AdventNet

Excellence Matters

AdventNet Web NMS 2.3

http://www.adventnet.com
nms-support@adventnet.com

Table of Contents

About AdventNet Web NMS

Introduction

In the networked global village, more and more mission critical applications like e-business, ERP, CRM, supply chain management and collaborative applications such as distance learning and video conferencing are built on the network infrastructure. The key challenges faced by the enterprises and service providers are :

- Ability to move beyond offering reliable bandwidth to provide added value managed services. These services call for ubiquitous web access to management information that are independent of the countries or languages of its users and easy integration with a variety of applications as well as legacy systems.
- Rapid and flexible provisioning of services, coupled with ensuring that these services and the applications that run on top of them are highly available. It is absolutely vital to keep services highly available and accessible, as downtime translates into lost productivity, opportunities, revenue and profits.
- Investment protection in terms of ability of the management software to scale to very large configurations seamlessly.

The AdventNet Web NMS is architected with the above objectives in mind. A sampling of applications available on AdventNet Web NMS, categorized by specific domains are :

- ***Provisioning and OSS*** : end-to-end XML template driven service provisioning for service providers and equipment vendors.
- ***Core Network*** : optical and IP/ATM core.
- ***Metro Network*** : SONET/DWDM/Ethernet metro equipment.
- ***Edge and Access Network*** : cable, DSL, optical and wireless based broadband access technologies, with IP, ATM and SONET protocols.
- ***Storage*** : management for SAN and NAS topologies, with fibre-channel as well as IP based networks
- ***Security*** : firewall and VPN management.
- ***Applications Management*** : managing enterprise applications for high availability and performance.
- ***Extranets*** : managing supplier, customer and partner extranet applications.
- ***Managed services on subscription*** : enterprises can outsource network and applications management.

the list of Web NMS applications can go on and on. The following sections provide an overview and architecture of the AdventNet Web NMS and the standard technologies on which it is built.

About AdventNet Web NMS Platform

AdventNet Web NMS is an open, massively scalable, carrier-grade management infrastructure platform built for the Internet age. Leveraging the benefits of N-tier Internet based applications architecture, it provides unparalleled out-of-the-box application functions combined with tremendous flexibility to customize and adapt the framework for a variety of domain specific needs. The AdventNet Web NMS is built to address the needs of :

- ***The Systems Analyst & Developers*** : Architects solution(s) based on the AdventNet Web NMS; specializes and customizes the core product for a specific domain.
  The Web NMS facilitates building and deploying management solutions across distributed, heterogeneous computing environments, with intuitive, easy-to-use, powerful and ubiquitous interfaces.
- ***The Systems Administrator*** : who applies the out-of-the-box Web NMS solution for managing the mission critical infrastructure including the networks, systems and applications. Also, configure and administer the Web NMS solution to suit business and service needs including integration with OSS and other decision support systems.
- ***The NOC User (Operator)*** : Network Operations Center (NOC) users are those who use the product extensively to guarantee service levels. The NOC users typically need rich GUI functionality, to effectively manage the infrastructure availability and performance needs, with minimal training needs.
- ***The End User*** : who is primarily interested in self-service provisioning of services and service level reports. The end users prefer web based access to the system.

The following diagram provides an overview of the AdventNet Web NMS platform

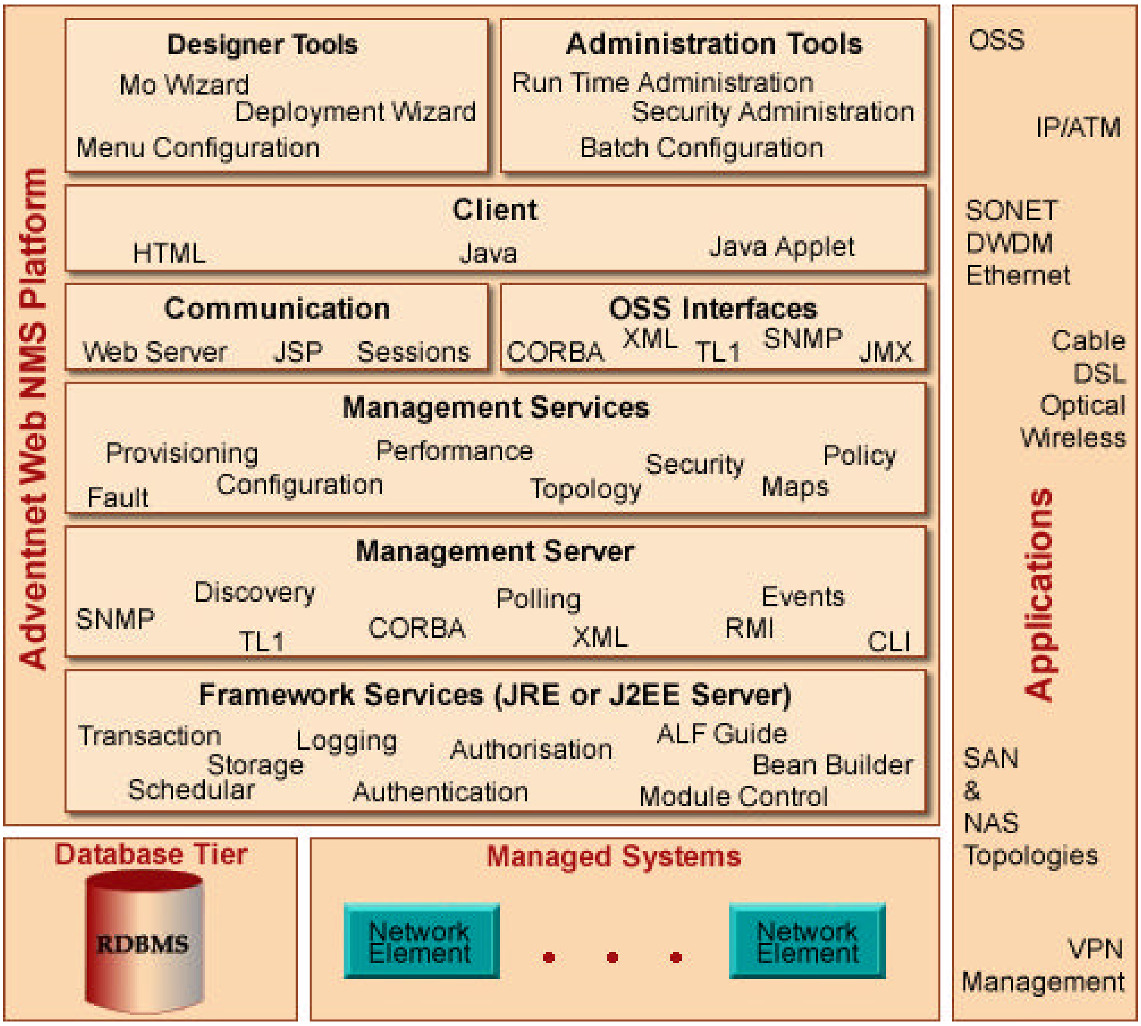


Fig 1 - AdventNet Web NMS Platform Overview

AdventNet Web NMS provides the necessary infrastructure, with a comprehensive suite of J2EE and JMS based cross platform development framework, tools, modules, APIs as well as pre-packaged applications. The Web NMS leverages EJB, JMS, JFC, XML, JMX, HTTP, JSP, JDBC, JTA, CORBA and other standards to deliver the most comprehensive management solution available anywhere. To communicate with managed systems (network elements, systems and applications) it supports SNMP, TL1, XML, CORBA, Telnet/CLI protocols out-of-the-box. To facilitate integration with OSS and other decision support systems, the Web NMS support north-bound interfaces like CORBA, RMI, JMX, SNMP, TL1, HTTP etc.

The features of AdventNet Web NMS that facilitate the above include

Open standards & Cross platform : The AdventNet Web NMS is built on standard technologies like EJB, JMS, JFC, JDBC, LDAP, HTTP, RMI, CORBA, SNMP, TL1, JMS, JMX etc. Given the power and ease-of-use of these new Internet technologies, developers can now provide better solutions faster and at a lower cost than ever before. The Web NMS supports deployment with various third-party components as follows :

- ***OS platforms*** : Solaris, Windows-2000, Windows-NT, Linux are supported out-of-the-box. HP-UX and IBM AIX support available.
- ***Application servers*** : WebLogic, JBoss and several others.
- ***Web servers*** : Apache, Java Web Server, Netscape, IIS and several others.
- ***Database*** : Any RDBMS that provide a JDBC driver are supported. Oracle, MS SQL Server, Sybase, Informix, SOLID, My SQL and others.
- ***Web Browsers*** : Internet Explorer, Netscape.
- ***Java Runtime Environments (JRE)*** : JRE 1.2.2, JRE 1.3.1.
- ***XML Parsers*** : Any JAXP 1.1 compliant parser.
- ***Language support*** : Internationalization and localization support built on JDK 1.3 Internationalization API.

