NASTEL
Driving Business Transaction Performance ®

Nastel AutoPilot® M6 User's Guide

Version 6

Document Title: Nastel AutoPilot® M6 User's Guide
Document Release Date: May 2022
Nastel Document Number: M6/Usr 630.009
Product Release: 6.0.32

Published by:
Research & Development
Nastel Technologies, Inc.
88 Sunnyside Blvd, Suite 101
Plainview, NY 11803

Table of Contents

Chapter 1: Introduction

Welcome to the Nastel *AutoPilot M6 User's Guide*. This guide describes AutoPilot M6 management platform, terms, concepts, architecture, as well as use and administration of AutoPilot M6 services, servers, and components. Nastel AutoPilot M6 will hereinafter be identified as AutoPilot M6 or M6.


NOTE

1. JDK 1.5 is required to run AutoPilot M6.
2. Service updates are non-cumulative service packs, and usually require a certain level of a service pack installed.
3. The term *managed node* used in versions 1 through 5 of AutoPilot has been changed to *CEP Server* to differentiate it from managed systems.

1.1 Use of this Document

This guide is intended for anyone administering or using an AutoPilot M6 product including AutoPilot M6 Enterprise, AutoPilot M6™, and AutoPilot M6 WMQ. The guide assumes that the user is familiar with general system and application management concepts, networking, and TCP/IP.

1.2 Guide Organization

Chapter 1: Introductory information relevant to the document and AutoPilot M6.

Chapter 2: Contains a brief description of AutoPilot M6 architecture, terms, and concepts.

Chapter 3: Provides instructions for operation and deployment of AutoPilot M6 and M6 Web Console.

Chapter 4: Contains task-oriented information and instructions for administering AutoPilot M6.

Chapter 5: Describes customization options of various AutoPilot M6 components.

Chapter 6: Describes general troubleshooting techniques.

Chapter 7: Gives solutions to some common AutoPilot M6 problems.

Appendix A: Provides detailed list of reference information required for the installation of AutoPilot M6.

Appendix B: Contains conventions used in AutoPilot M6 and document typographical conventions.

Appendix C: Defines AutoPilot M6 command line interface.

Appendix D: Contains AutoPilot M6 Best Practices.

Appendix E: Defines required Linux Platform Configurations.

Appendix F: Provides M6 Dashboard Database Schema.

Appendix G: Describes Web Service Interface.

Appendix H: Provides a list of derived metrics.

Glossary: Contains a listing of unique and common acronyms, words, and definitions.

Index: Contains an alphanumeric cross-reference of all topics and subjects of importance.

1.3 History of This Document

Table 1-1. Document History

Release Date	Document Number	AP M6 Version	Summary
August 2007	APM6/USR 600.001	6.0.1	Initial Release
April 2008	APM6/USR 600.002	6.0.2	Service Updates 1, 2, 3 and 4
July 2008	APM6/USR 600.006	6.0.6	Service Updates 5 and 6
August 2008	APM6/USR 600.008	6.0.7	Service Update 7
May 2009	M6/USR 600.009	6.0.8	Service Update 8
December 2010	M6/USR 600.010	6.0.8	Recording tab updates
July 2012	M6/USR 600.011	6.0.17	LDAP authentication
April 2013	M6/USR 650.001	6.0.19	Mantis 7536, update for version 6.0.19
July 2013	M6/USR 600.019	6.0.19	Added URL Monitor description (4.5.1.2)
November 2013	M6/USR 600.020	6.0.20	Added additional LDAP Server node.properties (Table 2-2)
March 2014	M6/USR 600.021	6.0.20	Errata (Mantis 8688 and 9361)
November 2014	M6/USR 600.022	6.0.21	Errata (Mantis 9758) and remove references to M6 Reports which are no longer supported (Mantis 10141)
January 2016	M6/USR 622.001	6.0.22	Update for Service Update 22 (Mantis 2635, 4817, 8619, 9218, 10007, 10285, 11021, 11044, 11656, 12027)
June 2016	M6/USR 623.001	6.0.23	Update for Service Update 23 (Mantis 4618, 6396, 6620, 7500, 10523, 10594, 12495, 13455, 13799, 13874, 14161, 14229)
January 2017	M6/USR 624.001	6.0.24	Update for Service Update 24 (Mantis 10054, 10241, 12951, 13696, 15088)
July 2017	M6/USR 625.001	6.0.25	Update for Service Update 25 (Mantis 14130, 14933, 15389, 15585, 15611, 15701, 15707, 15881)
September 2018	M6/USR 625.002	6.0.25	Updated step 6 of section 4.4.3.1.
February 2019	M6/USR 625.003	6.0.25	Clarify LDAP information.
July 2019	M6/USR 625.004	6.0.25	Added information on apnet "ignore" option to table C-2.

