select one or more projects to export. The list also provides some basic information about the project like the project name, type (Standalone/Shared) and description.
Export	Click this button to write the configuration archive.

Archive File

When you export a project, AMC writes a zip file containing configuration data. This configuration archive contains all of the data related to the project. You can then edit the information contained in the archive to modify the configuration data that must change in order for the project to function in the new AMC environment.

The file structure of the configuration archive and schemas for individual data files are listed in the sections that follow. You can use this information to modify configuration files as appropriate for your environment. You can also develop custom scripts to automate this task.

Note Cisco reserves the right to modify archive schemas as AON evolves. Schemas are not guaranteed to be backward compatible. Thus custom scripts may need to be adjusted from one release to the next.

For the complete archive file structure and schema, see the "Archive Schema" section on page A-1.

Importing Projects

Before importing a project, you must prepare a node mapping file. This file maps the configuration for each node in the source AMC to any combination of nodes in the new AMC environment.

The example below shows a sample node mapping file. In this example, sourceNode1 is mapped to a single node on the destination AMC. The configuration for sourceNode2 is mapped to two nodes in the destination AMC. The configuration for sourceNode3 is not needed in the destination AMC, so it is not mapped to a node.

```
<PromotionNodeMap>

    <SourceNode name="sourceNode1">
       <DestinationNode name="destinationNode1"/>
    </SourceNode>

    <SourceNode name="sourceNode2">
         <DestinationNode name="destinationNode1"/>
         <DestinationNode name="destinationNode2"/>
    </SourceNode>

    <SourceNode name="sourceNode3"/>

</PromotionNodeMap>
```

Note To mimic the import behavior of previous AON releases, map only a single node in the source AMC to a single node in the destination AMC.

Create a node mapping file and store it in a location where AMC has read access.

How to Get There

Go to **Admin > Data Migration > Import**.

The Import Configuration page contains the following elements:

Element	Description
Configuration File Path	Used to specify a location of the configuration archive on the AMC server's file system. If the given path does not start with a /, the path is considered to be relative to the directory from which AMC is run. AMC must have read permissions to the given path.
Next	Click this button to read the configuration archive. If the configuration archive is valid, AMC loads the next page.

The Configuration Information page contains the following elements:

Element	Description
Configuration Information	Displays the configuration file path, the date and time when the configuration file was exported from source AMC and source AMC identity (hostname or IP address).
Projects in Configuration File	Lists projects in the archive. You must select one or more of the displayed projects to import. The list also provides some basic information about the project like the project name, type (Standalone/Shared) and description.
Node Mapping File Path	Allows the user to specify the location of the node mapping file on the AMC's server. For example, `/tmp/node-mapping.xml`. If the given file path does not start with a /, the given path will be considered relative to the directory from which AMC is run. AMC must have read permissions to the path given.
Import Semantics	Allows the user to specify type of import: • **Replace**: the imported configuration replaces all resources in the selected projects with those in the archive. Any resources (of the selected project) that are currently in the AMC but not in the archive are deleted. • **Add-on**: Imports new resources and changes to existing resources. Unlike Replace, existing resources are not deleted even if they are not found in the archive; instead, a warning is generated for each such resource
Import	Imports the selected project into AMC.

After you click the Import button, the subsequent page reports either success or failure of the import operation, along with error, warning, or informational messages, if any.

Programmatic Management Interface

The programmatic management interface feature (PMI) provides an interface so that third-party applications can manipulate data in AMC. Using the appropriate APIs, you can configure an application to:

- Log in to AMC.
- Display a list of projects.
- Export projects
- Import projects
- Display a list of deployment requests (DRs)
- Deploy to nodes
- Export a list of users and their permissions
- Import a list of users and their permissions

Any client that uses these APIs to interact with AMC, must adhere to the following restrictions:

- The client must use HTTPS to connect to AMC.
- The client must have a server certificate in the AMC trust store to properly authenticate.
- The client must use the "aonsadmin" user ID to log into AMC.
- All web service calls must be RPC based.
- SOAP version 1.1 is supported. Version 1.2 is not supported.
- All error messages are handled as SOAPFault. The client is expected to process SOAPFaults appropriately.

Sample Promotion Workflow

Using the schemas and APIs documented in Appendix A, "AON Schemas," a custom application might promote resources from one AMC to another as follows:

1. The client application invokes Login API to gain access to AMC. It then invokes the Export API. AMC gathers up all of the golden resources belonging to the project, and exports them in a configuration archive.
2. The client invokes a script that opens the configuration archive and modifies environment-dependent parameters in various resources. When the changes are complete, the modified configuration archive is written back to disk.
3. The client tool logs into the staging AMC and calls the Import API. This operation reads the configuration archive and replaces all existing resources of a specified project in the staging tier with the contents of the archive. The staging AMC calculates all of the add, modify, and delete changes needed in each AON node to accomplish the replacement. These changes are placed in the project's global and node deployment requests.
4. The client invokes the Deploy API of the staging AMC. This pushes all of the project's deployment requests to the appropriate nodes.

For a complete list of APIs and information on the sample PMI client included with AMC 2.4, see the "Programmatic Management Interface APIs" section on page A-15.

CHAPTER 2

Working with Nodes

Nodes are the devices that perform the actual application-oriented networking in an AON environment. Nodes are primarily managed by AMC, but they also have a command-line interface (CLI) through which some features can be configured. Additionally, nodes have the capability to be configured to operate in stand-alone mode, enabling third party tools to perform management functions previously reserved for AMC.

This chapter includes the following topics

- Managing Nodes, page 2-1
- Managing Virtual Clusters, page 2-8
- Managing WCCP Servers, page 2-7
- Configuring Recovery, page 2-12
- Deploying to Nodes, page 2-12
- Configuring SNMP, page 2-13
- Configuring Syslog, page 2-15

Note You must have System Administrator or Network Administrator privileges to perform most of the tasks described in this chapter. Deploy and monitor tasks are also visible to some other users. See the "Assigning Roles to Users" section on page 5-3 for further details.

Managing Nodes

Nodes are the individual devices that process messages in an AON environment. After being configured for basic network connectivity, a node must be configured to register with an AMC. On receipt of proper credentials, the AMC assumes control of the node.

Note A node can also be configured to operate in standalone mode. See the "Configuring a Stand-Alone Node" section on page 2-5 for details.

From the perspective of the AMC, nodes exist in one of the following states.

- **Unregistered**—Node created in the AMC, but no successful establishment of a trust relationship with AMC.
- **Registered**—Node successfully established a trust relationship with AMC.

- **Active**—Node activated by the administrator. Active nodes are able to receive deployment requests and process messages.
- **Inactive**—Formerly active node that has gone offline.
- **Replaced**—Node replaced by another node. During replacement, the new node assumes all processing responsibilities of the node being replaced. Replaced nodes cannot be activated again, nor can they be further configured by an administrator.
- **Reachable**—AMC can contact the node.
- **Unreachable**—A networking issue is preventing AMC from contacting the node.
- **Unknown**—AMC is unable to determine if the node is reachable.

This section covers the following topics:

- Creating New Nodes, page 2-2
- Configuring WCCP for Traffic Redirection, page 2-4
- Editing Nodes, page 2-4
- Deleting Nodes, page 2-5
- Replacing Nodes, page 2-5
- Configuring a Stand-Alone Node, page 2-5
- Configuring a Node for Use with TACACS+, page 2-6

Creating New Nodes

This section describes the procedure for creating a new AON node. To complete this procedure, you need access to the command-line interface of the node you are adding, and you need administrator access to AMC.

How to Get There

- Go to **Network > Network Nodes > Manage**, then click the **New** button.

Prerequisites

- AMC must be installed and running, and you must have appropriate privileges to create network nodes.
- Your node must be configured for basic IP network connectivity.

Step 1 Connect to the command-line interface of the AON node. Use the **show version** command to obtain the module serial number (highlighted below).

```
aon-node> show version
CPU Model:                    Pentium III (Coppermine)
CPU Speed (MHz):              498.675
CPU Cache (KByte):            256
Chassis Type:                 C2691
Chassis Serial:               12345678901
Module Type:                  Cisco 2600/3700/ISR AON Module (NM-AON-K9)
Module Serial:                FOC082313YY
AON:                          2.1.0.135
AMA:                          2.1.0.135
```

Note the sample serial number in bold text above. You will need the serial number from your node to complete Step 3.

Step 2 Log in to AMC and Go to **Network > Network Nodes > Network Nodes > Manage** to load the Manage Network Nodes page. Click the **New** button to load the New Network Node page.

Step 3 Complete the entries on this page as described in Table 2-1.

Table 2-1 New Network Node Entries

Entry	Description
Name	Name of your choosing for this node.
Serial Number	Enter the serial number obtained in Step 1.
Description	Optional entry.
Enable Node Polling	Enable polling when AMC a firewall is between AMC and the node. Rather than waiting for the node to contact AMC, AMC will initiate contact with the node.
Agent Hostname	Name or IP address of the node.
Agent Port	Port used by node for management traffic.

Step 4 Click Save to create the network node. The new node is in the Unregistered state and remains in this state until you configure the AON module to communicate with the AMC in the next step.

Step 5 In Configuration Terminal mode on the AON module, create an AON configuration. This configuration enables the AON node to register with the AMC.

```
aon-node> configure terminal
Enter configuration commands, one per line.  End with exit.
aon-node(config)> aon config configuration_id create
aon-node(config)> aon config configuration_id ama host module_IP_address
aon-node(config)> aon config configuration_id amc host AMC_IP_address
aon-node(config)> aon config configuration_id activate
aon-node(config)> exit
CAUTION!! Configuration changed. Need to restart AONS.
Confirm restart[y]? y
graceful restart[y]? n
Start counting down before restart


This may take a while longer...
```

Step 6 After the module restarts, use the **write memory** command to save the configuration.

```
aon-node> write memory
```

Step 7 In your browser window, click the browser's **Reload** button to refresh the Manage Network Nodes page. The new node should now be registered.

Tip If your network node remains unregistered, verify that the serial number is entered exactly as described in Step 3. The AMC will not establish trust with a node if this information is incorrect.

Step 8 Click the **Activate/Deactivate** link to load the Activate/Deactivate Network Nodes page, then click the radio button for the registered node. Click the **Activate** button.

When the state changes to Active, the node is ready for configuration deployment.

Note You can make configuration changes to a node in the registered or unregistered state, however, you cannot deploy those configuration changes until the node becomes active.

Configuring WCCP for Traffic Redirection

AON nodes can be configured to use WCCP for traffic redirection. When this feature is configured, a node can intercept messages using a specific port, then redirect them to another destination for further processing.

How to Get There

- Go to **Network > Network Nodes > Network Nodes > Configure**. Select a node, then click the **WCCP for Traffic Redirection** button and click **New**.

Table 2-2 shows the entries available on the New WCCP Service Group page.

Table 2-2 New WCCP Service Group Entries

Entry	Description
Service Group ID	Unique number for each service group. Range is 51 – 99.
Multicast address	IP address to be used by members of this service group.
Authentication password	Password by members of this service group for authentication.
Port map	Comma-delimited string of destination ports to be redirected.
Listener port	Comma-delimited string of ports at which an adapter is listening for traffic.
Protocol	Choose TCP or UDP from the drop-down list. **Note** You must configure the protocol on AMC. AON nodes do not support the use of the command-line interface to configure protocol.

Editing Nodes

The AMC enables you to edit the name and description of any node. If a node is unregistered, you can also change the serial number.

How to Get There

Go to **Network > Network Nodes > Manage** then select a node and click the **Edit** button.

Actions to Take

You can take one of the following actions:

- Make changes to the Name or Description. If a node is unregistered, you can also make changes to the serial number.
- Click the **Save** button to preserve your changes.
- Click the **Cancel** button to return to the Manage Network Nodes page.

Deleting Nodes

You can delete any node, regardless of its state. If a node is active, the AMC instructs the node to stop message processing before it is deleted.

How to Get There

Go to **Network > Network Nodes > Manage**, then select a node and click the **Delete** button.

Actions to Take

You can take one of the following actions:

- Click the **Yes** button to delete the node.
- Click the **No** button to cancel deletion and return to the Manage Network Nodes page.

Replacing Nodes

You can replace a registered node with another registered node. Active and unregistered nodes cannot be replaced, while active, inactive, and unregistered nodes cannot serve as replacements. After a node has been replaced, you can no longer change its configuration in the AMC, nor can you activate it for message processing. The replacement node inherits the exact configuration of the node being replaced, and you are then able to activate it for message processing.

How to Get There

Go to **Network > Network Nodes > Network Nodes > Manage**. Click the radio button for the node you want to replace, then click the **Replace** button.

Actions to Take

You can take one of the following actions:

- Click the radio button for the node that is to serve as the replacement, then click the **Submit** button to save your change.
- Click the **Cancel** button to discard your change and return to the Manage Network Nodes page.

Configuring a Stand-Alone Node

In environments where a third-party management application, such as AlterPoint, will manage AON nodes, each node must be configured to operate in stand-alone mode. This mode enables a node to operate without the AON Management Console, and it enables the node to receive all required configuration input from the command-line interface (CLI).

This feature also provides the ability to use the CLI to configure four different adapters (http, aonp, jms, and pmode). Previously these adapters required AMC's Web interface for configuration.

Sample Configurations

The following example shows a node being configured for stand-alone mode. It also shows the commands to configure the promiscuous mode (Pmode) adapter.

```
aon-sm-1(config)> aon standalone
aon-sm-1(config)> adapter pmode
```

```
aon-sm-1(config-adapter)>  domain PmodeAdapter
aon-sm-1(config-adapter-domain)>   propertyset default
aon-sm-1(config-adapter-domain-propertyset)> set “Default Destination IP" "10.235.1.11"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Sampling Interval" "10"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Sampling Duration" "10"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Sampling Mode" "false"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Destination Port" "10001"
aon-sm-1(config-adapter-domain-propertyset)> exit propertyset
aon-sm-1(config-adapter-domain)> $exit domain
aon-sm-1(config-adapter)> $exit adapter
```

The following example shows the installation and configuration of a Pmode adapter extension:

```
aon-sm-1 aon install extension url http://10.0.0.1/RdfAdapterExtPackage.jar
aon-sm-1 configuration terminal
aon-sm-1(config)> adapter pmode
aon-sm-1(config-adapter)> domain PmodeAdapter
aon-sm-1(config-adapter-domain)> propertyset default
aon-sm-1(config-adapter-domain-propertyset)> set “Default Destination IP" "10.235.1.11"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Sampling Interval" "10"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Sampling Duration" "10"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Sampling Mode" "false"
aon-sm-1(config-adapter-domain-propertyset)> set “Default Destination Port" "10001"
aon-sm-1(config-adapter-domain-propertyset)> exit propertyset
aon-sm-1(config-adapter-domain)> exit domain
aon-sm-1(config-adapter)> domain PmodeAdapterExtension
aon-sm-1(config-adapter-domain)> propertyset rdflink
aon-sm-1(config-adapter-domain-propertyset)> set “ExtensionLink" "RDF-FRAMING-EXTN-1"
aon-sm-1(config-adapter-domain-propertyset)> exit propertyset
aon-sm-1(config-adapter-domain)> exit domain
aon-sm-1(config-adapter)> domain RdfExtension
aon-sm-1(config-adapter-domain)> propertyset rdftraffi
aon-sm-1(config-adapter-domain-propertyset)> extension RDF-FRAMING-EXTN-1
aon-sm-1(config-adapter-domain-propertyset-extension)>set "MonitorPort" "10002"
aon-sm-1(config-adapter-domain-propertyset-extension)>set "AdapterExtPolicyLink” "rdflink"
aon-sm-1(config-adapter-domain-propertyset-extension)>set "MonitorMask" "255.255.255.255"
aon-sm-1(config-adapter-domain-propertyset-extension)>set "MonitorAddress" "10.235.1.11"
aon-sm-1(config-adapter-domain-propertyset-extension)>exit extension
aon-sm-1(config-adapter-domain-propertyset)> exit propertyset
aon-sm-1(config-adapter-domain)> exit domain
aon-sm-1{config)> exit adapter
```

Configuring a Node for Use with TACACS+

When a TACACS+ server is configured, a node provides the following functionality:

- Users authenticated against the TACACS+ server when they log in.
- The node will verify each command entered by a user before executing it. If a user does not have permission to use a command, the command is not executed.
- The user named “admin” is a local user. This user can successfully log in when the TACACS+ server is unavailable. The “admin” user has access to all commands on the node.
- You can enter up to three TACACS+ servers. If the first server is not found, the node will contact the second server. If the first two servers are not found, the node will contact the third server. If the first server denies authentication to the user, the node does not contact the other two servers.
- You can use the **tacacs-server key** command to enter an encryption key. The default for this optional command is unencrypted communication with the TACACS+ server.