Comprehensive product life cycle support : The Web NMS includes tools that provide complete product life cycle support for your management solution. The range of tools provided includes:

- ***Designer Tools*** : tools like MOWizard, Discovery Configurator, Menu Configurator, ObjectToRelational etc. that make it easy to build a management solution.
- ***Testing & Debugging*** : Services like Logging and the various interfaces provided in the system make it easier to debug your application. Tools bundled with the Web NMS like MibBrowser, TL1Browser, Status Viewer etc. make it easy to identify the faulty component in the system. Also other AdventNet products like Simulation Toolkit and AutoTest Toolkit make it easier to simulate and regression test your management application.
- ***Packaging & System Testing*** : Tools like Deployment Wizard make it easier to package the application, including providing a way to deploy the application in the lab for system testing. The management application can be packaged along with the different components of Web NMS, based on your requirements. There is no need to package all the components and services of Web NMS. You can also package your application without the Web NMS platform, so that the application can be installed on an existing platform installation.

- ***Deployment*** : The application can be deployed in many different ways, depending on your requirements. This makes it easy for you to build your management application once and deploy it in many different ways.
- ***Maintenance & Support*** : The update manager component bundled with the Web NMS along with the service pack creation tools, make it easier to support your management application.
- ***AdventNet Support*** : AdventNet is committed to extending excellent support through out your product life cycle, from design through deployment and support. Our support and maintenance programs have been designed to enhance your experience with AdventNet products.

Massive Scalability & High availability : The Web NMS server consists of three tiers of server components, separate from the RDBMS. These include the management server tier which provides protocol mediation with the managed systems, the back-end server tier which provides database transaction and encapsulation service and the front-end server tier which provides client session management services. The front end-tiers are supported to have a large number of clients and the management servers support management of large networks, achieving scalability on both the size of the networks and systems that can be managed as well as the number of clients that connect to the servers. There can be more than one deployment of each one of these servers to achieve massive scalability. The servers can be deployed with some redundancy so that any failure results in fail-over to other server components, thus making the Web NMS solution highly available.

Personalization & Customization : The Web NMS platform is a comprehensive development environment which supports a very high degree of personalization and customization. These include :

- ***modelling managed systems*** : The network elements, systems, applications and other managed entities can be modelled using a simple, easy to learn and extensible information model that can be extended to suite the needs of the management solution.
- ***Extending the management services*** : Management services like discovery, topology, network maps and FCAPS and provisioning functions are integrated into the platform. All the management services are easy to configure, customize and extend, to meet your management requirements.
- ***Management protocols*** : Web NMS supports SNMP, TL1, CORBA, XML and Telnet/CLI out-of-the-box. The Management server also provides a simple provider interface that facilitates integration of other management protocols.
- ***Rich set of deployment options*** : Once the management solution is built, the same solution can be deployed in a number of ways i.e. craft management interface, single server deployment on JRE (or) on J2EE server like WebLogic, distributed server deployment on JRE (or) on J2EE servers. Each one of these deployments can support the Java application client, the Java applet client on a web browser and the HTML client, at the same time. Also, the management solution can be deployed in a variety of OS platforms, without any porting (or) migration issues.

Web NMS Application Deployment Architecture

AdventNet Web NMS enables building of scalable, high-performance, cross-platform management solutions by providing a multi-tier, database-driven platform to build your applications. This section explains the different application architectures you could have i.e. single server architecture & distributed server architecture.

Single Server Application Architecture

When you are looking to build a management application which has

- Less than 10,000 managed objects being managed.
- Less than 10-20 clients (depending on the type of client and the load) will be exercising the server at the same time.
- The data collection, status polling and event processing requirements are modest.

You should architect your management solution to follow the single server architecture. The following diagram gives an overview of the single server architecture.

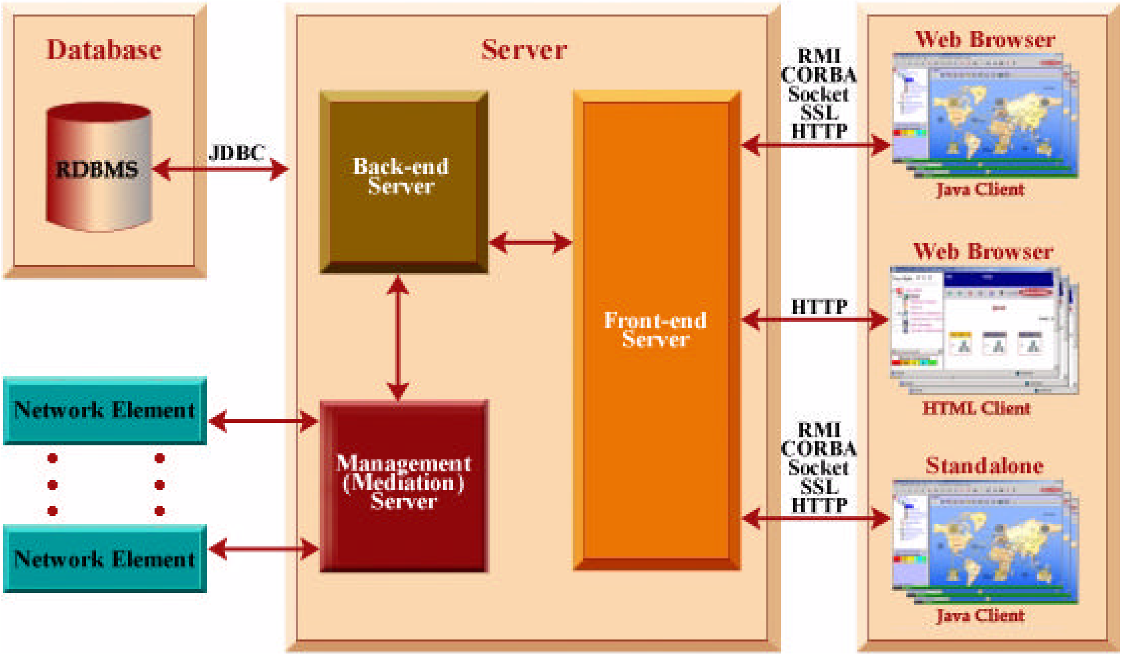


Fig 2 - Single Server Application Architecture

In the single server application architecture, the middle tier of the management solution (servers) built on the Web NMS is deployed to run in one Java Virtual Machine (JVM) process. This could be deployed on a JRE (Java Runtime Environment) or in a J2EE server environment (like WebLogic or JBoss).

Distributed Server Application Architecture

When you are looking to build a highly scalable management solution where a large number of clients will be accessing the management data from the NOCs, you should architect your solution on the distributed server architecture. The following diagram gives an overview of the distributed server architecture.

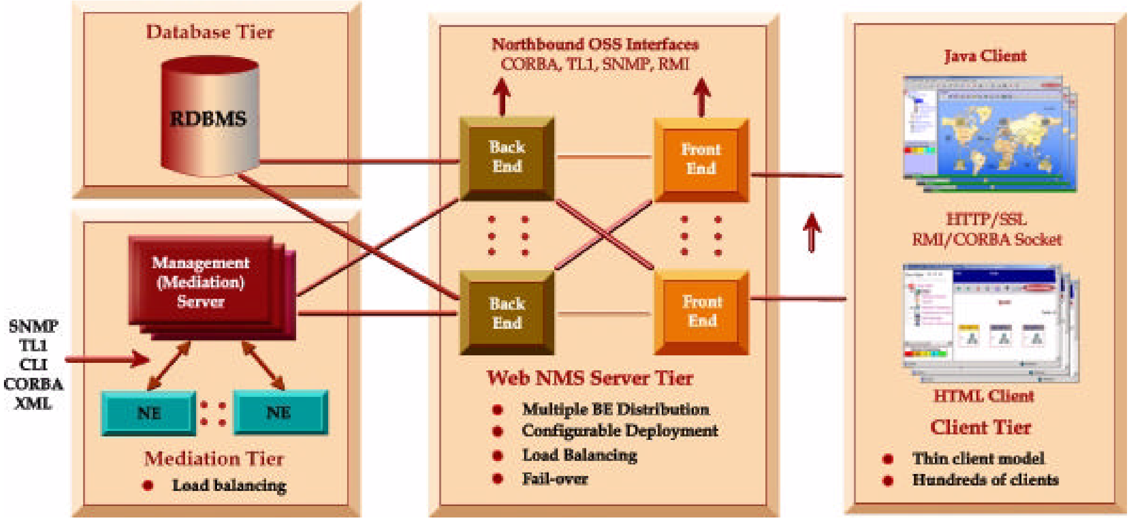


Fig 3 - Distributed Application Architecture

In the distributed server architecture, the middle tier of the management solution is distributed to run on multiple systems on the network. There can be multiple front-end servers to scale to a large number of clients. Based on scalability requirements, the back end server can also be run in a single server or multiple servers. There can be multiple management servers, based on the data collection, status polling and event processing capabilities, required out of the management solution. The servers support fail-over and load balancing capabilities. There is a choice of rich, thin clients based on Java or thin clients based on HTML. The client session management services are deployed in the front-end servers and they access the back-end business logic APIs using RMI or messaging.

When custom application management capabilities are integrated into the Web NMS, they should be deployed in all the servers. The deployment guide explains the details of deploying your application in a distributed environment to achieve scalability.