Table 1-1. Document History

Release Date	Document Number	AP M6 Version	Summary
August 2019	M6/USR 625.005	6.0.25	Update "server.domain.ldap.<ldapserver1>.user.class" in Table 2-2.
September 2019	M6/USR 625.006	6.0.25	Refresh index section.
November 2019	M6/USR 630.001	6.0.30	Update step #21 in section 4.9.2.2 (add information on changing e-mail behavior). Updates to document layout.
March 2020	M6/USR 630.002	6.0.30	Add LDAP configuration content to first bullet of section 2.4.1 and Java Keytool commands to section 2.4.2.2. Update copyright year.
April 2020	M6/USR 630.003	6.0.30	Updates throughout appendix C4.
January 2021	M6/USR 630.004	6.0.30	Update AP Insight name to XRay.
April 2021	M6/USR 630.005	6.0.30	Change reset_pwd to reset-pwd in Table C-2.
May 2021	M6/USR 630.006	6.0.31	Update %Cause in tables 4-31 and 4-34.
May 2021	M6/USR 630.007	6.0.31	Update section 4.7.5 (Health and Load Balancing Policy), 4.9.2.2 step 21. Update Table 5-2.
June 2021	M6/USR 630.008	6.0.31	Update "Database user ID" in the Log Sensor to Database table (Table 4-41).
May 2022	M6/USR 630.009	6.0.32	Added Internet Explorer caveat to System Requirements. Updates to Streaming Data Logging Policies.

1.3.1 User Feedback

Nastel encourages all users of AutoPilot M6 to submit comments, suggestions, corrections, and recommendations for improvement for all AutoPilot M6 documentation. Please send your comments via mail or *e*-mail. Send messages to: support@nastel.com. You will receive a written response, along with status of any proposed change, update, or correction.

1.4 Related Documents

Complete listings of documents related to AutoPilot M6 can be found in Appendix A.

1.5 Release Notes

See `README.HTM` file on installation media and root installation directory.

1.6 Intended Audience

This document is intended for personnel installing, customizing, and using AutoPilot M6. The user who installs the product should be familiar with:

- J2EE (Tomcat, WebLogic, WAS) for AutoPilot M6 Web deployment
- Java Run Time Environment 1.7 (JRE 1.7) or later
- Basic understanding of TCP/IP

1.7 System Requirements

All software and hardware requirements are defined in *AutoPilot M6 Installation Guide*.

All references in this document to Internet Explorer are subject to Internet Explorer being available. M6 console works with Internet Explorer if it is available. Customers who plan to move away from this older functionality are encouraged to consider using the equivalent functionality that is available within Nastel XRay. For each component, XRay's sensors can provide metrics about message processing and point to backlogs in the processing pipeline by showing lag times in message streaming and indexing.

1.8 Terms and Abbreviations

A list of Terms and Abbreviations used in this document is located in the Glossary.

1.9 Technical Support

If you need additional technical support, you can contact Nastel by telephone or e-mail. To contact Nastel technical support by telephone, call **800-963-9822 ext. 1**, if you are calling from outside the United States dial **001-516-801-2100**. To contact Nastel technical support by e-mail, send a message to support@nastel.com. You can also contact Nastel support via the support website. To access the Nastel automated support system (user ID and Password are required) go to: http://support.nastel.com/. Contact your local AutoPilot M6 Administrator for further information.

1.9.1 The Resource Center

The Resource Center is where AutoPilot M6 solution users solve problems, exchange ideas, and learn best practices from peers and Nastel staffers. This online community is a service provided by Nastel Support. Our experts often participate in these discussion groups to share their advice, but these groups are intended as a peer-to-peer resource.

The Resource Center also provides access to downloads, updates, documentation, support articles, product news, and a lot more. Registration and access are free to all users. We encourage everyone using Nastel products to join Nastel Resource Center at http://www.nastel.com/resources.

1.10 Conventions

Refer to Appendix B for conventions used in this guide.