- You can use the **tacacs-server port** command to specify the port used by the TACACS+ server. The default for this optional command is port 49.
- You can use the **tacacs-server timeout** command to specify the number of seconds the node is to wait for response from the TACACS+ server. The default for this optional command is 5 seconds.

Note This feature is supported on the AON Appliance, AON-SM, and AON-NME. The AON-NM does not support TACACS+.

The following example shows the configuration of a three TACACS+ servers on an AON node. Note that in this example, only the first command is required to configure TACACS+. The remaining commands are optional.

```
aon-sm-1(config)> tacacs-server host 10.10.10.1
aon-sm-1(config)> tacacs-server host 10.10.10.2
aon-sm-1(config)> tacacs-server host 10.10.10.3
aon-sm-1(config)> tacacs-server key encryption-key
aon-sm-1(config)> tacacs-server port port-number
aon-sm-1(config)> tacacs-server timeout seconds
aon-sm-1(config)> exit
```

The following example shows a sample configuration on a TACACS+ server for a user named “user123.” In this example, the user can use only the “**show**” commands. Use of any other commands by this user yields an “Authorization Failure” error.

```
user = user123 {
  login = cleartext "user123"
    cmd = show {
     permit version
    }
}
```

Managing WCCP Servers

A WCCP server is a router that redirects traffic to an AON node. A WCCP Server can also be used for load balancing. By configuring a WCCP server, you provide the AMC with the information that it uses to contact the server and configure it for traffic redirection or load balancing.

This section covers the following topics

- Creating WCCP Servers, page 2-7

Creating WCCP Servers

How to Get There

Go to **Network > Network Nodes > WCCP Servers > Define WCCP Servers**, then click the **New** button.

Note You must be in the System Project to configure WCCP, ACL/Classifier, or Recovery properties. If you have opened another project, you can only view these properties.

Table 2-3. shows the entries available on the New WCCP Server page.

Table 2-3 New WCCP Server Entries

Entry	Description
IP Address	IP address of the switch or router being configured.
User name	Username required to configure device.
Password	Password required to gain access to device.
Enable password	Enable password required to access privileged EXEC mode.
Access method	If the device is configured for SSH, select secure shell. Otherwise select telnet.

Note AON uses Base64 to mask passwords entered during WCCP configuration.

Managing Virtual Clusters

A virtual cluster is a set of identically configured network nodes. After nodes are added to a virtual cluster, you can update the entire clustered group by changing a single set of configuration parameters. Virtual clusters can be configured for the following:

- High availability—Nodes in a cluster can function as a single node. When a node is taken out of service, the other nodes in that virtual cluster assume the messaging processing responsibilities of the missing node.
- Load balancing—Nodes in a cluster can share workload, meaning no single node becomes overloaded with network traffic.

This section covers the following topics:

- Creating a Virtual Cluster, page 2-8
- Changing Nodes Within a Virtual Cluster, page 2-9
- Configuring WCCP for Cluster Management, page 2-10

Creating a Virtual Cluster

A virtual cluster consists of two or more AON nodes that are configured to share workload and ensure redundancy. The first node you choose for a cluster is called the master node. Other nodes that you add to the cluster will receive duplicate configurations to that of the master node. After the virtual cluster has been created, all nodes are equal, meaning no node is a master node.

If you create a virtual cluster that consists of nodes assigned to one or more projects, the following occurs:

- If a node is to become the master node of the virtual cluster, it is removed from any projects to which it is assigned. The new virtual cluster is automatically assigned to those projects.
- If the node is not the master node, it is automatically removed from any projects to which it is assigned. The new virtual cluster is not assigned to those projects.

Note You must be in the System Project to configure WCCP, ACL/Classifier, or Recovery properties. If you have opened another project, you can only view these properties.

Prerequisites

- You need at least two registered nodes. Nodes cannot be unregistered while they are being added to a virtual cluster. The master node can be active, however, the nodes being added must be in the registered state.
- All nodes in a cluster must be running on the same type of hardware. You cannot, for example, combine an AON-SM and AON-NM into a virtual cluster.

Step 1 Go to **Network > Network Nodes > Virtual Clusters > Create**. This loads the Create Virtual Cluster page.

Step 2 Select a master node (the node whose configuration will be duplicated on the other nodes in the cluster) and click the **Next** button. This loads the Create Virtual Cluster page.

Step 3 Complete the entries as appropriate for your network and select the other nodes to be added to the cluster.

Step 4 Click the **Finish** button to save your changes. A dialog is displayed giving you a final opportunity to create the virtual cluster or cancel the operation.

Step 5 Go to **Network > Network Nodes > Virtual Clusters > Manage** to verify that the cluster was configured.

Step 6 Go to **Network Nodes > Activate/Deactivate** to make the nodes in the cluster Active.

Changing Nodes Within a Virtual Cluster

After a virtual cluster is configured, you can perform any of the following actions:

- Add Nodes—When you add additional nodes, the new nodes receive identical configuration to that of the existing nodes in the cluster. If you add a node that is assigned to one or more projects, that node is removed from those projects. The virtual cluster is not assigned to those projects
- Remove Nodes—If you remove a node from a cluster, it is returned to the registered state. Remaining nodes in the cluster continue to operate in the absence of the removed node. The configuration of a node that is removed from a cluster is restored to the factory default when that node is activated outside of the cluster. Nodes removed from a virtual cluster are not assigned to any project.
- Delete—If you delete a cluster, all member nodes are returned to the registered state, and their configurations are restored to the factory default. After a cluster is deleted, the member nodes are not assigned to any project.

Note You must be in the System Project to configure WCCP, ACL/Classifier, or Recovery properties. If you have opened another project, you can only view these properties.

Configuring WCCP for Cluster Management

AON nodes use WCCP to detect when a member of a cluster goes offline. If this happens, other members of the cluster assume the missing node's message processing workload.

Note You must be in the System Project to configure WCCP, ACL/Classifier, or Recovery properties. If you have opened another project, you can only view these properties.

Prerequisites

- You must have a WCCP server available to add to the virtual cluster before beginning this configuration. See the "Managing WCCP Servers" section on page 2-7 to configure a WCCP server.

Table 2-4 shows the entries available on the New WCCP Service Group page.

Table 2-4 New WCCP Service Group Entries

Entry	Description
Service group ID	Unique number for each service group. Range is 51 – 99.
Multicast address	IP address to be used by members of this service group.
Authentication password	Password by members of this service group for authentication.

Step 1 After completing the entries, click the **Add Servers** button. This loads the a page that lists available WCCP servers.

Step 2 Choose one or more servers, then click the **Add** button. The servers are added to the WCCP service group.

Step 3 Click the **Configure Interfaces** button to specify the interface to be used by the WCCP server. This loads the Server Interfaces page.

Step 4 Enter the names, such as Service-Engine1/0, of the interfaces to be used by members of the service group, then click the **Save** button. After you are returned to the New WCCP Service Group page, click the **Save** button to save the entire service group configuration.

Configuring WCCP for Traffic Redirection

AON nodes use WCCP to for traffic redirection and load balancing. You can configure nodes to redirect messages based on the IP address or port.

How to Get There

- Go to **Network > Network Nodes > Configure**, then select a node and click the **WCCP for Traffic Redirection** button.

Prerequisite

- If traffic redirection is to be based on source or destination IP addresses, you must configure an ACL/Classifier for the cluster. See the "Configuring ACL/Classifiers" section on page 2-11 to specify IP address parameters for traffic redirection.

Table 2-5. shows the entries available on the New WCCP Service Group page.

Table 2-5 New WCCP Service Group Entries

Entry	Description
Service group ID	Unique number for each service group. Range is 51 – 99.
Multicast address	IP address to be used by members of this service group.
Authentication password	Password by members of this service group for authentication.
Port map	Comma-delimited string of destination ports to be redirected.
Listener port	Comma-delimited string of ports at which an adapter is listening for traffic.
Protocol	Choose TCP or UDP from the drop-down list. **Note** You must configure the protocol on AMC. AON nodes do not support the use of the command-line interface to configure protocol.

Step 1 Complete the entries as appropriate for your network, then click the **Add Servers** button. This loads a page that lists available WCCP servers.

Step 2 Choose one or more servers, then click the **Add** button. The servers are added to the WCCP service group.

Step 3 Click the **Configure Interfaces** button to specify the interface to be used by the WCCP server. This loads the Server Interfaces page.

Step 4 Enter the name of the interfaces to be used by members of the service group, then click the **Save** button.

Step 5 After you are returned to the New WCCP Service Group page, click the **ACL/Classifier** button. On the next page, click the **Add Entries** button to load the page that lists the available ACL/Classifiers.

Step 6 Choose an ACL/Classifier, then click the **Select** button to associate it with the WCCP service group.

Step 7 Click the **Save** button to save your changes and return to the New Service Group page. From there click the **Save** button to complete the configuration.

Configuring ACL/Classifiers

An ACL/Classifier contains an ordered list of access control entries. Each entry contains a source and destination IP address that are matched against the contents of a packet to determine if messages are to be redirected by WCCP. ACL/Classifiers are also used for message classification.

Note You must be in the System Project to configure WCCP, ACL/Classifier, or Recovery properties. If you have opened another project, you can only view these properties.

Step 1 Use one of the following navigation paths:

- For network nodes; **Network > Network Nodes > Configure**. Select a node, then click the **ACL/Classifier** button.

- For virtual clusters: **Network > Virtual Clusters > Configure**. Select a cluster, then click the **ACL/Classifier** button.

This loads the New ACL/Classifier Entry page.

Step 2 Complete the entries as required by your environment, taking the following into consideration:

- A 0 (zero) wildcard bit equates to "only"
- A 255 wildcard bit equates to "any"

Step 3 Click the **Save** button to save your changes.

Configuring Recovery

The AMC enables you to control the recovery parameters of network nodes and virtual clusters. Watchdog is a process that runs on an AON node and verifies that the AON application on that node is operating normally. When watchdog detects a failure, it can attempt to restart AON and WCCP.

How to Get There

- Network node: Go to **Network > Network Nodes > Configure.** Select a node and click the **Recovery** button.
- Virtual cluster: Go to **Network Nodes> Virtual Clusters > Configure**. Select a node and click the **Recovery** button.

Note You must be in the System Project to configure WCCP, ACL/Classifier, or Recovery properties. If you have opened another project, you can only view these properties.

Table 2-6. shows the entries available on the Recovery page.

Table 2-6 Recovery Entries

Entry	Description
AON Heartbeat Interval	Rate at which the AON process sends heartbeats to the watchdog process.
AON Startup Delay	Number of seconds watchdog waits for the AON process to start up before attempting to restart.
Watchdog Recovery Action	Action to be taken when a watchdog timer expires.
WCCP "Here I Am" Interval	Interval at which WCCP clients send the "Here I Am" message.
Enable Watchdog	Drop-down list to select if watchdog is enabled or disabled.
Watchdog Failure Detection Interval	Time that will elapse before watchdog detects that AON is down.

Deploying to Nodes

Changes made to the configuration of an AON node must be explicitly deployed to the node. These changes include those made in AMC and those uploaded from the AON Development Studio. Whenever a configuration change is made, it appears in a deployment request (DR). There are two types of deployment requests:

- Global Deployment Request—contains changes, such as a global properties, that apply to all nodes in a project.
- Node Deployment Request—contains changes, such a new PEPs or message types, that apply to an individual node.

To deploy changes to nodes, perform the following steps:

Step 1 **Go to Deployment > Manage Staging** to view the deployment requests waiting in the Open and Staged state.

Step 2 Click the radio button for the deployment request, then click the **Stage** button. This changes the state to Staged, which is the last stop before deployment.

Step 3 Click the Manage Deployment link, which loads the Manage Deployment page.

Step 4 Click the radio button for the deployment request, then click the **Deploy** button. The AMC deploys the request to the AON node.

Step 5 Click the Summary Link to verify that the request was successfully deployed.

How to Get There

Go to **Monitor > Logs**, then select a node and click the **View Logs** button.

How to Get There

Go to **Monitor > Events**, then select a node and click the **View Events** button.

Configuring SNMP

To enhance the manageability of AON, you can configure AON nodes to transmit simple network management protocol (SNMP) messages to network management applications. AON notifications are event-driven, rather than requiring a monitoring application to poll each node.

The table that follows lists the commands supported by the AON SNMP feature.

Command	Description
snmp-server community *string* **[ro | rw]**	Enables SNMP and sets the community string. Use **ro** to specify read-only access for management stations; use **rw** to specify read-write access.
snmp-server contact *text*	Sets the system contact (sysContact) string.
snmp-server host *ip-address* **traps version [1 | 2]** *community-string*	Defines host and version of the recipient of SNMP messages.
snmp-server location *text*	Sets the system location string (sysLocation).
show snmp configuration	Displays the current SNMP configuration for the node.

The sections that follow list the MIBs supported by AON:

- Industry Standard MIBs, page 2-14

- Cisco Standard MIBs, page 2-14

To translate MIBs, use the Cisco SNMP Object Translator.

Industry Standard MIBs

SNMPv2-MIB

- Entire MIB, including coldStart trap

IF-MIB

- ifTable

IP-MIB

- ip objects
- ipAddrTable

SYSAPPL-MIB

- sysApplInstalledPkgTable
- sysApplRunTable

HOST RESOURCES-MIB

- hrSystemNumUsers
- hrSystemProcesses
- hrMemorySize
- hrStorageTable
  - hrStorageDescr
  - hrStorageAllocationUnits
  - hrStorageSize
  - hrStorageUsed

Note In AON, hrStorageTable contains two entries. The first entry denotes the RAM in the system, and the second entry denotes the disk partition.

Cisco Standard MIBs

CISCO-PROCESS-MIB

- cpmCpuTotalTable
- cpmProcessTable

CISCO-SYSLOG-MIB

Configuring Syslog

AON nodes include the capability to forward log messages to syslog servers. Up to four syslog servers can be configured for each AON node, and each host can use a unique priority and rate-level setting.

The table that follows lists the commands supported by the AON SNMP feature.

Command	Description
logging host *ip-address* **priority** *priority-level* [**rate-limit** *bytes-per-second*]	Configures the IP address of the recipient of syslog message and one of the following priority levels: • **alert**—immediate action needed • **critical**—critical conditions • **emergency**—system is unusable • **error**—error conditions • **warning**—warning conditions The default priority level is warning. To control the bandwidth used for syslog messages, use the **rate-limit** keyword to specify the bytes per second. The default rate-limit is 0.
show logging	Displays current logging and syslog server configuration for the node.

CHAPTER 3

Managing AON Properties

Properties control how messages are processed in an application-oriented network. Properties can be applied globally to the entire AON environment, or they can be applied only to individual nodes.

Note Access to items on the Properties tab may be limited depending on the privileges assigned to you. For further details, see the "Assigning Roles to Users" section on page 5-3. Also, you must open a project to gain access to the Properties tab in AMC.

This chapter includes the following sections:

Note This chapter covers most properties that appear on the Properties tab of the AMC. Additional AMC properties related to security, authentication, and authorization are in Chapter 4, "Managing AON Security."

Monitoring Activity

Bladelet Monitoring Property

The Bladelet Monitoring Property configures which events are stored for retrieval using the screen at **Monitor > View Events**. You can configure this property globally, or you can apply it to individual nodes.

How to Get There

Open a project, then go to **Properties > Monitoring.**

Action to Take

To configure the Monitoring Property, change events that you want monitored to True, then click the **Submit** button.

Message Log Domain

AON nodes are able to capture application log messages and store them in a database for later retrieval. This functionality requires you to complete the following tasks:

1. Create a Message Log Database—This is the Oracle or Sybase database in which log messages are to be stored.
2. Configure Message Log Domain Property—This defines within AMC the database configuration details to be used to store log messages.

Upon completion of these steps, ADS users are able to use the Log bladelet to store messages in the database.