Standards Compatibility

The following table provides an overview of the standard technologies on which the Web NMS is built.

Standard Technology	Version(s) Supported	Key Benefits
EJB	1.1	EJB simplifies development of scalable, portable and transactional middleware components. The EJB facilitates building deployable components where the components of an application can be independently deployed and managed (as opposed to Java Beans components that are design time components).
XML	1.0	XML facilitates an efficient data interchange between components of an application and between applications by providing an extensible mechanism to describe the meaning and hierarchical structure of data.
JAXP	1.1	JAXP provides a standard API to process XML documents using DOM, SAX and XSLT. This makes it easier for applications to choose a XML parser implementation based on application need i.e. high performance parsers Vs. low memory foot print parsers etc.
JNDI	1.2	Java Naming and Directory Interface (JNDI) provides a unified interface to multiple naming and directory services, facilitating location transparency. The benefit of JNDI is that applications can locate the other services they are built on at runtime and use them (instead of assuming their availability in a pre-determined location).
JDBC	1.1 & 2.0	JDBC provides database vendor and technology independent (RDBMS, ODBMS etc.) and platform neutral API's to access data from multi-tier and client-server applications.
Java RMI	1.0	RMI facilitates distributed computing so that distributed applications with high scalability and performance goals can be built.
CORBA	2.3	CORBA provides a programming language independent distributed object computing environment on which highly scalable applications can be built.
HTTP	1.0 & 1.1	HTTP facilitates a standard communication mechanism between Web Browser Client and the Web Server. HTTP is also used as communication mechanism between other client and servers.

Standard Technology	Version(s) Supported	Key Benefits
JMX	1.0	Java Management Extensions (JMX) provides the tools for building distributed,Web-based, modular and dynamic solutions for managing devices, applications and service-driven networks. The benefit of JMX is that it makes the management information available through multiple interfaces such as SNMP, CORBA, RMI, TL1 etc.
LDAP	v2	The Light Weight Directory Access Protocol (LDAP) facilitates organizing information as a directory, where the information is read many times but rarely modified. LDAP is an ideal choice for having authentication and authorization information.
JFC	JDK 1.2 & JDK 1.3	Java Foundation Classes (JFC) is a collection of widgets, facilitates building user interfaces (GUI's) that have a platform independent visual presentation.
JSP	1.1	Facilitates having dynamic content in Web pages by allowing you to embed Java code in HTML pages (You can have your presentation and dynamic content separated by using Java Beans and JSP tags.
Servlets	2.2	Enables running classes in the server (through Servlets), based on inputs from the client. Presentation of dynamic response is also done from the Servlet.
TL1	1.0 (Highly Customizable)	Transaction Language 1 (TL1) is an ASCII based management protocol widely used in the telecom industry for managing the telecom infrastructure. Since TL1 is ASCII based, it can be used as both Human to Machine & Machine to Machine language, without any translations.
SNMP	v1, v2c & v3	The Simple Network Management Protocol (SNMP) is the most widely used management protocol, that defines the communication mechanism between the managed entities and the management station, the information model and the security and access control mechanisms for sharing management information.

What's Next?

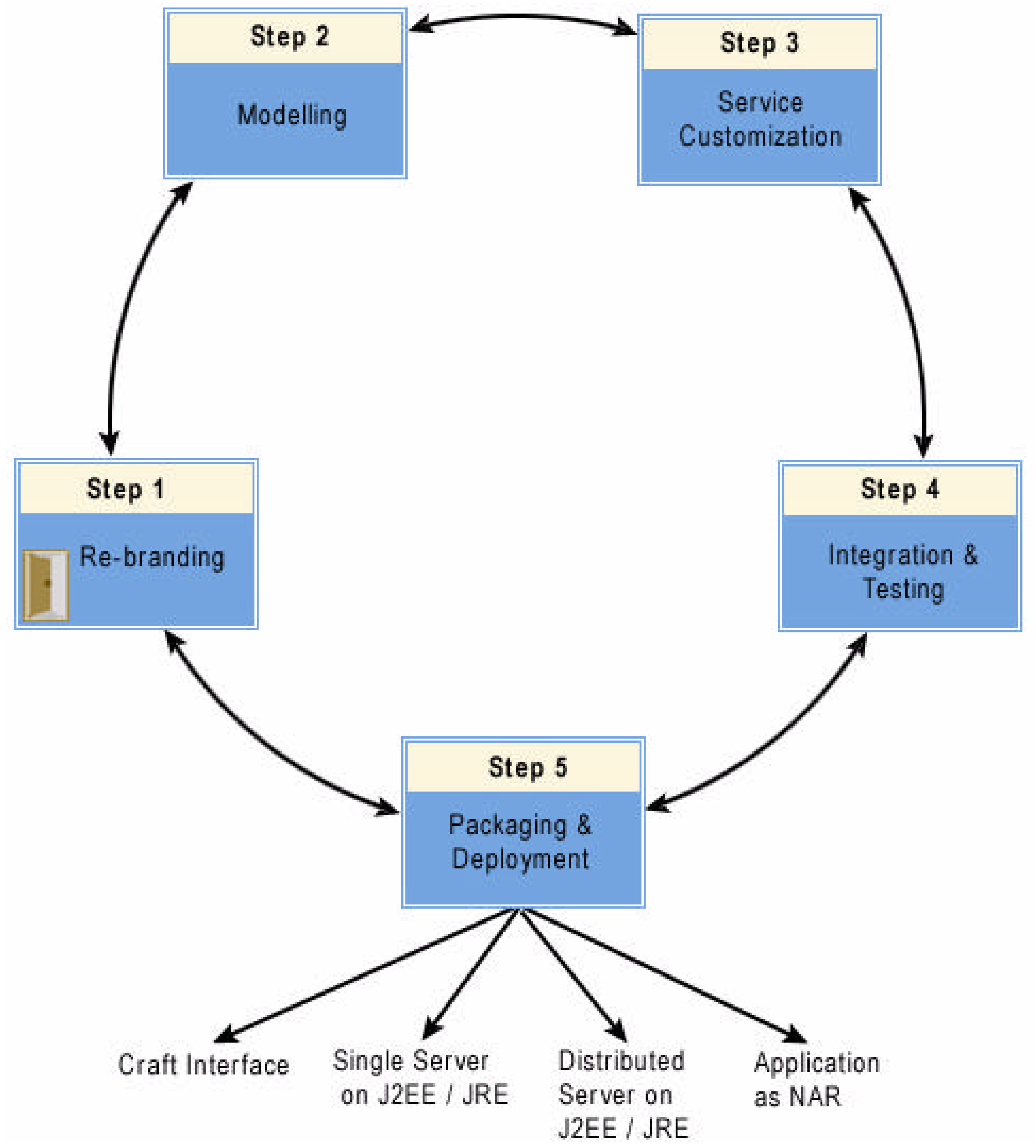
Download and give AdventNet Web NMS a try. We are sure, you will find AdventNet Web NMS as the most complete development environment giving you an unparalleled experience building management applications. To learn more about the Web NMS architecture and the features in the latest release, please read the Quick Tour. To jump start your development, go through the developer guide, tutorials and other developer resources in the product.

AdventNet Web NMS Experience - 5 Easy Steps to Build Management Solutions

AdventNet Web NMS offers a comprehensive development environment for building your management solution. The AdventNet Web NMS Experience explains how the complete product life cycle needs of your management solution are realizable using the product. They are captured in 5 easy steps, that you can follow to build your management solution, as below:

- **Step 1** : Re-branding the management solution
- **Step 2** : Modelling the managed elements
- **Step 3** : Customizing managed object services
- **Step 4** : Integration & Testing
- **Step 5** : Packaging and Deployment

The following diagram gives an overview of the experience of building management solutions with the AdventNet Web NMS



Step 1 : Re-branding the management solution

You can re-brand the application to display the name of your company, the name of your product and your logos. The i18n tool, packaged as part of the Designer Tools, helps you re-brand your application by replacing references to AdventNet and Web NMS. The logos, images and icons can be changed by modifying configuration files.

Step 2 : Modelling the managed elements

Each managed system comprises of many inter-related elements that need to be individually managed. You start with modelling your elements, so that you can capture the data, operations and state of the elements and the relationships between the elements. The Web NMS provides a comprehensive, simple and easy to learn information model, using which the various elements and hence the managed system can be modelled. The basic element of the Web NMS information model is the ManagedObject, which is explained in the managed object specification. The Web NMS also has models for various common IP network components like Network, Node, SnmpNode, TL1Node etc. These form the core objects of the Web NMS information model.

You should extend any of the core objects of Web NMS, to model your managed element. The core objects can be extended, by adding attributes, operations and state to those objects (modelling the data, operations and state of your element in addition to capturing the relationship)

This task can be easily accomplished by using the MO Wizard tool, packaged as part of the Designer Tools. It walks you through the steps in terms of the object that needs to be extended, the new attributes of your element etc. It then generates Java code and database schema files for your managed element. Once the element is modelled, a number of management services like discovery, status polling, data collection, maps etc. are available. Now that you have modelled the system you can proceed with customizing the various services offered by the Web NMS for the managed objects.