Create a Message Log Database

If you enable AON message logging, you can configure an external Oracle or Sybase database to store log messages. An existing Oracle database can be used for message logging. However, a Sybase database must have a specific configuration to be compatible with AON. For this reason, we recommend that you create a new database.

Step 1 Create a database and a user (for logins). Grant the user database privileges to create, query, delete, update, and insert.

Use one of the following for the Message Log Database:

- Oracle 9i (9.2)

  You can create a separate Oracle 9i database for AON Message Logging.

- Sybase 12.5.1

  You should create a separate Sybase 12.5.1 Adaptive Sever (database) for AON message logging, The requirements for this external database are summarized below.

  - Page size >= 8K
  - Procedure cache size - 100000

- Max memory 131072 (in 2k units, i.e. 131072 * 2k = 256MB)

Note See Oracle or Sybase documentation for specific database configuration instructions.

Step 2 Run the appropriate script to create the Message Log schema in your database. See Appendix A, "AON Schemas" for Sybase and Oracle scripts.

Configure Message Log Domain Property

After a database as been configured, you can configure Message Log Domain Property. This is a device level property.

How to Get There

Open a project, then go to **Properties** > **Application > Node**. Select a node, then click the **Edit Properties** button.

Data to Enter

The Message Log Domain Property page includes the entries described in Table 3-1.

Table 3-1 Message Log Domain Property Entries

Entry	Description
Name	Name of your choosing for this property.
Enabled	Select true to enable, false to disable.
Sub-protocol	**oracle.thin** or **oracle.oci** for Oracle. **sybase.Tds** for Sybase.
User ID	User ID required to log on to the database. The user must have permission to create, read, write, update, and query the database.
Password	The password to gain access to the database.
Database alias	Alias pointing to the database. This value depends on the configuration of the database. The format for this entry is *<IP address>* **:** *<port>* **:** *<name of database>* The following are examples: • Oracle—**@10.1.1.1:1521:aonmlog** • Sybase—**10.1.1.2:5000/aonmlog** The last part in the alias is the name of the database instance. The message log schema should be provided by your database administrator.
Driver	The JDBC driver name. AON supports the following two drivers: • Oracle—**oracle.jdbc.OracleDriver** • Sybase—**com.sybase.jdbc2.jdbc.SybDriver**
Max Queue Size	Maximum size of the Message Log queue.

Adaptive Load Balancer

Adaptive Load Balancer is used to change the adaptive load balancing algorithm used by AON.

How to Get There

Open a project, then go to **Properties** > **Application > Global**.

Data to Enter

The Adaptive Load Balancing property page includes the entries described in Table 3-2.

Table 3-2 Adaptive Load Balancer Entries

Entry	Description
Name	Name of your choosing for this property.
Maximum Request Discard	Number of requests to wait before discarding a server's average response time data.
Maximum Response Samples	Number of samples used for determining the most responsive server.

Adjusting Quality and Performance

AON allows you to measure and control runtime control quality and performance for message types that you specify.

Caching

AON includes a built-in cache engine that can be used as a proxy cache or reverse proxy cache depending on where and in which administrative domain the cache is placed. Use the Caching Property to configure how the AON cache engine operates. This is a device-level property, and it is used in conjunction with PEPs that include the CacheData and RetrieveCache bladelets.

How to Get There

Open a project, then go to **Properties** > **Application > Node**. Select a node, then click the **Edit Properties** button.

Data to Enter

The Caching Property page includes the entries described in Table 3-3.

Table 3-3 Entries on Caching Property

Entry	Description
Override no-cache Response Directive	If this value is set to **true**, the HTTP “no-cache” response directive is ignored.
Override no-store Response Directive	If this value is set to **true**, the HTTP “no-store” response directive is ignored.
Override private Response Directive	If this value is set to **true**, the HTTP “private” response directive is ignored.
Override no-cache Request Directive	If this value is set to **true**, the HTTP “no-cache” request directive is ignored.

Table 3-3 Entries on Caching Property (continued)

Entry	Description
Override no-store Request Directive	If this value is set to **true**, the HTTP “no-store” request directive is ignored.
Override Pragma:no-cache Request Directive	If this value is set to **true**, the HTTP “Pragma:no-cache” request directive is ignored.
Response Cache Default TTL	Default time to live (TTL) to be used for response caching.
Variable Cache Default TTL	Default TTL to be used for variable caching.
Max Objects Variable Cache	Determines the number of objects to store in the variable cache before replacement algorithms are activated.
Max Objects Security Cache	Maximum number of objects to be cached in the security cache before replacement algorithms are activated.
Response cache replacement Algorithm	This value must be set to **LRU**. This is the replacement algorithm to be used for response caching.
Variable Cache Replacement Algorithm	This value must be set to **LRU**. This is the replacement algorithm to be used for variable caching.
Security Cache Replacement Algorithm	Must be set to **LRU**. This is the replacement algorithm to be used for security caching.
Cache Server	Host name or IP address of the caching server. Must be set to localhost.
Cache Server Port	Port on which the caching server listens. Must be set to 60606.
Connection Timeout	Determines how long a request will wait for a response
Queue Size	The pending message queue contains references to messages that are awaiting response from the server. If a message remain in this queue beyond the timeout value, the server is assumed to be down. Typically set for 20–30 seconds.
Polling Interval	Determines the number of seconds the client will wait before checking if a failed server has returned to service.
Pending Message Queue Timeout	Determines the size of the client’s sending queue.
Timed-out Message Count	Determines how many failed messages are required for a server to be considered down.

Application QoS

The Application QoS feature enables AON to prioritize message processing based on the differentiated services code point (DSCP) contained in the IP header. Use the QoSMapping page in AMC to define appropriate DSCP values for the following categories (listed in priority order):

- Bulk data transfer
- Default
- Mission critical
- Network management
- Transactional data

These categories are available to PEP developers who use the Application QOS bladelet.

How to Get There

Open a project, then go to **Properties > Application > Global**, then select QoSMapping.

Actions to Take

Click the radio button for the property set you want to change, then click the **Edit** button. On the screen that follows, enter the new DSCP value and click the **Submit** button.

Fastpath

AON enables optimization of some PEPs, using a specialized process called Fastpath. With Fastpath, AON can optimally process messages that are classified with PEPs and meet specific conditions. This optimization substantially increases throughput. For more information on configuring PEPs for FastPath, see the AON Programming Guide.

How to Get There

Open a project, then go to **Properties > Application > (Global or Node) > Fastpath**

Actions to Take

Use this page to change the HTTP port on which the node listens for messages to be processed by Fastpath. The default port is 5556.

Working with Message Content

AON allows you to work with the content of your messages based on properties that you set.

Content Parser

The content parser property specifies a Java class that implements a content parser to use for reading an input content and converting it to an equivalent XML content. This property can also specify a Java class to use to perform the transformation instead of using XSLT-based transformation.

How to Get There

Open a project, then go to **Properties** > **Application > Node**. Select a node, then click the **Edit Properties** button.

Data to Enter

The Content Parser Property page includes the entries described in Table 3-4.

Table 3-4 Entries on Content Parser Property

Entry	Description
Name	Name of the Content Parser property.
Transformation Factory	This parameter specifies the class name that implements a custom transformer.

Table 3-4 Entries on Content Parser Property (continued)

Entry	Description
Parser Class Name	This parameter specifies the name of Java class that is used to parse the input message content and convert it to equivalent XML content.
Name of Package	Specifies the name of the transform package.

Content Validation

A Content Validation application property imposes an external schema on an XML message that contains no predefined grammar declarations. This property is used when input XML does not contain any grammar declaration (XSD or DTD) but is expected to conform to a receiver point schema. It is also used when Input XML is transformed within AON and is expected to conform to a target schema.

How to Get There

Open a project, then go to **Properties** > **Application > Node**. Select a node, then click the **Edit Properties** button.

Data to Enter

The Content Validation property page includes the entries described in Table 3-5.

Table 3-5 Entries on Content Validation Property

Entry	Description
Name	Name of the Content Validation property.
Target Schema Name	Target schema to be imposed on XML messages running a particular PEP.
Target Namespace	Namespace for the target schema named above.

Working with XSL Transformation

This property configures AON to perform XSL transformation (XSLT). The Transformation property determines the document style sheet, target content type, and transformation package. This property can be configured globally or for individual nodes.

How to Get There

Open a project, then go to **Properties** > **Application > (Global** or **Node** and select a node), then click the **Edit Properties** button.

Data to Enter

The Transformation page includes the entries described in Table 3-6.

Table 3-6 Transformation Property Entries

Entry	Description
Name	Name of the Transformation property.
Name of XSLT Stylesheet	Specifies the name of the transform file to use. The file must be present in the Transform Bundle specified by the parameters below.
Target Content Type	This is used to set the content type of the target content when the input content is stream content and its type is not known.
Transformation Factory	Choose an XSLT transformer to be used.
Name of Package	Specifies the name of the transform package.

Controlling Message Delivery

Message delivery properties define the delivery characteristics associated with a message type. All message types have a default delivery property, which is specified when you create the message type in the ADS. After a message is classified, the delivery properties of that message are dictated by the delivery property associated with that message type. Message delivery properties must be configured in the following order:

1. Configuring Send Properties.
2. Configuring Delivery Semantics.
3. Binding Message Delivery Properties to a Message Type.

After you configure Send Properties and Delivery Semantics, synchronize ADS with the AMC to begin using the new delivery properties with message types.

Configuring Send Properties

The Send Properties page specifies how long a message type should wait for a timeout.

How to Get There

Open a project, then go to **Properties** > **Application > (Global** or **Node**). Then select **Send Properties**.

Data to Enter

The Delivery Notification property page includes the entries described in Table 3-7.

Table 3-7 Delivery Connection Property Entries

Entry	Description
Name	Name of the Send Property.
Request Timeout	Length of time to wait for a response from the endpoint for a timeout, measured in milliseconds.
Retry Interval	Length of time to try re-sending a message.

Table 3-7 Delivery Connection Property Entries (continued)

Entry	Description
Retry Count	Number of times AON will attempt to resend the message.

Configuring Delivery Semantics

The Delivery Semantics property specifies delivery properties for a message type. Use this property in conjunction with the Send Properties page to configure the delivery of messages.

Note You must perform the actions described on the Configuring Send Properties page before you can configure Delivery Semantics.

How to Get There

Open a project, then go to **Properties** > **Application >** (**Global** or **Node** and select a node). Then select **Delivery Semantics**.

Data to Enter

The Delivery Semantics page includes the entries described in Table 3-8.

Table 3-8 Delivery Semantics Property Entries

Entry	Description
Name	Name of your choosing for this property.
Time to live	How long either request message or response message can stay in the system Specified in milliseconds.
Send Properties	Select a Send Property. See Configuring Send Properties for further details.

Actions to Take

Use the **Edit List** button to choose a delivery notification and connection property.

Binding Message Delivery Properties to a Message Type

After you configure message delivery properties in the AMC, the property is available to ADS users when they configure message types.

After you configure an Encoding profile, it is available to ADS users when they configure a message type.

Next Hop Domain

Next Hop Domain Property enables a device to forward all traffic using a specified protocol to a designated AON node. Next Hop Domain is a device-level property.

Note You must configure next hop domain in the System Project. Next hop domain will fail if you configure in another project.

Note In a two-node scenario, configure this property on the client proxy with the configuration details necessary to route messages to the server proxy.

How to Get There

Open the System project, then go to **Properties** > **Application > Node**. Select a node, then click the **Edit Properties** button.

Data to Enter

The Next Hop Domain Property page includes the entries described in Table 3-9.

Table 3-9 Entries on Next Hop Domain Property

Entry	Description
Name	Use the hostname or IP address of the destination and the port on which the host is listening for messages. *ip_address***:***port* or *hostname***:***port*
Address	IP address or hostname for next hop device.
Port	Port on which device is listening for next hop traffic.
Protocol	One of the following protocols: • **AONP-HTTP** • **AONP-TCP**
Mode	Choose **secure** for encrypted or **clear** for unencrypted.

Node Capabilities

The Node Capabilities property enables you to configure message delivery persistence on a node. Node Capabilities is a device level property.

Note If message delivery persistence is to be stored in a database, you must configure two databases before you configure this property. See the "Create a Message Log Database" section on page 3-2 for information on configuring a database.

How to Get There

Open a project, then go to **Properties** > **Application > Node**. Select a node, then click the Edit Properties button.

Data to Enter

The Node Capabilities property page includes the entries described in Table 3-10.

Table 3-10 Node Capabilities Property Entries

Entry	Description
Name	Name of your choosing for this property.
Persistence	Choose **off** to disable persistence. Choose database to **enable**
WCCP Service Group	Enter the WCCP service group for the virtual cluster configured for multi-blade message delivery.
Wait Timeout	Specified in milliseconds.
Multi-Blade Database	Click the Edit List button to choose an available Database

Working with Adapters

You can use AMC to control how adapters function within your AON implementation. You can also configure additional properties and extensions for each adapter. Adapters can only be configured in the System project. For more details about adapters, properties, and extensions, see the *AON Programming Guide.*

Adapter Registry

The Adapter Registry page enables you to manage the properties of both built-in and custom adapters. You can activate or deactivate an adapter, change the start-up mode, and change the protocol to be used by the adapter.

How to Get There

Open the System project, then go to **Properties > Adapter.**

Adapter Listener Domain

Adapter Listener Domain enables you to configure the listening parameters of an adapter. You can specify the port on which the adapter listens, and you can choose either clear or secure communication.

How to Get There

Open the System project, then go to **Properties > Application > Adapter Listener Domain.**

For more information about adapters, see the *AON Programming Guide*.

Service Profiles for Adapters

Service Profiles are used in conjunction with the development of custom bladelets and custom adapters. Available services include the following:

- Compression
- Content Lookup
- Content Validation

- Encryption
- Signature

Developers can create profiles, which are sets of attributes that describe how the services listed above are implemented in custom bladelets or adapters. Profiles contain multiple named contexts for a service, and these profiles must be created in AMC in order for developers to access these contexts by name.

For more details about custom bladelets, custom adapters, and external services, see the *AON Programming Guide.*

How to Get There

Open a project, then go to **Properties > Service Profiles.**

Working with Message Transport

Encoding

The Encoding property enables you to configure AON nodes to compress outgoing traffic. After you configure an encoding property, that property is available to ADS users. When message types are configured, each message type can be associated with an encoding property.

How to Get There

Open the System project, then go to **Properties** > **Application > Node**. Select a node, then click the **Edit Properties** button.

Data to Enter

The Encoding property page includes the entries described in Table 3-11.

Table 3-11 Encoding Property Entries

Entry	Description
Name	Name of your choosing for this property.
Request Encoding	Choose the encoding for the request portion of the PEP.
Response Encoding	Choose the encoding for the response portion of the PEP.

Configuring JMS Properties

Use JMS properties to configure the way AON nodes handle JMS messages. You must configure JMS properties in the following order:

1. JMS Destination Property, page 3-13
2. JMS Source Property, page 3-13
3. JMS Reply To, page 3-14
4. JMS Connections Property, page 3-14
5. JMS Naming Property, page 3-15

JMS Destination Property

The JMS Destination Property enables you to specify a new destination for JMS messages.

How to Get There

Open a project, then go to **Properties** > **JMS > Node**. Select a node, then click the **Edit Properties** button.

Data to Enter

The JMS Destination Configuration page includes the entries described in Table 3-12.

Table 3-12 JMS Destination Configuration Entries

Entry	Description
Name	Name of your choosing for this configuration.
Destination name	Name of the destination JMS broker.
Delivery Mode	Choose PERSISTENT or NON_PERSISTENT as appropriate.
Time To Live	Use the value specified by the JMS broker. This entry is required.
Priority	Use the value specified by the JMS broker. This entry is required.

JMS Source Property

The JMS Source Property Page enables you to specify a new source for JMS messages. It requires you to specify a JMS Destination, which you should have configured in the previous section.

How to Get There

Open a project, then go to **Properties** > **JMS > Node**. Select a node, then click the Edit Properties button.

Data to Enter

The JMS Source Configuration page includes the entries described in Table 3-13.

Table 3-13 JMS Source Configuration Entries

Entry	Description
Name	Name of your choosing for this configuration.