Step 3 : Customizing the managed object services

Web NMS offers a number of management services to the managed objects. The southbound services that populate the database with information from the elements like data collection, status polling etc., are classified as the mediation services. The services, that enable the user to perform network planning, error management and service deployment tasks are classified as the management services. Management services include event correlation, element configuration, service provisioning, access control etc.

Based on the above classification, step 3 is divided into customizing management services and customizing mediation services Using the module management services available as part of the Web NMS framework, you could also build other management application modules.

Step 4 : Integration & Testing

At the end of re-branding, modelling and service customization steps, you will have

Java source files: Those of the managed object(s) and the service customization. These should be compiled with the Web NMS and class files generated.

Modified configuration files : The config file checker tool should be used to identify any errors in the file.

Before attempting to compile, make sure the Java development environment is complete i.e. JDK (or) the J2EE/EJB development environment is available. For compiling and running the application ensure that the "classpath" has been set correctly. If you are deploying the application in a distributed architecture then the startup script files (shell or batch files, depending on your development environment) corresponding to the BE server, FE server and the client have to be modified correctly.

Once the Java source files are compiled, the database schema for the modelled managed objects and the service extensions should be generated. The object-to-relational tool can be used to generate the database schema. Alternately, the MOWizard can be used to compile the source files and generate the database schema for the same. The deployment wizard can be used to package the application for testing.

Before starting to test your management solution, make sure all the third-party packages are installed correctly and you have the required privileges to use them for your testing. Once you start your application, look at the Web NMS server log files to make sure all the services are started successfully and are running. You can have your application specific logs, using the Web NMS logging service, for information on the functioning of your application. The Web NMS logging service helps you debug and trouble shoot your application. The different parameters for logging like the logging levels, the number of log files and other log parameters are specified in a configuration file. New log files can also be created with the parameters and operations to be logged

AdventNet offers other products like Simulation Toolkit, which help in setting up a simulated test bed and AutoTest Toolkit, which helps in automating the functional and performance testing of your management applications.

Step 5 : Packaging and Deployment

You can package your application using the deployment wizard. It allows you to choose the package structure of the application, viz. package including the Web NMS platform as a zip file or package the application resources alone as a NAR (NMS Archive file) that can be installed over the Web NMS separately. As part of the packaging for deployment, you can specify the distribution licensing mechanism for your management application. Applications can be packaged as craft management console, single server deployment on JRE (or) J2EE, distributed server deployment on JRE (or) J2EE platform, depending on your application needs.

Having deployed your application at a customer's site, you will be required to support the product and provide upgrades as part of support. The tools available with the Web NMS make it easy to handle version management, upgrades, specials etc.

Customizing mediation services

South-bound protocol services : This service enables support for a number of south-bound management protocols in the Web NMS. Web NMS supports management protocols like TL1, SNMP, CORBA, XML, CLI/Telnet protocols out-of-the-box. You can also add custom protocol provider for supporting any other protocol.

Discovery service : This service involves discovering and adding the managed elements to the Web NMS database. Discovery service is configured to discover your managed elements using the discovery configurator tool. The discovery configurator tool allows you to specify various attributes of discovery service like IP address of the network/element, range of addresses in which the element should be searched etc. Managed elements can either be discovered or manually added from the client. A programmatic interface is also available to add managed elements. Web NMS provides the ability to perform custom operations like managed object attribute initialization, filtering etc. through well defined interfaces.

Status polling service: The Web NMS periodically polls the managed elements for their health status. Change in the status is automatically updated on the managed object and is propagated to the other managed objects, as per the modelled relationships. The client is also updated to show the status of the managed element / system. The time interval of the status polling service can be customized by modifying an entry in the configuration file.

Network notification service: Notifications from the systems like autonomous messages in TL1 or traps in SNMP are received and processed by the Web NMS. The port for receiving these notifications can be configured by modifications to appropriate entries in configuration file. Filters can be configured for these notifications to invoke actions. For example, you can use the Trap Parser configuration tool for invoking a filter that will query the system for more information related to the trap.

Data Collection service: The Web NMS periodically polls the systems for performance data. The time period for the data collection service can be configured through appropriate modifications in the configuration file. The collected data can be stored in the database, as per application requirements. Various reports can be generated based on the collected data.

Internal notification service: Operations within Web NMS can generate notifications which can be processed further. A simple example of these notifications are events generated by thresholds that have been configured on the systems for performance monitoring.

Customizing management services

Event Correlation Service: This service automatically correlates many errors from a system into a single alarm, making error identification and resolution faster. You can configure the parameters for event correlation through the trap parser configuration tool.

Alarm management service: Alarms in the systems, resulting due to the status poll service or network notification service, need to be prioritized, assigned to users and rectified. You can specify filters on these events and alarms for invoking useful actions like sending email, running applications or suppressing the notification. Such filters can be configured through the GUI or through tools like event filter configuration tool and alarm filter administration tool. To simplify error management you can create custom views for filtered display of the events and alarms or annotate the alarms with additional information, etc.

Report generation service: Web NMS can generate reports and graphs, of the data collected through the data collection service, for trending and historical analysis. You can configure the reports to display device specific parameters, the duration of the data collection, statistical averages etc. This customized report can be generated by making modifications in the configuration file or by accessing the database through the programmatic interfaces.

Threshold service: Using this service you can enable performance degradation notifications which aid in network planning and preventive maintenance. You can configure at runtime a threshold setting for a system through the appropriate menu item in the GUI. When the system performance crosses the threshold setting a notification will be generated to alert the user of the degraded performance.

Configuration service: The Web NMS provides this service to modify system configuration for optimizing its and the network's performance. You can leverage its task driven approach to configure and persist system configuration data. Through the Element configuration UI you can pre-configure the tasks and choose the systems which need to be configured. The persisted task, saved as an XML file, can be modified and applied to other such systems.

Software distribution service: Software upload and download service aids the network administrator in managing systems inventory. You can configure a task through the GUI for uploading and downloading the software to a TFTP enabled system. The software can be a new application, a OS upgrade patch etc.

Provisioning service: The Web NMS provides the capability to provision services. You can integrate the systems configuration service with other services, filters and forms to develop a service. A simple example is to provision a DHCP service by which a new system in the network will be assigned an IP address automatically. This service involves 3 system configuration tasks, forms for inputting other information and filters.

Authentication service: This service enables the management of user authentication. Through the user administration tool you can add and delete users, assign and modify passwords. Based on the authentication a number of user privileges can be defined.

Access control / Authorization service: Using this service a network administrator can define user privileges for secure and effective management of the systems. Through the security administration GUI you can define groups with the authorization for views and operations. Then you can associate users to these groups. Thus you can easily configure any user's views and operations.

Mapping service: This service defines the design and layout of the systems, symbols, background and links on the GUI. You can modify a configuration file to choose the symbols and backgrounds on the map. The layout of the map can be modified by choosing the appropriate menu item or you can define your own layout through the programmatic interfaces.

Operator interface service: Using this service you can build a customized view of the systems and their components that enable easy management. You can define the node on the navigation tree and the corresponding map or list view in the configuration file. Through a form you can specify the position of the nodes on the navigation tree. The menu items in the menu bar can be defined through the menu configurator tool. New screens can be built in the Bean builder tool and incorporated into the GUI. Thus you can build a custom GUI in quick, easy steps.

AdventNet Web NMS Architecture

Goals

AdventNet Web NMS architecture is designed to fulfill the following goals.

Market Goals : The Web NMS is designed to address the needs of the following markets

- ***OEM*** : Reduce time-to-market for OEMs and leverage latest technologies to deliver best-in-class element and network management solutions.
- ***Enterprises*** : Provide excellent application integration to make it easy to have a managed enterprise and extranet infrastructure.
- ***Service Providers*** : Make it easy to offer added value managed services on which mission-critical services are offered and easy migration on the service value chain.

Technology Goals : The Web NMS is designed with the following key technology goals.

- ***RDBMS centric design*** : Makes it easy to distribute the solution for scalability and performance needs. Also offers other major benefits such as transaction encapsulation, high availability, ease of performance tuning, sizing etc.
- ***Open, cross-platform, modular and multi-tier components***: The architecture is designed to leverage middle tier component technologies such J2EE/EJB. The benefits of leveraging such technologies result in enhanced productivity and investment protection. It results in lower total cost of ownership, as the cost of tools, testing/QA and maintenance are lower.
- ***Loose coupling, stateless and XML messaging based servers***: The distributed server components are loosely coupled, making it easy to handle component failures. This, together with the stateless and XML messaging based servers, make it easy to handle fail-over resulting in high availability of management solutions.
- ***Separation of database views & database commits***: The generation of database views are handled in a separate front-end server(s) and all database commits are forwarded to the back-end server(s). This results in providing massive scalability in the number of clients that can be supported. The database tables are normalized so that each back-end server handles the commit for certain logical database. This result is massive scalability in the number of managed systems that can be managed.
- ***Thin and rich clients***: The model is created from the database making the clients thin. Since the complete database is available for access and control from the client, it makes the interface really rich. The application GUI can be built from the bundled IDE tool and also existing ones customized from the same, making it easy to personalize the client.