Table 3-13 JMS Source Configuration Entries (continued)

Entry	Description
Source Name	Name of the source sending JMS messages.
Message Selector	Enter a header entry or property reference that is to be used to identify messages of interest.
Delivery Failure Property	Undelivered messages are placed in the dead letter queue (DLQ) and committed.
Destination	Click the Edit List button to choose an available JMS Destination property.

JMS Reply To

The JMS ReplyTo property enables you to specify a new reply queue to be used by JMS clients.

How to Get There

Open a project, then go to **Properties** > **JMS** > **Node**. Select a node, then click the Edit Properties button.

Data to Enter

The JMS Reply To Property page includes the entries described in Table 3-14.

Table 3-14 JMS Reply To Property Entries

Entry	Description
Name	Name of your choosing for this property.
ReplyTo Name	Name of the ReplyTo queue.
Definition Type	Choose one of the following types of queues: • template • static • temporary
Message Delivery	Choose the appropriate type of delivery.
Batch Size	Size of batch count.
Number of Queues/Topics	Enter number of queues.

JMS Connections Property

How to Get There

Open a project, then go to **Properties** > **JMS** > **Node**. Select a node, then click the Edit Properties button.

Data to Enter

The JMS Connection Property page includes the entries described in Table 3-15.

Table 3-15 JMS Connection Configuration Entries

Entry	Description
Name	Name of your choosing for this connection configuration.
Source Name	Name of the JMS broker.
Type	Choose Topic or queue.
User	Enter the user name if one is required by the JMS broker.
Password	Enter the password if one is required by the JMS broker.
Vendor Name	Choose Tibco, BEA, or MQ from the drop-down list.
Dead Letter Destination	Specify the queue where AON can store undeliverable messages.
Transaction store	Specify the transaction queue.
Reply To List	Click the Edit List button to make a selection.
Destination List	Click the Edit List button to make a selection.
Source List	Click the Edit List button to make a selection.
SSL configuration	Choose an available configuration from the drop-down list.
Destination Batch Size	Size of the batch at the destination broker.
Destination Batch Interval	Specified in milliseconds.

JMS Naming Property

Note Before configuring this property, go to **Admin > Extensions > JMS Resources** to upload a JMS resource file. See the *AON Programming Guide* for information on creating a JMS resource file.

How to Get There

Open a project, then go to **Properties** > **JMS > Node**. Select a node, then click the Edit Properties button.

Data to Enter

The JMS Naming Property page includes the entries described in Table 3-16.

Table 3-16 Entries on JMS Naming Property

Entry	Description
Name	Name of your choosing for this property.
Naming Service	Choose remote or local.
JMS Resource File	Use the drop-down list to select the file you have uploaded.
Initial Context Factory	Constant that holds the name of the environment property for specifying the initial context factory to use.
Provider URL	Constant that holds the name of the environment property for specifying configuration information for the service provider to use.

Table 3-16 Entries on JMS Naming Property (continued)

Entry	Description
Security Protocol	Constant that holds the name of the environment property for specifying the security protocol to use.
Security Authentication	Constant that holds the name of the environment property for specifying the security level to use
Authoritative	Constant that holds the name of the environment property for specifying the authoritativeness of the service requested.
URL Package Prefixes	Constant that holds the name of the environment property for specifying the list of package prefixes to use when loading in URL context factories.
State Factories	Constant that holds the name of the environment property for specifying the list of state factories to use.
Language	Constant that holds the name of the environment property for specifying the preferred language to use with the service.
Batch Size	Constant that holds the name of the environment property for specifying the batch size to use when returning data via the service's protocol.
Security Principal	Constant that holds the name of the environment property for specifying the identity of the principal for authenticating the caller to the service.
Object Factories	Constant that holds the name of the environment property for specifying the list of object factories to use.
Referral	Constant that holds the name of the environment property for specifying how referrals encountered by the service provider are to be processed.
Security Credentials	Constant that holds the name of the environment property for specifying the credentials of the principal for authenticating the caller to the service.
DNS URL	Constant that holds the name of the environment property for specifying the DNS host and domain names to use for the JNDI URL context.
Connection List	Click the Edit List button to choose a JMS Connections property.

Step 3 Click the **Properties** tab on the top right window of AMC.

Step 4 Ensure that the **Adapter** menu is selected on the left pane.

Step 5 Click on the **Node** sub-menu under **Adapter**.

Step 6 Select the AON node for which to configure the connection.

Step 7 Click on **Edit Properties** button.

Configuring Cisco AON Promiscuous Mode

Promiscuous mode (PMode) enables out-of-band message processing using a Cisco AON node. It provides the capability to receive and process messages without introducing latency in the flow of inline network traffic, supporting out-of-band monitoring and analysis.

Prerequisites for Promiscuous Mode

- Ensure that AMC and all AON nodes are correctly configured and running.
- Ensure that any nodes to be used in this procedure are active on AMC.
- Ensure that you have available a valid framing extension. HTTP framing extensions, in addition to FIX extensions, are available for download with other AON software.
- Ensure that the switch or router that hosts any node using PMode meets the requirements in Table 3-17.

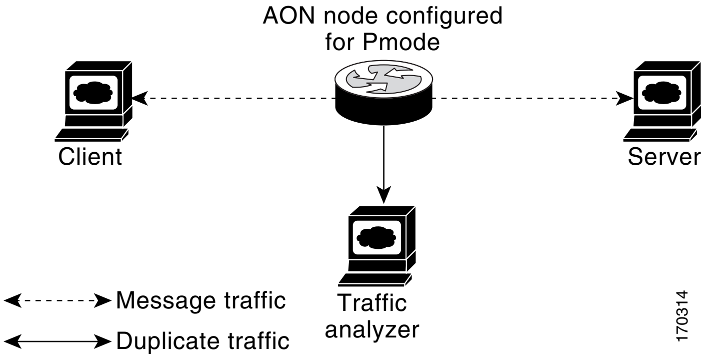
Table 3-17 PMode Operating System Requirements

Platform	Required Operating System
AON-SM with Supervisor Engine 2	Catalyst OS Release 8.5(3) recommended **Note** Minimum requirement is Catalyst OS Release 8.4(2a).
AON-SM with Supervisor Engine 720	PMode not supported
AON-NM	Cisco IOS Release 12.3(14)T1

Information About Promiscuous Mode

Promiscuous mode allows for message traffic monitoring without affecting traffic flow. When promiscuous mode is enabled, message packets are duplicated in the node and forwarded, in the form of framed application messages, to a third-party application. The forwarded messages can be analyzed or otherwise processed. Figure 1 shows a sample runtime topology where an AON node is using PMode to forward traffic to a traffic analyzer.

Figure 1 Promiscuous Mode Sample Topology



The sample topology shown in Figure 1 requires the following runtime components:

- **Client**—sends traffic to the server. The client is configured with a default gateway IP address that is assigned to an interface on the router hosting the AON node.
- **Server**—receives traffic from the client through the AON node. The server is configured with the default gateway IP address of the router interface into which it connects.
- **AON node**—the router or switch, configured with IP addresses and port numbers for the traffic to be captured. The node makes copies of this traffic and passes it to AON. AON in turn processes these messages, packages them into AON monitoring messages (AMM), and sends them to the analyzer. Depending on the node's location in the network, the AON node requires a specific IP and VLAN configuration to perform this function.
- **Traffic analyzer**—receives duplicate traffic from the AON node. The analyzer is a third-party or customer-provided component. It is not part of the AON product.

Pmode Deployment Options

You can run promiscuous mode both on AON-NM and on AON-SM.

When you use AMC to deploy PMode on an AON-NM, PMode is enabled, by default, on the external interface—with the option of changing to an internal monitoring interface. You can choose to use either of the interfaces, or set up a deployment that uses both interfaces simultaneously. For information on changing to an internal monitoring interface, see the section Enabling the Internal Interface on an AON-NM, page 3-19.

When you use AMC to deploy PMode on an AON-SM, PMode is enabled on Gigabit Ethernet 3, a deployment for which you must configure either SPAN or VACL for forwarding the traffic.

For copying traffic to AON, you can select from the following options:

- Configure RITE (Router IP Traffic Export) at the router.
- Use SPAN or VACL in a switch to capture and direct traffic to AON.

Note When using RITE, AON can reside in the same router as that you configure for RITE, or it can reside in a separate router—if in a separate router it must be within the same VLAN.

To configure RITE, see the following:

- http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a0080455b94.html

To configure either SPAN or VACL see the following:

- SPAN—http://www.cisco.com/warp/public/473/41.html
- VACL—http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/vacl.htm

Note Use either SPAN or VACL, but not both.

For information on SPAN and on VACL configurations, see the following documents:

- SPAN—http://www.cisco.com/warp/public/473/41.html
- VACL—http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/122sx/swcg/vacl.htm

Promiscuous Mode Enhancements

Beginning with AON Version 2.4, PMode provides the following enhancements:

Promiscuous Mode Support for UDP Packets

Pmode now has the ability to capture UDP packets. The feature is enabled by default; no further configuration is necessary.

Promiscuous Mode Support for Time-Based Interval Sampling

To configure time-based sampling, open a project and go to **Properties > Adapter** and select the **pmode** adapter. Click the **Properties** button, then choose the **PmodeAdapter** and click the **New** button. The page that loads includes three new elements, which are described in the following table:

Element	Description
Default Sampling Mode	**True** enables time-based interval sampling. **False** disables it.
Default Sampling Delay	Number of seconds to wait before sampling commences.
Default Sampling Duration	Number of seconds sampling will last once it commences.

How to Configure Promiscuous Mode

PMode configuration involves the following:

- Enabling the Internal Interface on an AON-NM, page 3-19 (optional)
- Configuring PMode Adapter on AMC, page 3-19 (required)

Enabling the Internal Interface on an AON-NM

This optional procedure is required only if you are enabling PMode on the internal interface of an AON-NM. To configure this, complete the following steps.

Step 1 Establish a session to the AON-NM and enter configuration terminal mode.

```
aon-nm> configuration terminal
Enter configuration commands, one per line. End with exit.
```

Step 2 Use the **aon monitoring interface** command to enable the internal interface.

```
aon-nm(config)>aon monitoring interface internal
```

Step 3 Exit configuration terminal mode.

```
aon-nm(config)>exit
```

Configuring PMode Adapter on AMC

Activate the Cisco AON PMode adapter by performing the following steps.

Step 1 Click the **Properties** tab in the top menu of AMC.

Step 2 Click **Global** in the **Adapter** menu on the left side of the window. The global properties of each registered adapter are displayed.

Step 3 Make sure **PMode** adapter is checked. On the bottom part of the window, click **Edit** to display the **Edit Property Set** window.

Step 4 In the **Is Active** field, choose **true**.

Step 5 Click **Submit**.

Configuring the PMode Adapter

To configure the PMode adapter and deploy the changes to the node, perform the following steps:

Step 1 Click on the **Properties** Tab in the top menu of AMC.

Step 2 Select the **Adapter** menu in the left pane.

Step 3 Select the sub-menu **Global** under **Adapter**.

Step 4 Select **PMode**.

Step 5 Click the **Properties** button.

Step 6 Select **PmodeAdapter** under **Categories**.

Step 7 Click on **New**.

Step 8 Enter the name as **default**.

Warning **If you enter a name other than "default," the configuration will fail.**

Step 9 Enter the **Default Destination port** as 5011 for our example.

Step 10 Enter the **Default Destination IP** as the IP address of the analyzer.

Step 11 Click **Submit**.

Step 12 Click the **Deploy** Tab in the top menu of AMC.

Step 13 Click **Manage Staging** on the menu in the left window.

Step 14 Notice a **Global Deployment Request**. Select the Global deployment request and click **Stage** as shown below.

Step 15 Click on **Manage Deployment** in the menu in the left window.

Step 16 Select the Global deployment Request and click **Deploy**.

Once deployed, a message ‘Successfully deployed all configurations to the node’ displays.

Loading the HTTP Extension

To load the HTTP extension, perform the following steps:

Step 1 Click **Admin** tab on the top right of the window of AMC.

Step 2 On the left side window, click Adapter Extension Packages in the Extensions menu.

Step 3 Click **Upload**.

Step 4 Click **Register** on the **Upload and Register Package** window.

Enabling the HTTP Extension

To enable the HTTP Extension, perform the following steps:

Step 1 Click the **Properties** tab on the top of the window.

Step 2 On the left hand side of the window, click **Global** in the **Adapter** menu.

Step 3 Select the **PMode** radio button, then click the Extensions button.

The **PMode Adapter Extensions: Global Properties** window displays.

Step 4 Select the **HTTP-FRAMING-EXTN-1** radio button, then click the **Edit** button.

Step 5 The **Adapter Extension Registry: Edit Property Set** window displays. In the **Is Active** field choose **True**, then click the **Submit** button.

The adapter extension is now activated.

Configuring HTTP Extension

To configure the HTTP Extension, perform the following steps:

Step 1 Click on the **Properties** Tab.

Step 2 Make sure that **Global** is selected under **Adapter** menu in the left hand pane. Select the **PMode** radio button, then click the **Properties** button.

Step 3 Select **PmodeAdapterExtension** and click the **New** button.

Step 4 Enter a name for this extension, then click the **Edit List** button.

Step 5 Select **HTTP-FRAMING-EXTN-1**, click the **Save** button, then click the **Submit** button.

Step 6 Click the **Deploy** Tab in the top menu of AMC.

Step 7 Click **Manage Staging** on the menu in the left window.

Step 8 Notice a new **Global Deployment Request**. Select the Global deployment request and click **Stage** as shown below.

Step 9 Click on **Manage Deployment** in the menu in the left window.

Step 10 Select the Global deployment Request and click **Deploy**.

Once deployed a message ‘Successfully deployed all configurations to the node’ is displayed.

Step 11 Click the **Properties** tab on the top right window of AMC.

Step 12 Ensure that the **Adapter** menu is selected on the left pane.

Step 13 Click on the **Node** sub-menu under **Adapter**.

Step 14 Select the AON node for which to configure the connection.

Step 15 Click on **Edit Properties** button.

Step 16 Select the **PMode** property from the list of Node Properties for the AON node.

Step 17 Click on the **Extensions** button.

Step 18 Select **HTTP-FRAMING-EXTN-1** and then click the **Properties** button.

Step 19 Under **Categories** select **HTTPExtension** and click **New**.

Step 20 For **Name**, enter a name for the connection—**Connection1** in the example

Step 21 For **Monitored IP**, enter the IP address of the Server machine—10.221.1.14 in the example.

Step 22 For **Mask**, enter 255.255.255.255

Step 23 For **Monitored Port**, enter the value of the port to monitor—while 9000 is used in the example below, the default value is 80.

Step 24 Click the **Edit List** button next to **AdapterExtPolicyLink**.

Step 25 Select the **HTTP_EXTN** and click **Save**.

Step 26 Review your entries and click the **Submit** button.

Step 27 Select the **Deploy** Tab.

Step 28 Notice a listing under **Open Node Deployment Requests**. Stage the request by clicking **Stage**.

Step 29 Select **Manage Deployment** in the left hand window and select the deployment request, then deploy it by clicking **Deploy**.

Once the request is successfully deployed, a message ‘Successfully deployed configuration to node’ displays.

Step 30 Establish a session to the AON node and restart it as follows:

```
aon-node> enable
aon-node# aon restart force
CAUTION! Stopping all AON processes!
Are you sure[n]? y
```

Note The PMode configuration will take affect once the node is restarted.

Connecting to Databases

Database properties enable AON to read and write to databases. For example, PEPs that use the Log bladelet need a database property that tells AON where to write log data. This is a global property.

How to Get There

Open a project, then go to **Properties > Application > Global**, then select Databases.

Data to Enter

The Database Property page includes the entries described in Table 3-18.

Table 3-18 Entries on Database Property

Entry	Description
Name	Name of your choosing for this database property
User ID	User ID required to log on to the database. The user must have permission to create, read, write, update, and query the database.

Table 3-18 Entries on Database Property (continued)

Entry	Description
Password	Password required to log on to the database
JDBC URL	Location of database. This entry must use one of the following formats:\ Oracle: **jdbc:oracle:thin:**@*ip_address*:*port***:***database_name* Sybase: **jdbc:sybase:Tds:** *ip_address***:***port***/***database_name*
Database name	One of the following: • Oracle • Sybase

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes.

Click **Cancel** to discard your changes and return to the previous screen.

CHAPTER 4

Managing AON Security

This chapter describes AON functions relating to security, authentication, and authorization. It includes the following topics.