Management solution Goals: The Web NMS is designed with the goals of providing the following key management solution capabilities.

- ***Multi-protocol support***: The management server mediation framework for the southbound protocol mediation and the JMX agent for the northbound protocol mediation make it easier to support multiple protocols such as CORBA, HTTP, RMI, SNMP, TL1, CLI/Telnet etc. It is also easy to integrate any of your management protocol into the list of supported ones.

- ***Deployment options*** : Build once, deploy many ways support using which the same management solution can be deployed as a craft management console (or) as an EMS / NMS system. The solution can also be integrated with OSS, provisioning and other decision support systems such as capacity planning tools etc.
- ***Application integration*** : It is easy to integrate domain specific applications such as SONET, WDM, DSL, Optical, SAN etc. The support for application packaging independent of the platform, enables easy installation and uninstallation of applications on the platform.
- ***Evolution with business growth*** : The management solution can grow along with business growth, where components and services can be added to the deployed solution to meet the performance, scalability and availability goals.
- ***Ease of administration*** : The Web NMS has a rich set of administration tools, which make the task of administration including service provisioning, reconfiguration, performance tuning etc. easy.
- ***Maintenance & upgrade*** : Web NMS is designed for easy maintenance and fast, low-risk upgrades.

Components Overview

The following diagram gives an overview of the different components in the Web NMS and the interactions between them.

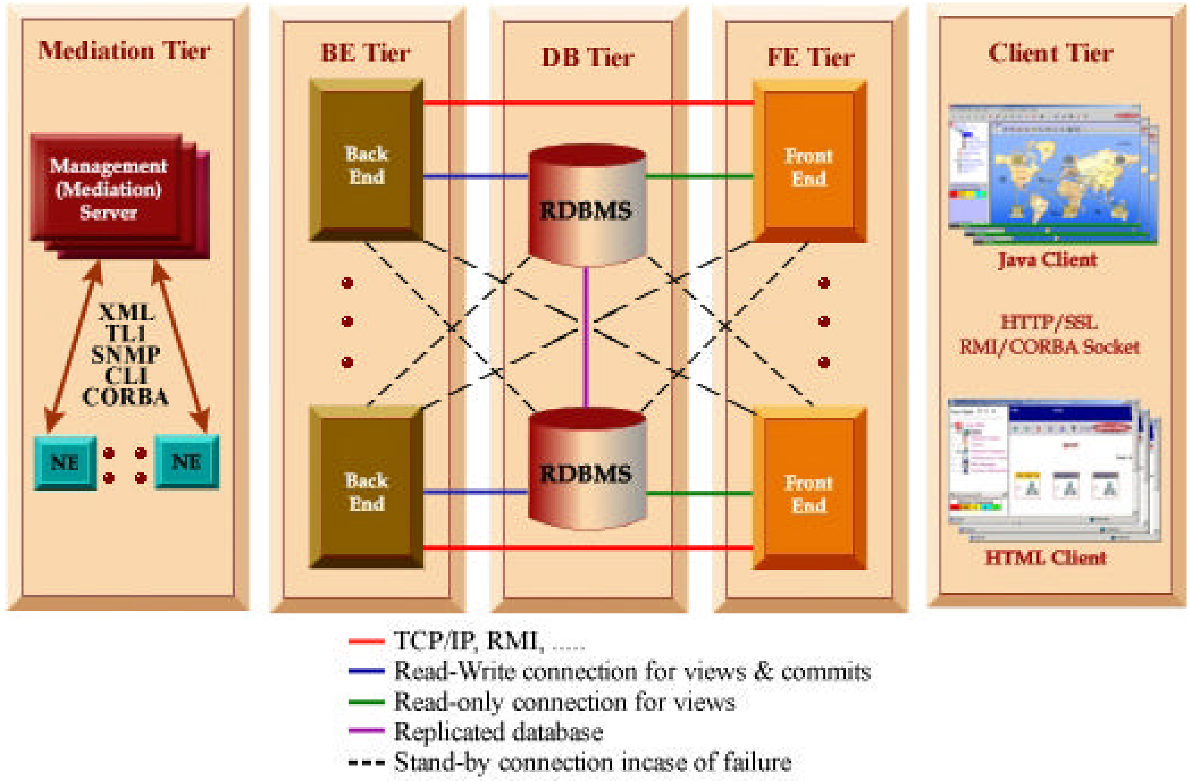


Fig - AdventNet Web NMS Platform Architecture

Each one of the components of this n-tier design can be distributed. Also, there can be one or more components of each type, based on the performance, scalability and availability requirements of the deployed solution. An overview of the key features and benefits of each component is given below.

Database tier : Any RDBMS that provides a JDBC driver are supported. The key benefits of the database tier are

- Enables quick, focused application development around data due to the availability of tools and expertise.
- State can be maintained in database, making it easier to distribute the different components and handle fail-over to meet the high-availability goals.
- Leverage the benefits of transaction support, database synchronization and object locking for data integrity, security and availability.

Management server tier : The management server provides XML mediation for all southbound management protocols like SNMP, TL1, CORBA, TFTP, XML, CLI/Telnet etc. The benefits of the management server mediation tier are

- Simple provider interface facilitates easy integration of any other protocol support. Supports dynamic deployment and mediation of new protocols.
- The common management functions are handled in the mediation layer, making the protocol providers thin with protocol specific operations only.
- Provides multiple interfaces for application development like XML message interface, Java API etc. Supports many transports out-of-the-box such as TCP, RMI, CORBA, HTTP etc. Also, provides powerful registration capabilities for accessing the services of the mediation tier.

Back-end server tier :The back-end server tier consists of the core business logic related to management functions like fault, configuration, performance, security, service provisioning etc. The benefits of the back-end server tier are

- ***Rich out-of-the-box functions*** : The core management modules support advanced functions like event correlation, alarm notification and management, template based provisioning, batch configuration and rollback etc. These modules are highly customizable and extensible.
- ***Module management*** : The back-end tier provides a powerful module control interface that facilitates starting and stopping of different modules independently.
- ***Security & Audit*** : All the back-end tier modules support authorization and audit, so that it is easy to trace the various administrative operations that result in provisioning, reconfiguration etc.

Front-end server tier : The front-end tier consists of the web container that provides web access to management information, the client communication management module and the session beans for the different management functions, that generate views for the clients and forward commit requests to the back-end tier. The benefits offered by the front-end server tier are

- ***Client communication*** : Option to choose from a range of transport protocols like TCP, RMI, HTTP, HTTPS, SSL etc. Flexibility to have the transport of your choice, with the powerful transport provider interface.
- ***Session beans*** : The stateless, EJB deployable session beans directly generate views from the database, based on client requests. Session beans can be customized to generate any view, based on requirements, making it easy to build rich clients. Various database tools, bundled along with the product, make it easy to generate various views for the clients.

- ***Updates to client*** : A powerful subscription based notification model, making it easy for the FE to handle very high rates of updates to the client. In this model, clients register for updates (they are interested in) and the FE server notifies them on any change in data.

Client tier : Multiple options like Java client (for rich GUI) and HTML client (for web access over low speed links) are supported. The key features of the client are

- ***Rich set of operator and administrator functions*** : The client facilitates administrative tasks like user administration, discovery configuration, security administration, audit logs etc. It also provides excellent operator interface for management information in terms of fully functional network maps, alert administration, performance reports etc.

- ***Extensive customizability*** : Configuration files facilitate customization of menus, toolbars, tooltip-text, fonts, severity, color, icons etc. The client can be extended by the integration of custom management applications such as NMS Panels, NMS Frames, etc. The AdventNet Look and Feel Guideline (ALF Guide) makes it easy to build interfaces that are consistent with the Web NMS client interface.
- ***Multi-lingual clients*** : The support for internationalization and localization makes it easy to build and deploy client interfaces that are independent of language, country, time zone etc.

Designer Tools : The Web NMS is packaged with excellent tools for application development.