- Managing Keystores, page 4-1
- Configuring Security Properties, page 4-6
- Configuring Authentication and Authorization Properties, page 4-8

Note You must have System Administrator or Security Administrator privileges to perform most of the tasks described in this chapter. Application Administrator and Application Developer have limited abilities on the Keystore Tab. See the "Assigning Roles to Users" section on page 5-3 for further details.

Managing Keystores

The Keystore tab is used for managing the keypairs, trustpoints, and root certificates used in the AON network. See the following sections:

- Configuring a Keystore Passphrase, page 4-1
- Managing Keypairs, page 4-2
- Manage Public Certificates or Root Certificates, page 4-5

Configuring a Keystore Passphrase

When AMC is started for the first time, the global keystores used by AMC are automatically created with the passphrase **aonsadmin**. To ensure the security of the keystores, it is recommended that you immediately change this password.

How to Get There

Go to **Keystores > Configuration**. Enter your old and new passwords, then click the **Submit** button.

Managing Keypairs

Keypairs are the public and private keys used by devices in the AON network to encrypt messages. Most keypair management tasks are performed in the Active Repository. AMC also includes a keypair archive, for expired or revoked keypairs.

How to Get There

Go to **Keystores > Keypairs > Active Repository**. This opens the Keypair Active Repository.

Actions to Take

You can perform any of the following actions:

- Upload a PCKS#12 file. See the "Upload PKCS#12" section on page 4-2.
- Generate and register a MPKI Keypair. See the "Generate and Register a New Key" section on page 4-3.
- Generate a self-signed keypair. See the "Generate a Self-Signed Keypair and Certificate" section on page 4-3.
- Add an SSL Certificate. See the "Generate an SSL Certificate" section on page 4-4.
- Import a keystore from another source. See the "Import a Keypair or Keystore" section on page 4-4.

Upload PKCS#12

PKCS#12 is a standard for securely storing private keys and certificates. You can upload a PKCS#12 file (with a .pfx file extension) containing this information.

How to Get There

Go to **Keystores > Keypairs > Active Repository > Upload PKCS#12**.

Data to Enter

The Upload PKCS#12 File page includes the entries described in Table 4-1.

Table 4-1 Upload PKCS#12 File Entries

Entry	Description
Alias	Name of your choosing for this key.
PKCS#12 file	Full path and file name. Click the **Browse** button to locate the file to be imported. The file must have a .pfx extension.
Password	Password used to secure the key.

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes and upload the file.
- Click **Cancel** to discard your changes and return to the previous screen.

Generate and Register a New Key

If you have a managed public key infrastructure (PKI) account with Verisign, you can use AMC to generate and register a new key.

How to Get There

Go to **Keystores > Keypairs > Active Repository > MPKI Keypair**.

What to Enter

The Generate and Register Key page includes the entries described in Table 4-2.

Table 4-2 Generate and Register Key Entries

Entry	Description
Key name	The key name is provided by your managed PKI administrator. It looks similar to the following: http://xkms.verisign.com/keyname?jurisdiction=d7ea68c518b2602ca4bbc....
Passcode	The passcode is provided by your managed PKI administrator.
Key alias	Name of your choosing for this key. Lower case characters only.
Revocation password	Enter a password to be used should this key need to be revoked.
XKMS service	Click Pilot for pre-production environments. Click Production for production environments.

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes and upload the file.
- Click **Cancel** to discard your changes and return to the previous screen.

Generate a Self-Signed Keypair and Certificate

If you do not need a key validated by third parties or business partners, AMC can generate a key without a managed PKI account.

How to Get There

Go to **Keystores > Keypairs > Active Repository> Self-Signed Keypair**.

Data to Enter

Complete the entries as required for your organization and click the **Submit** button.

Generate an SSL Certificate

AMC includes the ability to submit a Certificate Signing Request (CSR) to Verisign. This request can be for a free trial certificate valid for 14 days, or if you are a MPKI SSL customer, it can be for a permanent certificate.

How to Get There

Go to **Keystores > Keypairs > Active Repository > SSL Certificate**.

Data to Enter

Complete the entries as required for you organization and click the **Submit** button. AMC generates the server certificate and displays it on the Add SSL Server ID page.

Actions to Take

Use the mouse to select and copy the entire Certificate Signing Request. You will paste this certificate into the appropriate form at the Verisign.

After copying the CSR and clicking Next, a new browser window opens and loads the Verisign where you complete the process for registering your SSL server ID.

Complete the enrollment process to register the certificate generated by AMC.

After completing the process at Verisign, return to the Active Repository in AMC and click the Pending link for your new certificate.

On the screen that loads, click the **Next** button to display the Install SSL Digital Certificate page.

Actions to Take

Paste the certificate you received from Verisign and click the **Submit** button.

Import a Keypair or Keystore

You can import an existing keystore that contains your public and private certificates.

How to Get There

- **Keystores > Keypairs > Active Repository > Import Keystore**.
- **Keystores > Public Certificates > Active Repository > Import Keystore**
- **Keystores > Root Certificates > Active Repository > Import Keystore**

Data to Enter

The Import Keystore page includes the entries described in Table 4-3.

Table 4-3 Import Keystore Entries

Entry	Description
File	Full path and filename. Click the **Browse** button to locate the file to be imported. The file must be a Java 1.4 JKS format keystore file
Keystore password	Password used to secure this keystore.

Table 4-3 Import Keystore Entries (continued)

Entry	Description
Is Keystore Password Different from Key Alias Password	Click **Yes** or **No**

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes.
- Click **Cancel** to discard your changes and return to the previous screen.

Manage Public Certificates or Root Certificates

The procedure for managing public certificates and root certificates are identical. This section covers the following functions:

- Add a Certificate, page 4-5
- Import a Keystore, page 4-6

Add a Certificate

The Add Certificate page enables you to retrieve, upload, or paste a digital certificate.

How to Get There

Navigate one of the following paths:

- **Keystores > Public Certificates > Active Repository > Add Certificate**
- **Keystores > Root Certificates > Active Repository > Add Certificate**.

Data to Enter

The Add Certificate page includes the entries described in Table 4-4.

Table 4-4 Add Certificate Entries

Entry	Description
Alias	Name of your choosing for this certificate.
URL	URL from which AMC can retrieve the certificate. Click the **Get from SSL connection** radio button to use this entry.
File	Full path and file name. Click the **Browse** button to locate the file to be imported. Click the **Upload** radio button to use this entry.
Base64 certificate	Paste the certificate in this entry. Click the **Cut and paste digital certificate** radio button to use this entry.

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes.
- Click **Cancel** to discard your changes and return to the previous screen.

Import a Keystore

You can retrieve a certificate by importing an existing keystore. See the "Import a Keypair or Keystore" section on page 4-4 for detailed instructions.

Configuring Security Properties

These properties enable you to configure the security settings of individual nodes. This section covers the following sections:

- Endpoint SSLID Property, page 4-6
- SSL Configuration Property, page 4-6
- SSL Binding Property, page 4-7

Endpoint SSLID Property

The Endpoint SSLID property is used to specify the keypair alias to be used by a node for SSL.

How to Get There

Go to **Properties > AON Security > Node > Endpoint SSLID > New**.

Data to Enter

Enter a name for the Endpoint SSLID property, then click the **Next** button. This loads a page on which you can choose a keypair to associate with this property.

SSL Configuration Property

SSL Configuration Property specifies SSL-related parameters to be used by a node.

How to Get There

Go to **Properties > AON Security > Node > SSL Configuration**

Note Before configuring the SSL Configuration Property, you must configure SSLID. See the "Endpoint SSLID Property" section on page 4-6 for details.

Data to Enter

The Security Property page includes the entries described in Table 4-5.

Table 4-5 Security Property Entries

Entry	Description
Name	Name of your choosing for this property.
Endpoint Identity	Choose an available SecurityID from the drop-down list.
SSL Protocol Version	Drop-down list of available versions of SSL. Choose either of the following properties: • **TLS_v1**—Transport Layer Security version 1 (also known as SSL version 3.1) • **SSL_v23**.— Secure Sockets Layer version 2 or version 3.
Extract Peer Certificate	Specifies whether peer certificate extraction is to be used. If PEPs are to use the extracted certificate, this option must be set to **yes**.

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes.
- Click **Cancel** to discard your changes and return to the previous screen.

SSL Binding Property

The SSL Binding property enables you to bind a message's source IP, destination IP, and destination port to an SSL property.

How to Get There

Go to **Properties > AON Security > Node > SSL Binding**

Note Before configuring SSL Binding, you must configure SecurityID and Security Property. See the "Endpoint SSLID Property" section on page 4-6 and the "SSL Configuration Property" section on page 4-6 for details.

Data to Enter

The SSL Binding property page includes the entries described in Table 4-6.

Table 4-6 SSL Binding Property Entries

Entry	Description
Source IP Address	IP address of source.
Destination IP Address	IP address of destination.
Destination Port	Port on which outbound peer is listening for SSL traffic.
Inbound SSL Property	Select an available SSL property from the drop-down list.

Table 4-6 SSL Binding Property Entries (continued)

Entry	Description
Outbound SSL Property	Select an available SSL property from the drop-down list.
Inbound Peer Verification	Select yes or no to specify whether inbound peer verification is to be used.
Outbound Peer Verification	Select yes or no to specify whether outbound peer verification is to be used.

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes.
- Click **Cancel** to discard your changes and return to the previous screen.

Configuring Authentication and Authorization Properties

This section covers the following properties:

- Configuring LDAP, page 4-8
- Configuring Kerberos, page 4-9

Configuring LDAP

Lightweight Directory Access Protocol (LDAP) is a protocol for accessing online directory services. This property can be configured at the node or global levels. After configuring this property, nodes in your AON network are able to access an LDAP directory for authentication and authorization.

How to Get There

- **Properties > Authentication & Authorization > Global > LDAP**
- **Properties > Authentication & Authorization > Node > Edit Properties**

Data to Enter

This information varies from site to site. Contact your LDAP administrator for proper configuration data.

Actions to Take

After completing the entries, you can take one of the following actions:

- Click **Submit** to save your changes.
- Click **Cancel** to discard your changes and return to the previous screen.

Configuring Kerberos

Kerberos is an authentication protocol that enables entities communicating over an insecure network to prove their identities to each other. In so doing, Kerberos provides detection of modification and the prevention of eavesdropping.

Kerberos configuration is controlled by three properties, which must be configured in the following order:

1. Kerberos Services.
2. Kerberos Realms.
3. Kerberos Info.

In order to complete this configuration, you need specific data from the Kerberos service running on your network.

Step 1 Go to **Properties > Authentication & Authorization > Node > Kerberos Services**.

Data to Enter

This information varies from site to site. Contact your Kerberos administrator for appropriate values.

Step 2 Go to **Properties > Authentication & Authorization > Node > Kerberos Realms.**

Data to Enter

This information varies from site to site. Contact your Kerberos administrator for appropriate values.

Step 3 Click the **Edit List** button and select the Kerberos Services property you created.

Step 4 Go to **Properties > Authentication & Authorization > Node > Kerberos Info**.

Data to Enter

This information varies from site to site. Contact your Kerberos administrator for appropriate values.

Step 5 Click the **Edit List** button and select the Kerberos Realms property you created.

Step 6 Click the **Submit** button to save your changes.

CHAPTER 5

AMC Administration

This chapter covers the Admin portion of AMC. It includes the following sections:

Note Only the System Administrator can perform all of the tasks described in this chapter. Other user types have limited abilities on the Admin Tab. See the "Assigning Roles to Users" section on page 5-3 for further details.

AON Licensing

AMC provides the ability to upload licenses that enable additional features and functionality. Contact your Cisco representative to obtain more information about licensing.

How to Get There

Go to **Admin > Licensing**, then click the Add button.

Actions to Take

Click the Browse button to navigate to the location on your PC where the license file is stored, then click the Upload button to send the file to the AMC.

Managing AON Users

AMC users fall into one of the following categories:

- Local users—these users are created and managed within AMC.
- External users—these users are created on and managed by an external LDAP server.

Note A new installation of AMC includes several local users with **aonsadmin** as their default password. To ensure that only authorized personnel have access to the AMC, change the default passwords or delete unneeded users.

Managing Local Users

Local users are created and managed by the AMC. You can use this page add to perform the following tasks:

- Add and delete users
- Display information about users
- Edit a user's information, including privileges.
- Change a user's password

How to Get There

Go to Admin > Users > Manage Local Users

Actions to Take

Click one of the following buttons:

- **New**—creates a new users. See Creating New Users, page 5-2
- **Show**—displays information on the selected user. See Displaying Information on Users, page 5-3
- **Edit**—changes information about the selected user. See Editing Users, page 5-3
- **Password**—changes the password of the selected user.
- **Delete**—removes the selected user from the system.

Creating New Users

AMC enables you to create new local users.

How to Get There

Go to **Admin > Users > Manage Local Users**, then click the New button.

Actions to Take

Enter the appropriate information for the user and select a role. Use Control+click to select multiple roles. For description of available roles, see Assigning Roles to Users, page 5-3.

After completing the fields, click the **Submit** button to save your changes.

Displaying Information on Users

You can use AMC to display information on a selected user, including name, email address, and roles assigned.

How to Get There

Go to **Admin > Users > Manage Local Users**, then select a user. Click the Show button to display the information.

Editing Users

AMC provides the ability to edit the properties of existing local users.

How to Get There

Go to Admin > Users > Manage Local Users, then select a user. Click the **Edit** button.

Actions to Take

Make changes as necessary. If you are changing the role of a user, see the "Assigning Roles to Users" section on page 5-3.

Once you have made your changes, click the **Submit** button to save them.

Deleting Users

The **Delete User Confirmation** page lists details of the user being deleted, including projects to which the user may be assigned. Click **No** to cancel the deletion and return to the previous page. Clicking **Yes** brings about one of the following options:

- If the user is not working on an open project, the user is immediately deleted.
- If the user is working on an open project, a new page will ask for final confirmation before deleting the user.

Assigning Roles to Users

AMC users can be given roles based on their need to perform certain actions on AMC. Each role grants specific privileges within AMC. For example, the Application Designer role can only access the project to which it is assigned and upload extensions to the AMC, however, a Network Administrator can access functions related to managing and monitoring nodes. To give a user full access to AMC, assign the System Administrator role to that user.

Role-Based Access Control

Two new user roles are introduced in AON Version 2.4. Users with the System Administrator role are permitted to access any project. Users with the Application Developer role are able to access only the projects to which they are assigned. Users that were assigned the roles Network Administrator, Security Administrator, and Application Administrator in previous AON releases are now assigned the role System Administrator, If a user does not have permission to access system-wide resources, AMC hides those resources from the user.

The table below shows the roles available in AMC, and the sections on each tab these roles can access.

Table 5-1 AMC User Roles

Role	Project Tab	Network Tab	Properties Tab	Deploy Tab	Monitor Tab	Keystores Tab	Admin Tab
Application Admin	All actions on assigned projects except deletion and new project creation	—	• Application • JMS • Monitoring • AON Security • Service Profiles	All	All	• Keypairs • Public Certificates	• AMC Diagnostics • Extensions • AMC Security • Data Migration • Users
Application Designer	Open and view assigned projects only	—	—	—	—	—	• AMC Diagnostics • Extensions
Application Developer	Open and view assigned projects only	—	• Application • JMS • Monitoring • AON Security • Service Profiles	All	All	• Keypairs • Public Certificates	• AMC Diagnostics • Extensions • AMC Security • Data Migration • Users
Network Admin	Open and view all projects	All	Monitoring	—	All	—	—
Security Admin	Open and view all projects	—	• Authentication and Authorization • AON Security • Node Management Security	—	All	All	• Users • AMC Security
System Admin	All	All	All	All	All	All	All

Editable Roles

When you edit users' roles, you can assign only those roles that are equal to or less than your own role. For example, an application administrator cannot give the system administrator role to a user. The table below lists each role and the roles that user is able to edit.

Assigned Role	Editable Roles
System administrator	All
Application administrator	All application-related roles
Application designer	None
Application developer	None
Network administrator	None
Security administrator	All except system administrator

Note Do not remove the system administrator role from the user named **aonsadmin** unless you have first assigned that role to at least one other user.

To assign roles to a user, see one of the following sections:

- Creating New Users, page 5-2
- Editing Users, page 5-3
- Assigning Roles to External Users, page 5-6

Managing External Users

AMC provides the ability to use an existing LDAP server for user management. To do this, complete the following tasks in the order specified:

1. Creating an LDAP Profile, page 5-5
2. Assigning Roles to External Users, page 5-6
3. Creating an Authentication Realm, page 5-6

Creating an LDAP Profile

An LDAP profile provides the information needed by AMC to retrieve user data from an existing LDAP server.

How to Get There

Go to **Admin > Users > Manage Local Users > LDAP**, then click the **New** button.

Actions to Take

Complete the fields as appropriate for the LDAP server being used. Contact your LDAP administrator for details.

Assigning Roles to External Users

How to Get There

Go to **Admin > Users > Manage Local Users > Role Mapping**, then click the **New** button.

Data to Enter

Table 5-2 shows the field of the Role Mapping page.

Table 5-2

Entry	Description
Name	Name of your choosing for this property set.
LDAP Attribute Name	The LDAP attribute that is to be used to specify the AMC role.
Condition Operator	Choose one of the following from the drop-down list: • equals—information retrieved from LDAP server must match exactly with LDAP attribute value specified below. • contains—information retrieved from LDAP server must contain LDAP attribute value specified below. • defineRoles—information retrieved from LDAP will define the role of the user.
LDAP Attribute Value	The value for the attribute specified above.
Assign Roles	Click the **Edit List** button to choose roles that are to be assigned to users who match the LDAP attribute. See "AMC User Roles"

Actions to Take

After completing the fields, click the **Submit** button to save your changes.

Creating an Authentication Realm

The LDAP Authentication Realm binds the LDAP information specified in the "Creating an LDAP Profile" section on page 5-5 with the role mapping information specified in "Assigning Roles to External Users" section on page 5-6.

How to Get There

Admin > Users > Manage Local Users > Authentication Realm, then click the **New** button.

Data to Enter

Table 5-3 shows the Authentication Realm page.

Table 5-3 Authentication Realm

Entry	Description
Name	Name of your choosing for this property set.
Realm Name	Name of your choosing for the realm.

Table 5-3 Authentication Realm (continued)

Entry	Description
LDAP Connection Profile	Choose an available LDAP profile from the drop-down list. See the "Creating an LDAP Profile" section on page 5-5 to create a new profile.
Role Mapping Policies	Click the Edit List button to select from the available Role Mapping property sets. See the "Assigning Roles to External Users" section on page 5-6 to create a new property set.

Actions to Take

After completing the fields, click the Submit button to save your changes. Once you completed this task, the LDAP configuration appears in the drop-down list on the AMC log-in page.

Managing AMC Certificates

The AMC Security Page enables you to manage the keypairs and certificates used by AMC for secure communication.

- **Keypairs**–view, edit, or delete keypairs that have been assigned to AMC.
- **Public Certificates**–view, edit, or delete public certificates that have been assigned to AMC.
- **Root Certificates**–view, edit, or delete root certificates that have been assigned to AMC.

How to Get There

- Go to **Admin > AMC Security**

Note If no keypairs or certificates are present, you must use the Keystores tab to generate them.

Managing Extensions

The Extensions page enables you to upload custom software to adapt your AON environment to the specific business needs of your network. This page provides the ability to upload the following:

- JMS resources
- Schema packages
- Transform packages
- Transform parser packages
- Custom bladelets and extensions

If you have opened the System project, you can also upload the following:

- Adapter packages
- Adapter extension packages

Extensions are uploaded into and deployed from a particular project. Any property set categories included in the extension are accessible only to the project that uploaded the extension. Each project must upload its own unique extensions. An extension uploaded by an application project cannot be shared among other projects. However, extensions uploaded into the system project can be shared.

When an extension package is uploaded, AMC appends the project prefix to the package name. However, the prefix is not applied to the custom extension itself, nor is it applied to any custom policy categories and predefined property sets in the package.

AMC does not allow extension packages to have duplicate names. An attempt to upload a duplicate extension will fail. Additionally, AMC verifies that any custom property sets contained in an extension do not exist in other projects. If there is duplication, the upload attempt will fail.

Note Before you can upload an extension with AMC, you must use AON Development Studio to package it.

For more information about developing and packaging extensions, see the *AON Programming Guide*.

APPENDIX A

AON Schemas

This appendix contains schemas used by AMC. It includes the following:

- Archive Schema, page A-1
- Programmatic Management Interface APIs, page A-15
- Message Log Schemas, page A-26

Note This information spans multiple pages, making it difficult to select and copy when this guide is viewed in Adobe Acrobat. For best results, go to Cisco.com to view this information on a single page:

http://www.cisco.com/univercd/cc/td/doc/product/aon/admin/amc23/aondb.htm

Archive Schema

When you export a project in AON 2.4, AMC writes a zip file containing configuration data. This configuration archive contains all of the data related to the project. You can then edit the information contained in the archive to modify the configuration data that must change in order for the project to function in the new AMC environment.

The file structure of the configuration archive and schemas for individual data files are listed in the sections that follow. You can use this information to modify configuration files as appropriate for your environment. You can also develop custom scripts to automate this task.

Note Cisco reserves the right to modify these schemas as AON evolves. Schemas are not guaranteed to be backward compatible. Thus custom scripts may need to be adjusted from one release to the next.

```
sample_archive
|-- ArchiveInfo.xml (see ArchiveInfo.xml Schema)
|-- network
|   |-- global
|   |   `-- wccpservers
|   |       `-- 192.168.10.31.xml (see WCCP Server Schema)
|   `-- nodes
|       `-- bangalore
|           |-- NodeInfo.xml (see NodeInfo.xml Schema)
|           |-- acls
|           |   `-- acl1.xml (see ACL Schema)
|           |-- amaproperties
|           |   `-- ha.properties
|           `-- wccpservicegroups
|               `-- 55.xml (see WCCP Service Group Schema)
`-- projects
    |-- shared
    |   `-- System
    |       |-- global
    |       |   |-- extensions (see Extensions)
    |       |   |   |-- bladeletextensions
    |       |   |   |   `-- ABladeletExt.scar
    |       |   |   |-- bladelets
    |       |   |   |   `-- ABladelet.scar
    |       |   |   |-- jmsresourcefiles
    |       |   |   |-- schemas
    |       |   |   |   `-- ASchema.sar
    |       |   |   |-- transformparsers
    |       |   |   |   `-- AParser.xfn
    |       |   |   `-- transforms
    |       |   |   |   `-- ATransform.xfn
    |       |   |-- licenses
    |       |   `-- propertysets
    |       |       `-- com.acme.policies.a
    |       |           `-- 1.0
    |       |               |-- AttributeDomain.xml (see AttributeDomain.xml Schema)
    |       |               |-- ps1.xml (see Property Set Schema)
    |       |               `-- ps2.xml
    |       `-- nodes
    |           `-- bangalore
    |               `-- propertysets
    |                   `-- com.cisco.aons.policies.mec.NextHopDomain
    |                       `-- 1.0
    |                           |-- AttributeDomain.xml
    |                           `-- jms.aontest.com_7600.xml (see Property Set Schema)
    |                           `-- jms.aontest.com_7600.xml.metadata
    `-- standalone
        `-- applProject1
            |-- ProjectInfo.xml
            |-- global
            |   |-- extensions
            |   |   |-- bladeletextensions
            |   |   |-- bladelets
            |   |   |-- jmsresourcefiles
            |   |   |-- schemas
            |   |   |-- transformparsers
            |   |   `-- transforms
            |   `-- propertysets
            `-- nodes
                `-- bangalore
                    |-- flows
                    |   `-- MDS_TEST.MDS_TWOWAY
                    |       |-- META-INF
                    |       |   |-- Flow.xml (see Flow.xml Schema)
                    |       |   |-- rules.xml
                    |       |   `-- MANIFEST.MF
                    |       `-- com.cisco.aons.ads.userlayout.xml
                    |-- msgtypes
                    |   |-- MDS_1W_LrgTTL.xml (see Message Type Schema)
                    |   |-- MDS_2W_LrgTTL.xml
                    |   `-- msgtype_order.xml (see Message Type Order File Schema)
                    `-- propertysets
                        `-- com.cisco.aons.policies.mec
                            `-- 1.0
                                |-- AttributeDomain.xml Schema
                                |-- mds-dst-jms.aontest.com_7666.xml
                                |   (see Property Set Schema)
                                `-- mds-dst-jms.aontest.com_7666.xml.metadata
```

ArchiveInfo.xml Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="ArchiveInfo">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="SourceAMCInfo" type="AMCInfo" />
            </xsd:sequence>
            <xsd:attribute name="formatVersion" type="xsd:string"
                default="1.0">
            </xsd:attribute>
            <xsd:attribute name="createdAt" type="xsd:dateTime"
                use="optional">
            </xsd:attribute>
        </xsd:complexType>
    </xsd:element>

    <xsd:complexType name="AMCInfo">
        <xsd:attribute name="hostname" type="xsd:string"
            use="required">
        </xsd:attribute>
        <xsd:attribute name="aonVersionString" type="xsd:string"
            use="required">
        </xsd:attribute>
    </xsd:complexType>
</xsd:schema>
```

NodeInfo.xml Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="NodeInfo">
        <xsd:complexType>
            <xsd:choice>
                <xsd:element ref="StandaloneNode"></xsd:element>
                <xsd:element ref="VirtualClusterNode"></xsd:element>
            </xsd:choice>
            <xsd:attribute name="schemaVersion" type="xsd:string"
                default="1.0">
            </xsd:attribute>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="StandaloneNode">
        <xsd:complexType>
```

```
            <xsd:attribute name="platform" type="xsd:string" use="required">
            </xsd:attribute>
            <xsd:attribute name="name" type="xsd:string"
                use="required"/>
            <xsd:attribute name="aonVersionString" type="xsd:string"
                use="required"/>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="VirtualClusterNode">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="StandaloneNode"
maxOccurs="unbounded" minOccurs="1"/>
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string"
                use="required"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

ProjectInfo.xml Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="ProjectInfo">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Description" type="xsd:string" />
                <xsd:element ref="Users" />
                <xsd:element ref="Nodes" />
            </xsd:sequence>
            <xsd:attribute name="schemaVersion" type="xsd:string"
                default="1.0" />

            <xsd:attribute name="name" type="xsd:string" use="required" />

            <xsd:attribute name="prefix" type="xsd:string"
                use="required" />

            <xsd:attribute name="shared" type="xsd:boolean"
                use="required" />

        </xsd:complexType>
    </xsd:element>

    <xsd:element name="Users">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="User" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="User">
        <xsd:complexType>
            <xsd:attribute name="name" type="xsd:string" use="required" />

            <xsd:attribute name="realm" type="xsd:string"
                default="AMCLocal" />

        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Nodes">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="Node" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="Node">
        <xsd:complexType>
            <xsd:attribute name="name" type="xsd:string" use="required" />
        </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Node Mapping File Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
     elementFormDefault="qualified">

    <xsd:element name="PromotionNodeMap">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="SourceNode" maxOccurs="unbounded" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="SourceNode">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="DestinationNode"
                   minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
            <xsd:attribute name="name" type="xsd:string"/>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="DestinationNode">
        <xsd:complexType>
            <xsd:attribute name="name" type="xsd:string"/>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Flow.xml Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
    <xs:element name="flow">
        <xs:annotation>
            <xs:documentation>
               Root element for describing a flow
             </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="flow-vars" minOccurs="0"/>
            <xs:element name="flow-init" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="operator-blocks" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="flow-steps">
                <xs:complexType>
                    <xs:sequence>
                        <xs:elementname="request-action"
                            type="actionType"/>
                        <xs:elementname="response-action"
                            type="actionType"
                            minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="compensation-steps"
                            type="actionType" minOccurs="0"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <!-- Scope Global scope is necessary -->
            <xs:element ref="scope-blocks" minOccurs="1"/>
            <xs:element ref="operator-blocks" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="package" type="xs:string" use="optional"/>
        <xs:attribute name="description" type="xs:string" use="required"/>
        <xs:attribute name="interactionStyle" type="interactionType"
           use="required"/>
        <xs:attribute name="logging" type="xs:string"
           use="optional" default="none"/>
        <xs:attribute name="sla" type="xs:unsignedInt"
           use="optional" default="0"/>
        <xs:attribute name="enableRM" type="xs:boolean"
           use="optional" default="false"/>
        <xs:attribute name="compress" type="xs:boolean"
           use="optional" default="false"/>
        <!-- Scope (Global) ScopeId is optional for flow -->
        <xs:attribute name="scopeId" type="xs:string" use="optional"/>
        <xs:attribute name="opId" type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
  <xs:element name="flow-vars">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="flow-var"
                 minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:attribute name="type" type="xs:string"
                       use="required"/>
                    <xs:attribute name="name" type="xs:string"
                       use="required"/>
                    <xs:attribute name="id" type="xs:string"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:simpleType name="interactionType">
    <xs:annotation>
        <xs:documentation>
           Defines the interaction type for the flow
         </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:enumeration value="Request-Response"/>
        <xs:enumeration value="Request-Only"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="actionType">
    <xs:sequence>
        <xs:element name="first-bladelet" minOccurs="0">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required"/>
            </xs:complexType>
        </xs:element>
        <xs:element ref="bladelet" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="optional"/>
    <!-- Scope ScopeId is optional for actionType -->
    <xs:attribute name="scopeId" type="xs:string" use="optional"/>
</xs:complexType>
<xs:element name="bladelet">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="param-values"
               minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="exported-value"
               minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="next-steps" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="UUID" type="xs:string" use="optional"/>
        <xs:attribute name="description" type="xs:string" use="required"/>
        <xs:attribute name="id" type="xs:string" use="required"/>
        <xs:attribute name="opBlockId" type="xs:string" use="optional"/>
        <!-- Scope ScopeId is required for bladelet -->
        <xs:attribute name="scopeId" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="operator-blocks">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="operator-block"
                 minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="assignment"
                              minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType mixed="true">
                                <xs:attribute name="from"
                                    type="xs:string" use="required"/>
                                <xs:attribute name="to"
                                    type="xs:string" use="required"/>
                                <xs:attribute name="fromType"
                                    type="xs:string" use="required"/>
                                <xs:attribute name="toType"
                                    type="xs:string" use="required"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attribute name="id"type="xs:string"
                        use="required"/>
                    <xs:attribute name="name" type="xs:string"
                        use="optional"/>
                    <!-- Scope ScopeId is optional for operator blocks -->
                    <xs:attribute name="scopeId" type="xs:string"
                        use="optional"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
  <!-- Scope ScopeBlock id and parent same for global one -->
<xs:element name="scope-blocks">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="scope" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="flow-vars"/>
                    </xs:sequence>
                    <xs:attribute name="id"type="xs:string"
                      use="required"/>
                    <xs:attribute name="name" type="xs:string"
                      use="optional"/>
                    <xs:attribute name="parent" type="xs:string"
                      use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="list-element">
    <xs:complexType>
        <xs:choice>
            <xs:element ref="value"/>
            <xs:element ref="param-values" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute name="type" type="xs:string"
          use="optional" default="string"/>
        <xs:attribute name="elementClass" type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="param-values">
    <xs:complexType>
        <xs:choice>
            <xs:element ref="value" minOccurs="0"/>
            <xs:element ref="list-element"
              minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="param-values"
              minOccurs="0" maxOccurs="unbounded"/>
        </xs:choice>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="designName" type="xs:string" use="optional"/>
        <xs:attribute name="type" type="xs:string"
            use="optional" default="string"/>
        <xs:attribute name="paramGroupFQN" type="xs:string"
            use="optional" default="string"/>
        <xs:attribute name="keyName" type="xs:string" use="optional"/>
        <xs:attribute name="elementClass" type="xs:string" use="optional"/>
        <xs:attribute name="bindId" type="xs:string" use="optional"/>
        <xs:attribute name="bindType" type="xs:string" use="optional"/>
        <xs:attribute name="output" type="xs:boolean" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="exported-value">
    <xs:complexType>
        <xs:attribute name="type" type="xs:string" use="required"/>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="designName" type="xs:string" use="optional"/>
        <xs:attribute name="bindId" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="next-steps">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="next-step"
                      minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="label" type="xs:string"
                           use="optional"/>
                        <xs:attribute name="id"type="xs:string"
                           use="optional"/>
                        <xs:attribute name="opBlockId" type="xs:string"
                           use="optional"/>
                        <xs:attribute name="isBreak" type="xs:boolean"
                           use="optional"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="response-link" minOccurs="0">
                    <xs:complexType>
                        <xs:attribute name="label" type="xs:string"
                           use="optional"/>
                        <xs:attribute name="id"type="xs:string"
                           use="optional"/>
                        <xs:attribute name="opBlockId" type="xs:string"
                           use="optional"/>
                        <xs:attribute name="isBreak" type="xs:boolean"
                           use="optional"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="on-exception"
                    minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="exId" type="xs:string"
                           use="optional" default="default"/>
                        <xs:attribute name="id" type="xs:string"
                           use="optional"/>
                        <xs:attribute name="opBlockId" type="xs:string"
                           use="optional"/>
                        <xs:attribute name="isBreak" type="xs:boolean"
                           use="optional"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="value" type="xs:string"/>
</xs:schema>
```