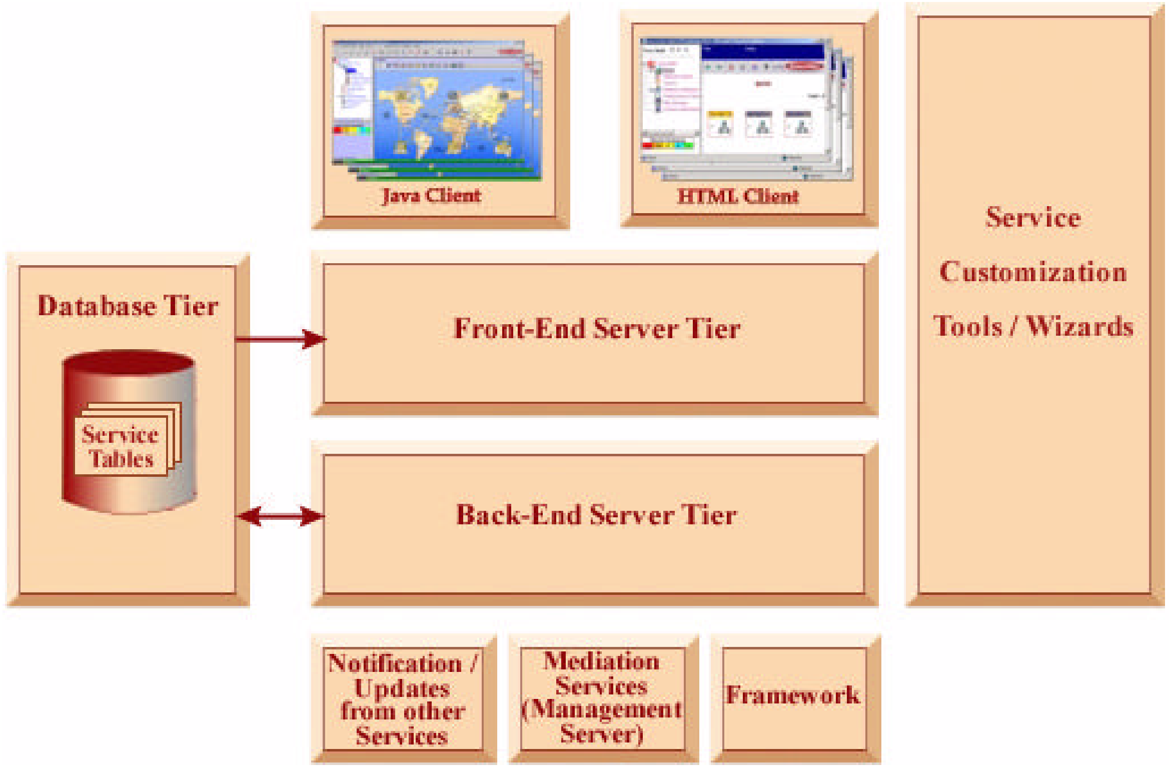
- ***modelling of Management Systems*** : MOWizard facilitate modelling of managed systems, using which the data, operations and the state of managed systems can be modelled.
- ***Service customization tools*** : The discovery configurator, menu configurator, status polling configuration etc. help in management service customization.
- ***Client development and customization*** : The Bean Builder bundled with the Web NMS provides an IDE environment where application GUI can be built. It also facilitates customization of the Web NMS client.

Web NMS Management Services Architecture

Web NMS management services are built on the N-tier architecture model provided by the framework services modules. Each service consists of data modelled as database tables and maintained in the database tier, back-end and front-end tier server components, a client tier with the administrator and operator interfaces and the tools required to customize and extend the services. The goals of this service architecture are

- Functions of different modules in each tier is clearly defined , making it easy to build new services, applications. Each module is built to leverage relevant standard technology such as J2EE, EJB, XML, JFC, JNDI etc.
- Customizable & extensible business logic layer, making it easy to create required views and operations on the management information.
- Stateless business logic layer supports fail-over and high availability. Also maintaining states in the database facilitates other applications of management information.

The following diagram provides an overview of the architecture of the Web NMS management services



Java Client
HTML Client
Service Customization Tools / Wizards
Database Tier
Service Tables
Front-End Server Tier
Back-End Server Tier
Notification / Updates from other Services
Mediation Services (Management Server)
Framework

Fig. 1 - Web NMS Management Services Architecture

The various modules and components of each service, along with how they are distributed across the different tiers of the Web NMS framework are explained.

Database tier

The database tier hosts the data, modelled as different database tables, for each Web NMS service. The table data includes the states and status of the different components of a service, facilitating easy testing of applications for data integrity and transaction behavior. The table design also captures the relationship with tables in the other services i.e. integrity constraints, foreign keys etc.

Back-end server tier

The key component of the back-end server tier is the business logic module which performs the various operations of the service such as processing the requests from the service users and processing the events from the mediation services and the other services. The processing results in state transitions, status updates, notifications to service users and commits to the database tables. Some of the important aspects of the processing module are the stateless server design, transaction encapsulation for handling database commits including the commits across the different services and service customization / extension interfaces. Using the service customization interface, the service capabilities can easily be extended and customized to suit management solution requirements

The following diagram provides an overview of the architecture of the back-end server tier of all Web NMS services.

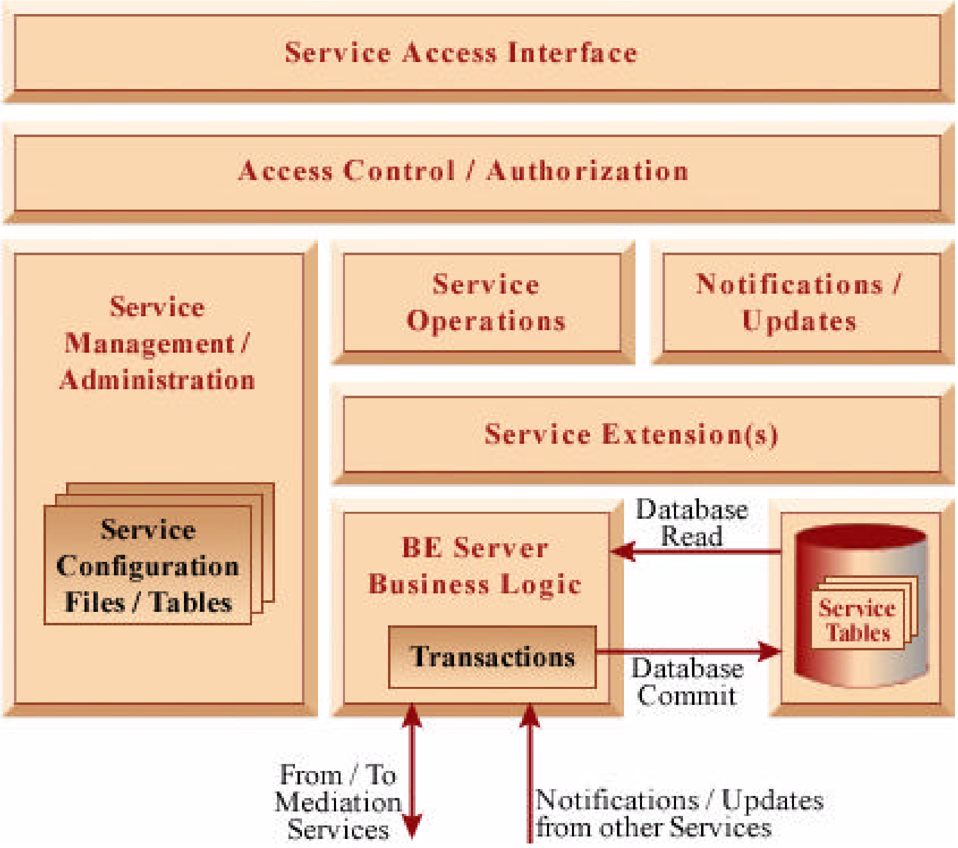

Service Access Interface
Access Control / Authorization
Service Management / Administration
Service Configuration Files / Tables
Service Operations
Notifications / Updates
Service Extension(s)
BE Server Business Logic
Transactions
Database Read
Database Commit
Service Tables
From / To Mediation Services
Notifications / Updates from other Services

Fig. 2 - Back-end server tier of Management Services

The back-end server tier also includes other modules such as service management / administration module, the notification interface, service operations interface, the access control / authorization module and the service access interface. The service management /

administration module facilitates extensive configuration of the service and it includes XML configuration file(s) related to the service configuration. The service access interface provides multiple ways of accessing the service such as Java API, RMI, CORBA and message based access.

Front-end server tier

The front-end server component of each service is built on the database and back-end tiers. The business logic module of the front-end tier directly works with the data available in the database tables to create the different views based on user requests. It uses the back-end server for any operation that will result in commits to the database and for the service management / administration requests. The back-end services are accessed using RMI and the message based interface provided by the back-end server module. The FE business logic layer does not maintain any state and uses extensive caching of data to improve responsiveness. It also facilitates creating views based on tables belonging to different services, by providing extension interfaces. The FE business logic module is built on the subscription based notification model, which results in service scalability in terms of the number of clients that can be handled and the number of updates that can be processed

The following diagram provides an overview of the architecture of the front-end server tier of all Web NMS services