Message Type Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="msgType">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="fTuple"/>
        <xs:element ref="uri"/>
        <xs:element ref="params"/>
        <xs:element ref="headers"/>
        <xs:element ref="content"/>
        <xs:element ref="policyPart"/>
      </xs:sequence>
      <xs:attribute name="name" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="fTuple">
    <xs:complexType>
      <xs:attribute name="name" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="uri">
    <xs:complexType>
      <xs:attribute name="pattern" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="params">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="param"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="param">
    <xs:complexType>
      <xs:attribute name="name" use="required" type="xs:NCName"/>
      <xs:attribute name="op" use="required" type="xs:NCName"/>
      <xs:attribute name="value" use="required" type="xs:integer"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="headers">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="header"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="header">
    <xs:complexType>
      <xs:attribute name="name" use="required" type="xs:NCName"/>
      <xs:attribute name="op" use="required" type="xs:NCName"/>
      <xs:attribute name="value" use="required" type="xs:integer"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="content">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="expr"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="expr">
    <xs:complexType>
      <xs:attribute name="op" use="required" type="xs:NCName"/>
      <xs:attribute name="value" use="required" type="xs:NCName"/>
      <xs:attribute name="xpath" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="policyPart">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="flow"/>
        <xs:element ref="policy"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="flow">
    <xs:complexType>
      <xs:attribute name="flowId" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="policy">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="attr"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="attr">
    <xs:complexType>
      <xs:attribute name="domain" use="required"/>
      <xs:attribute name="propertyset" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Message Type Order File Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="MsgTypesOrder">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref="MsgType"
                    maxOccurs="unbounded" minOccurs="1"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="MsgType">
        <xsd:complexType>
            <xsd:attribute name="name" type="xsd:string"
                use="required"/>
        </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

AttributeDomain.xml Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.0">
    <xs:element name="AttributeDomain">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="AttributeDefinition"
                    minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="ForwardReference" minOccurs="0"/>
                <xs:element ref="InverseReference" minOccurs="0"/>
                <xs:element ref="KeyDefinition" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="desc" type="xs:string" use="optional"/>
            <xs:attribute name="domainName" type="xs:NMTOKEN" use="required"/>
            <xs:attribute name="fixed" type="xs:boolean"
                use="optional" default="false"/>
            <xs:attribute name="schemaVersion" type="xs:string"
                use="optional" default="1.0"/>
            <xs:attribute name="version" type="xs:string"
                use="optional" default="1.0"/>
            <xs:attribute name="policyGroup" type="xs:string" use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="AttributeDefinition">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="PossibleValue"
                minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="DefaultValue"
                minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="passwordfield" type="xs:boolean"
                use="optional" default="false"/>
            <xs:attribute name="readonly" type="xs:boolean"
                use="optional" default="false"/>
            <xs:attribute name="psref" type="xs:NMTOKEN" use="optional"/>
            <xs:attribute name="attributeName" type="xs:string"
                use="required"/>
            <xs:attribute name="type" type="attributeType"
                use="optional" default="string"/>
            <xs:attribute name="visible" type="xs:boolean"
                use="optional" default="true"/>
            <xs:attribute name="desc" type="xs:string" use="optional"/>
            <xs:attribute name="tooltipKey" type="xs:string" use="optional"/>
            <xs:attribute name="nullable" type="xs:boolean"
                use="optional" default="false"/>
            <xs:attribute name="minValue" type="xs:integer" use="optional"/>
            <xs:attribute name="maxValue" type="xs:integer" use="optional"/>
            <xs:attribute name="maxLength" type="xs:integer" use="optional"/>
            <xs:attribute name="externalDataClass" type="xs:string"
                use="optional"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="PossibleValue">
        <xs:complexType>
            <xs:attribute name="value" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="DefaultValue">
        <xs:complexType>
            <xs:attribute name="value" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="ForwardReference" type="xs:string"/>
    <xs:element name="InverseReference" type="xs:string"/>
    <xs:element name="KeyDefinition">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="AttributeReference"
                    minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="AttributeReference">
        <xs:complexType>
            <xs:attribute name="name" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>

    <xs:simpleType name="attributeType">
        <xs:restriction base="xs:string">
```

```
            <xs:enumeration value="string"/>
            <xs:enumeration value="enum"/>
            <xs:enumeration value="list"/>
            <xs:enumeration value="integer"/>
            <xs:enumeration value="ipv4host"/>
            <xs:enumeration value="port"/>
            <!--
            <xs:enumeration value="password"/>
            <xs:enumeration value="boolean"/>
            <xs:enumeration value="url"/>
            <xs:enumeration value="uriList"/>
            <xs:enumeration value="jdbcUrl"/>
            -->
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Property Set Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  version="1.0">
    <xs:element name="PropertySet">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Attribute"
                    minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="key" type="xs:string" use="required"/>
            <xs:attribute name="version" type="xs:string" use="optional" default="1.0"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Attribute">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="Value"
                    minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="name" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="Value">
        <xs:complexType>
            <xs:attribute name="value" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

ACL Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="fTuple">
    <xs:complexType>
        <xs:attribute name="match" type="xs:string"/>
        <xs:attribute name="srcIp" type="xs:string"/>
        <xs:attribute name="srcPort" type="xs:string"/>
        <xs:attribute name="srcMask" type="xs:string"/>
        <xs:attribute name="destIp" type="xs:string"/>
        <xs:attribute name="destPort" type="xs:string"/>
        <xs:attribute name="destMask" type="xs:string"/>
      <xs:attribute name="name" use="required" type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Extensions

Extensions are exported in the same format as when they were originally uploaded in AMC.

License

Licenses are exported in the same format as when they were originally uploaded in AMC.

WCCP Service Group Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="WCCPConfig">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ServiceGroup"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ServiceGroup">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="InterceptionPorts"/>
        <xs:element ref="ApplicationListenPort"/>
        <xs:element ref="fTuple" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="Client"/>
        <xs:element ref="Server" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="forClusterMgmt" use="required" type="xs:NCName"/>
      <xs:attribute name="id" use="required" type="xs:integer"/>
      <xs:attribute name="loadBalancingRule" use="required"
        type="xs:integer"/>
      <xs:attribute name="multicastGroup" use="required" type="xs:NMTOKEN"/>
      <xs:attribute name="status" use="required" type="xs:NCName"/>
      <xs:attribute name="authenticationPwd" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="InterceptionPorts" type="xs:string"/>
  <xs:element name="ApplicationListenPort" type="xs:integer"/>
  <xs:element name="fTuple">
    <xs:complexType>
      <xs:attribute name="name" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="Client">
    <xs:complexType>
      <xs:attribute name="cn" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="Server">
    <xs:complexType>
      <xs:sequence>
```

```
        <xs:element ref="GroupListenInterfaces"/>
        <xs:element ref="RedirectInterfaces"/>
      </xs:sequence>
      <xs:attribute name="ipAddress" use="required" type="xs:NMTOKEN"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="GroupListenInterfaces" type="Interfaces"/>
  <xs:element name="RedirectInterfaces" type="Interfaces"/>
  <xs:complexType name="Interfaces">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="Interface"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Interface">
    <xs:complexType>
      <xs:attribute name="name" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

WCCP Server Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="Server">
    <xs:complexType>
        <xs:attribute name="accessMethod" type="xs:string"/>
        <xs:attribute name="enablePassword" type="xs:string"/>
        <xs:attribute name="ipAddress" type="xs:string"/>
        <xs:attribute name="password" type="xs:string"/>
        <xs:attribute name="status" type="xs:string"/>
        <xs:attribute name="userName" type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Programmatic Management Interface APIs

The programmatic management interface feature (PMI) provides an interface so that third-party applications can manipulate data in AMC. This feature operates under the following assumptions:

- The client must use HTTPS for transport security to connect to AMC. The client must have a server certificate in the AMC trust store to properly authenticate.
- The client must use the “aonsadmin” user ID to log into AMC.
- All the web service calls are RPC based.
- SOAP version 1.1 is supported. Version 1.2 is not supported.
- With the exception of loginUser, all error messages are handled as SOAPFault. The client is expected to process the SOAPFault appropriately.

The following APIs are included:

- loginUser API, page A-16
- listOfProjects API, page A-17
- exportProject API, page A-18

- importProject API, page A-19
- listOfDRs API, page A-20
- deployDR API, page A-21
- exportAccessControlMapping API, page A-22
- importAccessControlMapping API, page A-24

A sample PMI client is included with the installation of AMC 2.4. See the "Sample PMI Client" section on page A-25 for further details.

loginUser API

This operation allows the PMI client to login and use the subsequent Web service operations. The client is expected use the "aonsadmin" user for PMI Web service operations.

Input Parameters

- loginRequest element contains the following:
  - name: userid
  - password: user's password
  - versionId: future versioning perspective

Output Parameters:

- loginResponse element contains the following:
  - name: userid
  - errorCode: 0 for success. Non-zero values for errors
  - errorMessage: optional error message element

Schema

```
<xs:complexType name="LoginRequestType">
    <xs:sequence>
        <xs:element ref="amcData" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="amcData">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="login" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="login">
    <xs:complexType>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="password" type="xs:string" use="required"/>
        <xs:attribute name="versionId" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>

<xs:complexType name="LoginResponseType">
    <xs:sequence>
        <xs:element ref="amcDataResult" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
```

```
</xs:complexType>

<xs:element name="amcDataResult">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="login" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="amcVersion" type="xs:string" />
    </xs:complexType>
</xs:element>

<xs:element name="login">
    <xs:complexType>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="errorCode" type="xs:string" use="required"/>
        <xs:attribute name="errorMessage" type="xs:string" />
    </xs:complexType>
</xs:element>
```

listOfProjects API

This operation lists all of the projects from the AMC source environment. Later clients will iterate through the project list and may invoke export operation.

Input Parameters

- listOfProjectsRequest element contains the following:
  - /* Returns always the available projects with status */

Output Parameters

- listOfProjectsResponse element contains the following:
  - list of project details (project names)

Schema

```
<xs:complexType name="ListOfProjectsRequestType">
</xs:complexType>

<xs:complexType name="ListOfProjectsResponseType">
    <xs:complexType>
        <xs:sequence>
            <xs:element  ref ="projectdetails" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:complexType>

<xs:element name="projectdetail">
    <xs:complexType>
        <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>

<xs:element name="projectdetails">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="projectdetail"  minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
```

```
</xs:element>
```

exportProject API

This operation exports the resources for a project in a given location.

Input Parameters:

- exportRequest element contains the following:
  - type: string (Type of the entity to be exported. Supported type is project)
  - list of names (name uniquely identify the entity such as project)
  - filename: string (Archive file name)
  - location: string (Location to which the archive file will be stored for export or from which the file will be retrieved.)

Output Parameters:

- exportResponse element contains the following:
  - status: string (success or fail)

Schema

```
<xs:complexType name="ExportRequestType">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="type"  />
            <xs:element ref="projects"  minOccurs="0" />
            <!-- if there is no projects element, then it is all projects -->
            <xs:element name="filename" />
            <xs:element name="location" />
        </xs:sequence>
    </xs:complexType>
</xs:complexType>

<xs:element name="project">
    <xs:complexType>
        <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>

<xs:element name="projects">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="project"  minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:complexType name="ExportResponseType">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="status">
                <xs:simpleType>
                    <xs:restriction base="xs:token">
                        <xs:enumeration value="success"/>
                        <xs:enumeration value="fail"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:complexType>
```

importProject API

This operation imports a project and its resources from the given location.

Input Parameters

- importRequest element contains the following:
  - type: string (Type of the entity to be imported. Supported type is project)
  - filename: string (Archive file name)
  - location: string (Location from which the archive file will be retrieved for import)
  - mappingfile: string (name of mapping file)
  - mappinglocation: string (location of mapping file)
  - import type (possible values are add or merge.)

Output Parameters

- ImportResponse element contains the following:
  - status: string (success or fail)

Schema

```
<xs:complexType name="ImportRequestType">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="type"  />
            <xs:element name="filename"  />
            <xs:element name="location"  />
            <xs:element name="nodemappingfile"/>
            <xs:element name="mappinglocation"/>
            <xs:element name="importtype">
                <xs:simpleType>
                    <xs:restriction base="xs:token">
                        <xs:enumeration value="ADDON"/>
                        <xs:enumeration value="REPLACE"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:complexType>

<xs:complexType name="ImportResponseType">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="status">
                <xs:simpleType>
                    <xs:restriction base="xs:token">
                        <xs:enumeration value="success"/>
                        <xs:enumeration value="fail"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:complexType>
```

listOfDRs API

This operation lists all the DRs for a given project. Later clients will iterate through the DR and invoke deploy operation.

Input Parameters:

- listOfDRRequest element contains the following:
  - type: string (Type of the entity (for which DRs created). Supported type is project.)
  - name: string (Uniquely identify the entity such as project ID). If this element is not present, default is all projects.
  - statustype: string (type of DR such as created, staged, deployed). If this element is not present, default is all status types.

Output Parameters:

- listOfDRResponse element contains the following:
  - list of drs (DR ID and statustype)
  - status: string (success or fail)

Schema

```
<xs:complexType name="listOfDRRequestType">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="type"  />
            <xs:element name="name" />
            <xs:element name="statustype">
                <xs:simpleType>
                    <xs:restriction base="xs:token">
                        <xs:enumeration value="Saved"/>
                        <xs:enumeration value="Staged"/>
                        <xs:enumeration value="Deployed"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:complexType>

<xs:complexType name="ListOfDRResponseType">
    <xs:complexType>
        <xs:sequence>
            <xs:element  ref="drs" />
        </xs:sequence>
    </xs:complexType>
</xs:complexType>

<xs:element name="dr">
    <xs:complexType>
        <xs:attribute name="id" type="xs:string" use="required"/>
        <xsd:attribute name="nodeName" type="xsd:string" use="optional"/>
        <xsd:attribute name="statusType" type="xsd:string" use="required"/>
        <xsd:attribute name="created" type="xsd:string" use="required"/>
        <xsd:attribute name="project" type="xsd:string" use="required"/>
    </xs:complexType>
</xs:element>

<xs:element name="drs">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="dr"  minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

deployDR API

This operation used to stage and deploy the given DR for a particular project.

Input Parameters:

- deployRequest element contains the following:
  - drid: string (Uniquely identify the DR within AMC)

Output Parameters:

- deployResponse element contains the following:
  - status: string (success or fail)

Schema

```
<xs:complexType name="drAttributesType">
    <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>


<xs:complexType name="drsType" mixed="false">
    <xs:annotation>
        <xs:documentation xml:lang="en">

            Implementation only supports a single deployment request
            ID per SOAP request message.

        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="dr" type="drAttributesType"
            minOccurs="1" maxOccurs="1"/>
        </xs:sequence>

    </xs:complexType>

    <xs:complexType name="DeployRequestType">
        <xs:annotation>
            <xs:documentation xml:lang="en">

                Implementation only supports a single drs tag element
                per SOAP request message.