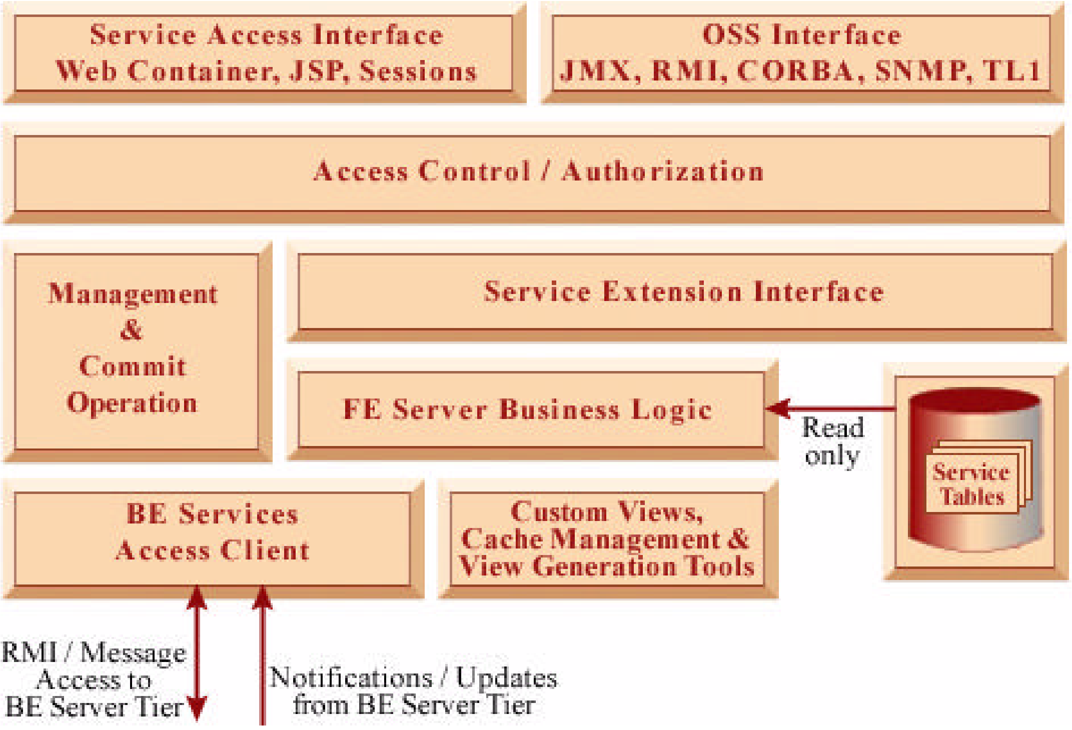

Service Access Interface
Web Container, JSP, Sessions
OSS Interface
JMX, RMI, CORBA, SNMP, TL1
Access Control / Authorization
Management & Commit Operation
Service Extension Interface
FE Server Business Logic
Read only
Service Tables
BE Services Access Client
Custom Views, Cache Management & View Generation Tools
RMI / Message Access to BE Server Tier
Notifications / Updates from BE Server Tier

Fig. 3 - Front-end server tier of Management Services

The front-end server tier also includes other modules such as the view generation tools, web container, session management module, the access control / authorization module, service access interface and the north-bound interfaces. The service access interface facilitates customization of the presentation layer and generation of views to suit management solution requirements

Client tier

The client tier is built on the services offered through the service access interface of the FE server module. The Java clients access FE services using a message interface over transports like TCP, HTTP, SSL, RMI etc. The HTML client accesses FE services through the web container interface provided by the FE.

The following diagram provides an overview of the architecture of the back-end server tier of all Web NMS services.

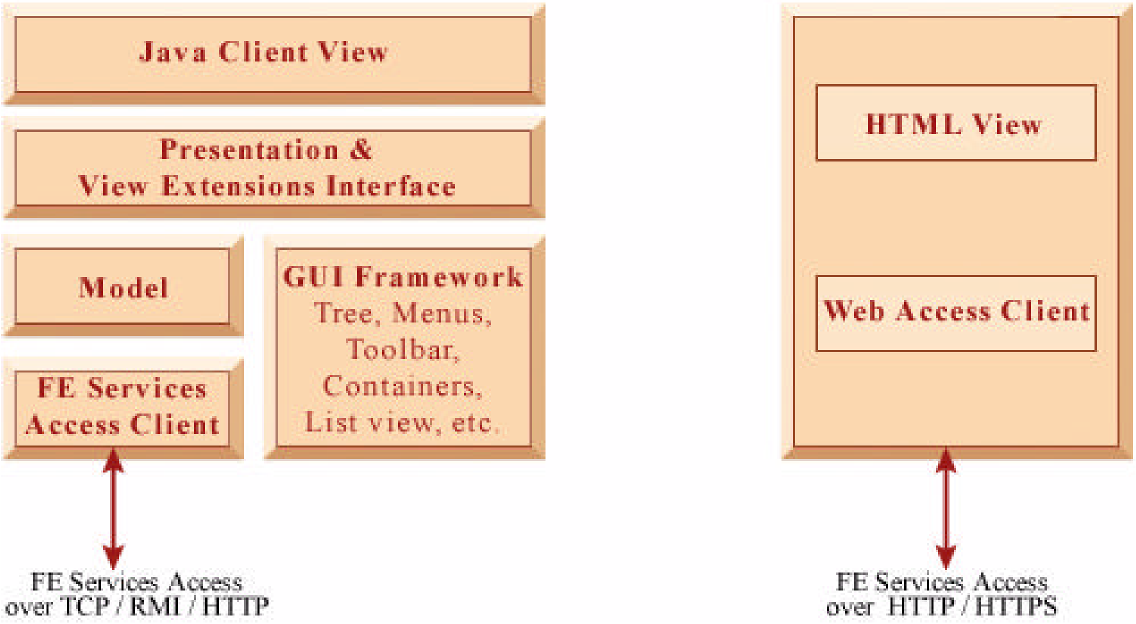

Java Client View
Presentation & View Extensions Interface
Model
GUI Framework Tree, Menus, Toolbar, Containers, List view, etc.
FE Services Access Client
FE Services Access over TCP / RMI / HTTP
HTML View
Web Access Client
FE Services Access over HTTP / HTTPS

Fig. 4 - Client tier of Management Services

The core of the Java client interface is the presentation of management information to suit management solution requirements. The Java client provides an extension interface that facilitates extensive customization of the presentation logic. The Java client is built on the ALF Guide, resulting in consistent user experience. All the parameters of ALF Guide can be configured, making it easy to customize the view to suit management solution requirements.

Web NMS Mediation Services Architecture

The Web NMS mediation tier provides a framework where various management protocol specific south-bound functions can be integrated to offer a protocol neutral interface to the management applications. The mediation tier, available as the Management Server component, offers multiple interfaces for building applications such as transport independent XML messaging, Java API and RMI/Corba API access. The protocol neutral interface, together with the rich set of out-of-the-box services offered by the mediation layer make it possible to build management services that are independent of the management protocol.

The following diagram provides an overview of the architecture of the Web NMS mediation services.

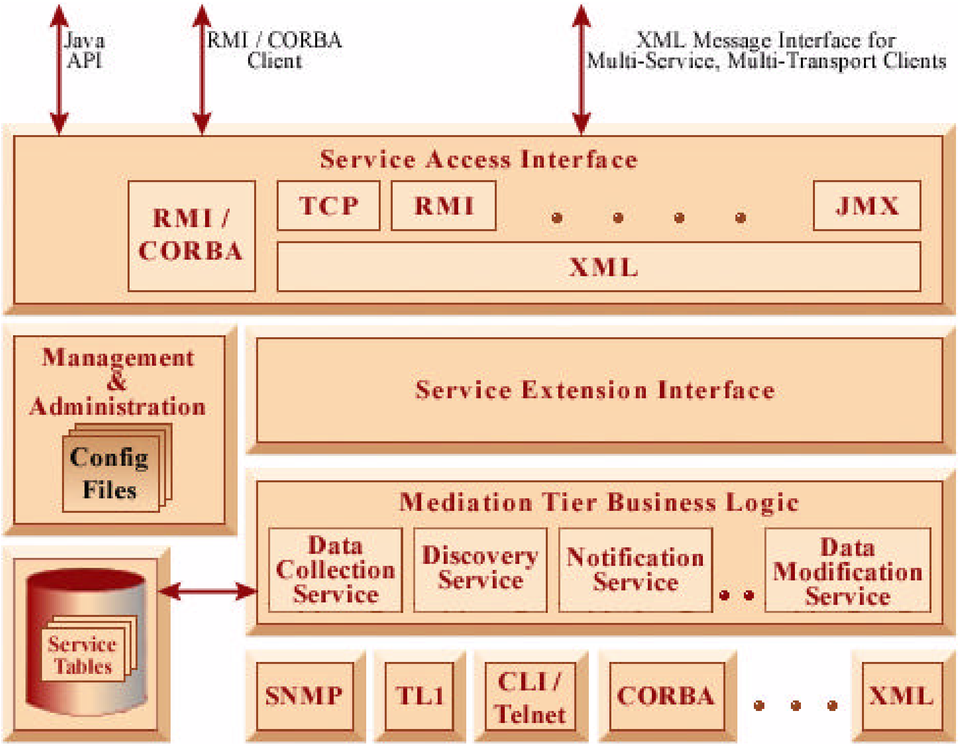


Fig. 1 - Web NMS Mediation Services Architecture

The core business logic module of the Management Server component offers services such as data collection (READ), data modification (WRITE), notification / asynchronous event processing, discovery etc. It offers a protocol provider interface using which various south-bound management protocols such as SNMP, TL1, CORBA, XML, RMI etc. can be integrated. The business logic module supports hot deployment of the south-bound providers, and automatic configuration of its services for the new providers. Using the service extension interface, all the business logic services can be extended to suit management solution requirements

The service management and administration module facilitates deployment of south-bound protocol providers and configuration of the business logic and protocol providers attributes. The service access interface makes it easy to build multi-vendor, multi-device, multi-protocol management solutions, by providing a protocol independent synchronous (request/response) and asynchronous (subscription based notification driven model) communication. The service access interface module also provides XML mediation and messaging over transports such as TCP, RMI, CORBA, and JMS etc. Other transport protocols, as per management solution requirements, can be integrated with the service access interface module.

What's New in AdventNet Web NMS Release 2.3

- For System Analysts and Developers
- For System Administrators & Operators

For System Analysts and Developers

J2EE Development

Web NMS supports management application development on the J2EE platform by leveraging core J2EE technologies such as EJB, JAXP, JNDI, JDBC, JSP etc. This results in scalable, reliable and modular solutions, at a lower cost. The Web NMS applications can be developed over any J2EE platform such as WebLogic, JBoss etc. Web NMS also supports JRE development, where applications can be developed and packaged without a J2EE server. One key benefit of Web NMS is that management applications are independent of the Java platform on which it is run i.e. the J2EE (or) the JRE platform. The Web NMS framework services module provides the required encapsulation.

New & Improved Help Documentation

The help documentation is totally revamped with a revised table of contents (TOC). The new TOC makes it easy to navigate the help documentation and locate topics of your interest. Also, the index keywords available with the document is greatly enhanced, making it easy to reference the topic of your interest. Many diagrams are included as part of the documentation, making it easier to grasp the high-level picture of the architecture, design and implementation of modules / features. The documentation includes specifications like MO specification, AdventNet Look and Feel Guideline (built on Java Look and Feel Guideline), many tutorials to use the product and an extensive glossary of terms.

Provisioning Service

Web NMS provides a provisioning module that facilitate building south-bound protocol neutral multi-device, multi-vendor service provisioning applications. The Web NMS provisioning module provides the necessary interfaces to facilitate integration with other service management, CRM and OSS applications. Another major benefit of the provisioning module is that the configuration tasks, defined to fulfill configuration management requirements, can be used as is, to build the provisioning application. The configuration management tasks perform all the management protocol specific functions and provide a management protocol neutral interface for the provisioning applications.

Designer Tools

Web NMS 2.3 includes a number of tools that will speed-up your development and make a significant impact on the time-to-market of your applications. The tools include:

- **i18n** - Tool that helps in the re-branding of the management solution. Also, makes it easy to internationalize your application.
- **MO Wizard** – Helps in modelling of managed element / system. Also includes wizards for customization of few services like status polling, discovery etc.

- **Bean Builder** - Provides an IDE environment for building your application screens and facilitates Java client customization. Also, makes it easy to package the application screens for Web NMS integration.
- **Object to Relational Wizard** - Facilitates the generation of database schema for your application information model.
- **Deployment Wizard** – Enables packaging of the Web NMS application for deployment.

Configuration Management

Web NMS provides a comprehensive configuration module that enables building configuration applications. The template based, task driven module facilitates simple scripting of most operations. The configuration module is built on the management mediation framework (management server) resulting in support for multiple south-bound protocols. Key configuration management functions like software download, file transfer, batch configuration of multiple managed entities, transaction and rollback etc. are supported. South-bound protocols like TL1, SNMP, CLI/Telnet, XML, CORBA are supported. Comprehensive audit functions are available, using which the configuration operations can be tracked. A pleasant, feature rich and customizable client is packaged as part of the Web NMS. The client can be customized using the bean builder, for which the bean builder projects are supplied along with the product.

Security Management

The security management module facilitates in providing secure and controlled access to management information. The authentication module in Web NMS is used to authenticate users while they login to the system and the authorization module does access control, while they access management information. Security module also maintains an audit of all operations that are authorized, making it easy to track operations on the management system. An easy-to-use and customizable administration tool for user administration and access control configuration is packaged along with the product. The client can be customized using the bean builder, for which the bean builder projects are supplied along with the product.

Multi-protocol Mediation Framework (Management Server)

Web NMS management services are built on the multi-protocol mediation framework. The common management operations like scheduling data collection at pre-defined intervals, threshold event generation based on collected data, network notification processing etc. are handled in the mediation layer. This makes the protocol provider layer very thin, making it easy to build and integrate any new protocol support. The management server supports TL1, SNMP (v1, v2c and v3), CLI/Telnet, CORBA and XML out-of-the-box.

For System Administrators & Operators

Fail-over, hot-standby & high availability

Web NMS provides hot-standby capability with automatic switch-over, thus enhancing the reliability of the application. The system works by polling for activity - i.e. the stand-by server polls to ensure the primary is active, and takes over when it finds it is not, by marking a table in the database. The primary will check to see if it still has control, and shuts down if not. The back-end server switch-over is detected by the front-end servers, which automatically switch-over to the new back-end. Fail-over is supported for all the server components. The fail-over capability, along with database replication provides highly available deployment.

Runtime Administration

Most of the Web NMS services can be administered at runtime, using the runtime administration tool. Runtime administration is supported for services such as discovery, maps, status polling, data collection, event correlation etc.

Northbound Support

Web NMS is packaged with a JMX technology based agent module that facilitates SNMP, TL1, CORBA, HTTP, RMI and XML access to the management information. The Web NMS agent provides access to all the management information available in the Web NMS database. The Web NMS application itself can be monitored and managed using the north-bound interface. The powerful range of functions available through the north-bound interface, makes it easy to integrate the Web NMS with any OSS, provisioning and other higher layer applications. It also helps build a hierarchical management solution, with distributed management responsibility.

Enhancements in the Client

The operator interface in the Web NMS has been substantially enhanced to provide great user experience. Key enhancements include enhanced search capabilities, ability to browse for resources on the server, extensive context sensitive help, many enhancements to the map etc. Both the Java UI and the HTML UI are greatly improved for usability, performance and look and feel. The Java Client is built on the AdventNet Look and Feel Guideline (ALF Guide), resulting in consistent user experience. The ALF Guide makes it easy to build applications that are consistent with the Web NMS GUI.

Internationalization

Internationalization support in Web NMS has been substantially enhanced. All the components of the management solution can be internationalized and localized using the i18N tool. Administrators can customize the management solution to suit the country, language and cultural needs of their users.

Contacting AdventNet

Technical Support

One of the value propositions of AdventNet to its' customers is excellent support. To extend this commitment our support and maintenance programs have been designed to enhance your experience with AdventNet products from development to deployment. During the evaluation phase the support program is extended to users "free of charge".

The hallmark of our maintenance and support is rapid and knowledgeable response to customer queries, from our product developers-cum-support specialists. To enable this level of support, consistently and comprehensively, we have built an in-house Support management system. This system enables automatic segregation of queries, authorized views, tracking and assigning of queries to the specialists and many other advanced features.

To leverage the benefits of this support management system and enable faster resolution of your issues we have designed mail aliases corresponding to the product modules. If you believe that your issue can be best addressed by a particular module, then send your query to the corresponding mail-id. Do NOT worry if you are not able to identify the module to which the issue has to be addressed. In such a situation just send the mail to our good old nms-support mail id. Provided below is the table that gives the mail aliases corresponding to the product modules.

Number	Product Module / Service	Mail ID
1	Any Web NMS product related suggestions (or) issues	nms-support@adventnet.com **Note** : Please mail the appropriate module/services team, for quick response
2	Discovery Services	discovery-support@adventnet.com
3	Topology Services	topology-support@adventnet.com
4	Network Mapping Services	map-support@adventnet.com
5	Fault Management Services	fault-support@adventnet.com
6	Configuration Management Services	configuration-support@adventnet.com
7	Performance Management Services	performance-support@adventnet.com
8	Security management Services	security-support@adventnet.com
9	Provisioning Services	provisioning-support@adventnet.com
10	Designer Tools	tools-support@adventnet.com
11	Protocol Mediation Services	protocol-services-support@adventnet.com
12	Web NMS Documentation issues	nms-doc-support@adventnet.com
13	Documentation related suggestions / issues for all AdventNet products	doc-support@adventnet.com

Following is the support format to be enclosed, while sending support mails

- **Version of the product** : Web NMS2.1/Web NMS2.2/WebNMS 2.3 etc.
- **Operating System** **:** Solaris/Win NT/Win95/Win98/Linux etc.
- **JDK/JRE Version** : (JRE1.3 or JRE1.2 or JDK1.1.x) etc.
- **Browser** : (Netscape/IE)
- **Browser Version**
- **Details of the problem** **:** (a) The relevant log files, (b) The steps to reproduce the problem.
- **Stack Trace (if any)**
- **Debug messages or Thread Dumps (if any)**
- **Log files (if any)**
- **CLASSPATH environment variable (if applicable)**

Feedback : If you have comments or suggestions about the Product, Customization features, API design, Java documentation and Help Documentation. please feel free to email us. While sending feedback please include the following information.

- Topic Title.
- Brief description of content (for example, are you reviewing step-by-step instructions that are inaccurate, grammatical errors in a specific paragraph, information that requires clarification or more details, etc.).
- Your suggestion for how to correct/improve documentation.

Please send email messages to nms-support@india.adventnet.com

Mailing List : The mailing list java-nm@adventnet.com is a resourceful forum for interacting with network management developers. To subscribe send mail to majordomo@adventnet.com with no subject and a message body: subscribe java-nm.

Commercial : To contact AdventNet please visit http://www.adventnet.com/contact.html