            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="drs" type="drsType"
                minOccurs="1" maxOccurs="1"/>
            </xs:sequence>
        </xs:complexType>

        <xs:element name="deployDR" type="DeployRequestType"/>

    </xs:schema>


    <xs:complexType name="DeployResponseType">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="status" >
                    <xs:simpleType>
                        <xs:restriction base="xs:token">
                            <xs:enumeration value="success"/>
                            <xs:enumeration value="fail"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:complexType>
```

exportAccessControlMapping API

This operation exports the users, permissions, user-roles, role-permissions from a source environment to an archive file.

There will be one XML file that will contain all the mapping information. Refer AccessControlMappingFileStructure type for details.

Input Parameters

- exportAccessControlMapping element contains the following:
  - type: string

    all is for users, permissions, user-roles, role-permissions

    user is for users, user-roles

    permission is for permissions, roles-permissions
  - filename: string (Archive file name)
  - location: string (Location to which the archive file will be stored for export)

Output Parameters

- accessControlMappingExportResponse element contains the following:
  - status: string (success or fail)

Schema

```
<xs:complexType name="userAttributesType" mixed="true">
    <xs:attribute name="id" type="xs:integer" use="required"/>
    <xs:attribute name="first_name" type="xs:string" use="required"/>
    <xs:attribute name="last_name" type="xs:string" use="required"/>
    <xs:attribute name="login_id" type="xs:string" use="required"/>
    <xs:attribute name="email_address" type="xs:string" use="required"/>
    <xs:attribute name="password" type="xs:string" use="required"/>
    <xs:attribute name="realm" type="xs:string" use="required"/>
    <xs:attribute name="active_flag" type="xs:string" use="optional"/>
    <xs:attribute name="inactivated_date" type="xs:date" use="optional"/>
</xs:complexType>

<xs:complexType name="userType" mixed="true">
    <xs:complexContent>
        <xs:extension base="userAttributesType">
            <xs:sequence>
                <xs:element name="roles" type="rolesType"
                    minOccurs="1" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>


    <xs:complexType name="usersType">
        <xs:sequence>
            <xs:element name="user" type="userType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="roleType" mixed="true">
        <xs:attribute name="id" type="xs:integer" use="required"/>
        <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>


    <xs:complexType name="rolesType">
        <xs:sequence>
            <xs:element name="role" type="roleType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="rolePermissionAttrType">
        <xs:attribute name="id" type="xs:integer" use="required"/>
        <xs:attribute name="name" type="xs:string" use="required"/>
    </xs:complexType>

    <!-- This element contains permissions on a specified resource type (ex: PEP, 
MessageType) -->
    <xs:complexType name="permissionAttrType" mixed="true">

        <!-- id is system generated identifier. This shouldn't be modified during export 
and import -->

        <xs:attribute name="id" type="xs:integer" use="required"/>
        <xs:attribute name="activate" type="xs:integer" use="required"/>
        <xs:attribute name="create" type="xs:integer" use="required"/>
        <xs:attribute name="delete" type="xs:integer" use="required"/>
        <xs:attribute name="deploy" type="xs:integer" use="required"/>
        <xs:attribute name="export" type="xs:integer" use="required"/>
        <xs:attribute name="import" type="xs:integer" use="required"/>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="read" type="xs:integer" use="required"/>
        <xs:attribute name="replace" type="xs:integer" use="required"/>
        <!-- resource_type is identifier of a resource type. This shouldn't be modified 
during export and import -->
        <xs:attribute name="resource_type" type="xs:integer" use="required"/>
        <xs:attribute name="type" type="xs:integer" use="optional"/>
        <xs:attribute name="update" type="xs:integer" use="required"/>
    </xs:complexType>

    <xs:complexType name="rolePermissionType" mixed="true">
        <xs:complexContent>
    <xs:extension base="rolePermissionAttrType">
        <xs:sequence>
            <xs:element name="permission" type="permissionAttrType"
                minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>


<xs:complexType name="rolepermissionsType">
    <xs:sequence>
        <xs:element name="rolepermission" type="rolePermissionType" minOccurs="1"
            maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>


    <xs:complexType name="AccessControlMappingFileStructureType">
        <xs:sequence>
            <xs:element name="users" type="usersType" minOccurs="0" maxOccurs="1"/>
            <xs:element name="rolepermissions" type="rolepermissionsType"
                minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>

        <xs:element name="accesscontrolmapping"
            type="AccessControlMappingFileStructureType"/>
```

importAccessControlMapping API

This operation imports the users, permissions, user-roles, role-permissions from the archive file to the target environment.

Input Parameters

importAccessControlMapping element contains the following:

- type: string
  - all is for users, permissions, user-roles, role-permissions
  - user is for users, user-roles
  - permission is for permissions, roles-permissions filename: string (Archive file name)
- filename: string (archive file name)
- location: string (location to which the archive file will be retrieved for import)

Output Parameters

accessControlMappingImportResponse element contains the following:

- status: string (success or fail)

Schema

```
<xs:complexType name="importAccessControlMappingType">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="type">
                <xs:simpleType>
                    <xs:restriction base="xs:token">
```

```
                        <xs:enumeration value="all"/>
                        <xs:enumeration value="user"/>
                        <xs:enumeration value="permission"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="filename"  />
            <xs:element name="location" />
        </xs:sequence>
    </xs:complexType>
</xs:complexType>

<xs:complexType name="accessControlMappingImportResponseType">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="status">
                <xs:simpleType>
                    <xs:restriction base="xs:token">
                        <xs:enumeration value="success"/>
                        <xs:enumeration value="fail"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:complexType>
```

Sample PMI Client

AMC version 2.4 includes a sample PMI client, located in **/opt/amc/admin/pmiclient**. The client consists of four files:

- PMIClient.java—The Java class. Modify this file to change testing parameters
- PMIHostNameVerifier—This is a helper class for PMIClient. It should not need modification
- cclient.sh—a shell script to compile PMIClient
- rclient.sh—A shell script to run a compiled PMIClient

Use the following workflow to test the PMI client:

1. Set up PMIClient.java to test the desired PMI command. Start with loginUser (see the PMI operation sections below).
2. Use cclient.sh to compile it.
3. Use rclient.sh to run it.
4. Look at the returned XML string to see if it worked or not.
5. Go to Step 1 and make changes to PMIClient.java until all variations of the PMI operations are tested.

Methods that test various PMI operations are invoked from main().

loginUser

- Change AMC_DEFAULT_PORT to be appropriate AMC Port if necessary.
- Change AMC_DEFAULT_HOST with the name or IP address of the AMC

- Change main so that none of the PMI command methods is called except for doPMIServiceLoginUser().

You should not have to change doPMIServiceLoginUser() method. Note that this method must be called for all PMI operations.

getProjectList

- Change main so that getProjectList() is called.
- Nothing to change in getProjectList() as getProjectList takes no parameters.

exportProject

- Change main so that callProjectExport() is called.
- Modify callProjectExport by changing “projectName”, “filename” and “location” variables.

importProject

- Change main so that callProjectImport() is called.
- Modify callProjectImport by changing “projectName”, “filename”, “location”, “mappingFile”, “mappingLocation” and “importType” variables.

getDRsList

- Change main so that getDRsList() is called.
- Modify getDRsList by changing “projectName” and “statusType” variables.

deployDR

- Change main so that deployDR() is called.
- Modify deployDR by changing “drid” variable.

callExportAccessControlMapping

- Change main so that callExportAccessControlMapping() is called.
- Modify callExportAccessControlMapping by changing “fileName”, “location”, and “type” variables.

callImportAccessControlMapping

- Change main so that callImportAccessControlMapping() is called.
- Modify callImportAccessControlMapping by changing “fileName”, “location”, and “type” variables.

Message Log Schemas

This section contains scripts that configure an Oracle or Sybase database for message log. For message log configuration instructions, see the “Message Log Domain” section on page 3-2.

Note Before running either of these scripts, be sure to remove any earlier versions of the message log schema.

Oracle Schema

Log in to SQLPlus as the user created for Message Log, then run this script to configure an Oracle database.

```
CREATE TABLE MESSAGE_LOG_INSTANCE
(   "USE_COUNT"    number(18,0),
    "ID"           varchar2(100),
    "DESCRIPTION"  varchar2(256),
    "VERSION"      varchar2(100)
);

INSERT INTO MESSAGE_LOG_INSTANCE (ID,USE_COUNT,VERSION,DESCRIPTION) VALUES
('AONS-MLOG-001', 0, '1.0', 'Database for storing AONS message logs');

CREATE TABLE MESSAGE_LOG
(
        "LOGID"        number(28,0) not null primary key,
    "HOSTNAME"                       varchar2(64),
    "SOURCE_NODE_ID"                 number(10,0),
    "ENTRY_TIME"     timestamp,
    "CREATION_TIME"  timestamp not null,
    "MESSAGE_ID"     varchar2(100) not null,
    "SESSION_ID"     varchar2(100),
    "DESTINATION"    varchar2(256),
    "NEXT_HOP"       varchar2(256),
    "SOURCE" varchar2(256),
    "SENDING_NODE"   varchar2(256),
    "FLOW_ID" varchar2(100),
    "BLADELET_ID"    varchar2(32),
    "FLOW_NAME"      varchar2(100),
    "BLADELET_NAME"  varchar2(100),
    "CONTENT_TYPE"   varchar2(64),
    "PAYLOAD_TYPE"   varchar2(32),
    "MESSAGE_TYPE"   varchar2(32),
    "MESSAGE_CLASS"  varchar2(64),
    "PROTOCOL"       varchar2(32),
    "LOG_VERSION"    varchar2(10),
    "LOG_TYPE"       varchar2(32),
    "LOG_LEVEL"      number(5),
    "SOAP_OPERATION"  varchar2(256),
    "STATUS"          number(10),
        "REASON"          varchar2(100),
    "PROTOCOL_HEADER" raw(2000),
    "CUSTOM_STRING1"  varchar2(32),
        "CUSTOM_STRING2"  varchar2(64),
        "CUSTOM_STRING3"  varchar2(128),
        "CUSTOM_STRING4"  varchar2(256),
        "CUSTOM_STRING5"  varchar2(1024),
    "CUSTOM_NUMBER1"  number(5,0),
    "CUSTOM_NUMBER2"  number(10,0),
    "CUSTOM_NUMBER3"  number(18,2)
);

CREATE INDEX "MESSAGE_ID_IDX" ON "MESSAGE_LOG" ("MESSAGE_ID");
CREATE INDEX "MESSAGE_DESTINATION_IDX" ON "MESSAGE_LOG" ("DESTINATION");
CREATE INDEX "MESSAGE_SOURCE_IDX" ON "MESSAGE_LOG" ("SOURCE");

CREATE TABLE SOURCE_NODE
(
    ID NUMBER(10) NOT NULL PRIMARY KEY,
    DN VARCHAR2(256) NOT NULL,
    CREATED_TIME NUMBER(18) NOT NULL,
    NODE_ID NUMBER(10),
    AMC_IP VARCHAR2(64),
    AMC_HOST_NAME VARCHAR2(256),
    AMC_ID VARCHAR2(256),
    AMC_VERSION VARCHAR2(64),
    VIRTUAL_NODE_ID NUMBER(10),
    VIRTUAL_NODE_NAME VARCHAR2(64)
);

create table MESSAGE_CONTENTS (
    LOGID   number(28,0) not null,
    CONTENT_TYPE    varchar2(64),
    NAME    varchar2(64),
    CONTENT  long raw,
    EXPRESSION varchar2(256),
        CONTENT_LENGTH  number(10,0)
);

CREATE TABLE FLOW_VARIABLES
(
    "LOGID" number(28,0) NOT NULL,
    "NAME"  varchar2(100),
    "VALUE" long raw,
    "TYPE"  varchar2(100)
);

CREATE SEQUENCE LOGID_SEQ
START WITH 1
INCREMENT BY 50000
NOMAXVALUE;

CREATE OR REPLACE PROCEDURE GET_LOGID_BLOCK  (
  blockSize out int,
  beginValue out number
 )
as
begin
  select LOGID_SEQ.nextval INTO beginValue from dual;
  select INCREMENT_BY INTO blockSize from USER_SEQUENCES;
  beginValue := beginValue - blockSize;
  return;
end;
/

select LOGID_SEQ.nextval from dual;
```

Sybase Schema

Log in to SQLAdvantage as the user created for Message Log, then run this script to configure a Sybase database.

```
CREATE TABLE MESSAGE_LOG_INSTANCE
(
    USE_COUNT    numeric(18),
    ID           varchar(100),
    DESCRIPTION  varchar(256),
    VERSION      varchar(100)
)

go
```

```
INSERT INTO MESSAGE_LOG_INSTANCE (ID,USE_COUNT,VERSION,DESCRIPTION) VALUES
('AONS-MLOG-001', 0, '1.0', 'Database for storing AONS message logs')

go


create table MESSAGE_LOG (
    LOGID                          numeric(28,0)                    not null primary key
,
    HOSTNAME                        varchar(64)                          null  ,
    SOURCE_NODE_ID                  numeric(10,0)                        null  ,
    ENTRY_TIME                      datetime                             null  ,
    CREATION_TIME                   datetime                         not null  ,
    MESSAGE_ID                      varchar(100)                     not null  ,
    SESSION_ID                      varchar(100)                         null  ,
    DESTINATION                     varchar(256)                         null  ,
    NEXT_HOP                        varchar(256)                         null  ,
    SOURCE                          varchar(256)                         null  ,
    SENDING_NODE                    varchar(256)                         null  ,
    FLOW_ID                         varchar(100)                         null  ,
    BLADELET_ID                     varchar(32)                          null  ,
    FLOW_NAME                       varchar(100)                         null  ,
    BLADELET_NAME                   varchar(100)                         null  ,
    CONTENT_TYPE                    varchar(64)                          null  ,
    PAYLOAD_TYPE                    varchar(32)                          null  ,
    MESSAGE_TYPE                    varchar(32)                          null  ,
    MESSAGE_CLASS                   varchar(64)                          null  ,
    PROTOCOL                        varchar(32)                          null  ,
    LOG_VERSION                     varchar(10)                          null  ,
    LOG_TYPE                        varchar(32)                          null  ,
    LOG_LEVEL                       numeric(5,0)                         null  ,
    SOAP_OPERATION                  varchar(256)                         null  ,
    STATUS                          int                                  null  ,
        REASON                          varchar(100)                         null  ,
    PROTOCOL_HEADER                 varbinary(2000)                      null  ,
    CUSTOM_STRING1  varchar(32)     null  ,
        CUSTOM_STRING2  varchar(64)                          null  ,
        CUSTOM_STRING3  varchar(128)                         null  ,
        CUSTOM_STRING4  varchar(256)                         null  ,
        CUSTOM_STRING5  varchar(1024)                        null  ,
    CUSTOM_NUMBER1  numeric(5,0)                         null  ,
    CUSTOM_NUMBER2  numeric(10,0)                        null  ,
    CUSTOM_NUMBER3  numeric(18,2)                        null
)

go


CREATE INDEX MESSAGE_ID_IDX ON MESSAGE_LOG (MESSAGE_ID)

go

CREATE INDEX MESSAGE_DESTINATION_IDX ON MESSAGE_LOG (DESTINATION)

go

CREATE INDEX MESSAGE_SOURCE_IDX ON MESSAGE_LOG (SOURCE)

go

CREATE TABLE SOURCE_NODE
(
    ID numeric(10,0) not null primary key,
    DN varchar(256) not null,
    CREATED_TIME numeric(18,0) not null,
    NODE_ID numeric(10,0) null,
    AMC_IP varchar(64) null,
    AMC_HOST_NAME varchar(256) null,
    AMC_ID varchar(256) null,
    AMC_VERSION varchar(64) null,
    VIRTUAL_NODE_ID numeric(10,0) null,
    VIRTUAL_NODE_NAME varchar(64) null
)

go

create table MESSAGE_CONTENTS (
    LOGID   numeric(28,0) not null,
    CONTENT_TYPE    varchar(64) null,
    NAME   varchar(64) null,
    CONTENT  image null,
    EXPRESSION varchar(256) null,
        CONTENT_LENGTH  int null
)

go


create table FLOW_VARIABLES (
    LOGID       numeric(28,0)   not null  ,
    NAME        varchar(100)    null  ,
    TYPE        varchar(100)    null  ,
    VALUE       image           null
)

go


create table LOGID_KEY (

    ID      numeric(28,0)  not null
)

go

insert into LOGID_KEY values (1)

go

CREATE PROCEDURE GET_LOGID_BLOCK
@blockSize int output,
@beginValue numeric(28,0) output
AS
  BEGIN
    select @beginValue=ID from LOGID_KEY
    select @blockSize=50000
    update LOGID_KEY set ID=@beginValue + @blockSize
 END


go


sp_procxmode 'GET_LOGID_BLOCK', chained
go
```