Chapter 2: About AutoPilot M6

2.1 Introducing AutoPilot M6

AutoPilot M6 is designed to monitor and control distributed IT services such as application servers, middleware, user applications, workflow engines, brokers, Service Oriented Architecture (SOA) and Enterprise Service Bus (ESB) based applications and their impact on business services. It reduces time and effort required to monitor IT services end-to-end without the need for custom development, complex configuration, and customization. M6 is based on SOA and can run on a variety of platforms. It employs a policy-based approach to monitoring across heterogeneous platforms and applications.

(Complex Event Processor) CEP-based Event Stream Processing and Correlation engine is at the core of M6 technology. This engine allows aggregation, sorting, filtering, merging, and joining of various events and metric streams in real-time using a wizard driven GUI interface.

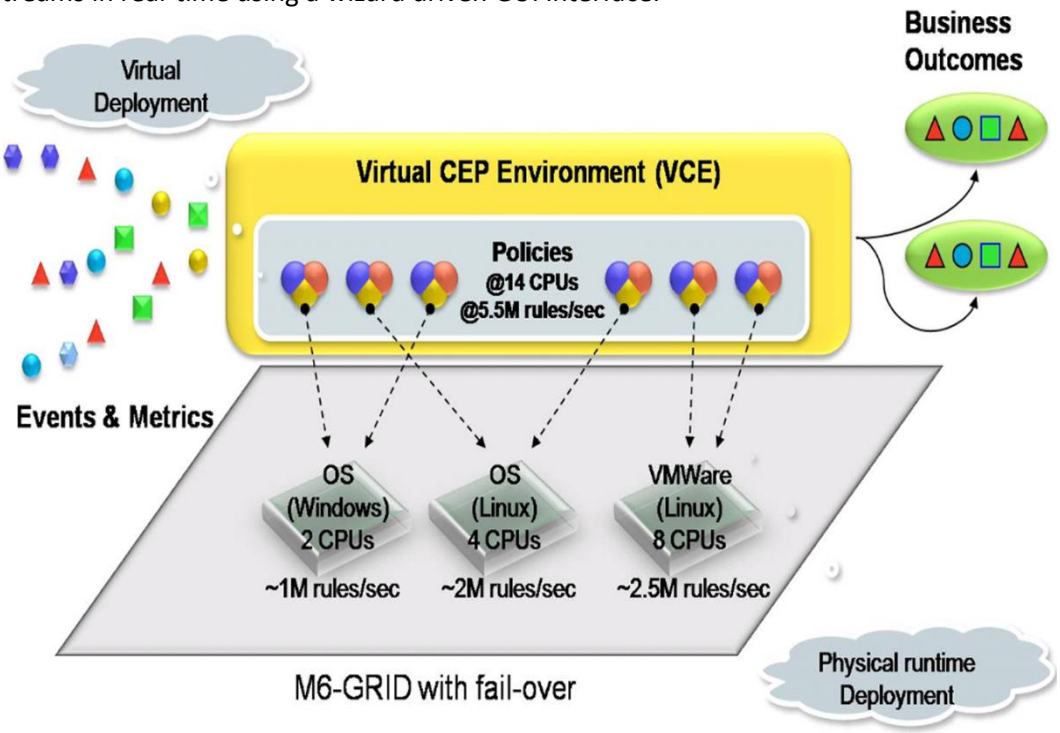


Figure 2-1. Virtual CEP

CEP instances can be replicated or auto-failed over with M6-GRID within minutes or even seconds and can be consolidated or distributed on one or more hardware or Virtual Machines (VM) -- such as VMWare or XEN. The replacement hardware/operating system does not have to be identical.

CEP has fault, performance, and security isolation. Each CEP instance is independent of one another. A crash or malfunction in one instance does not affect another instance. Its hardware includes CPU and memory isolation.

The entire state of the CEP is stored in a single deployment file. The entire CEP configuration can be transferred easily regardless of operating system, architecture, or processor.

M6 Virtual CEP Environment (VCE) does the following:

- Reduces time to install and configure a new monitoring domain from days/weeks to minutes
- Reduces the time to move a CEP instance to a new server or VM from days to minutes with uninterrupted operation.
- M6-GRID provides automatic CEP instances provisioning in case one or more are offline for maintenance or any other reason.
- CEP instances can simply be allocated on the fly without service interruptions.
- Resource allocations can be made dynamically.
- Policies can be reassigned and redeployed within seconds to any CEP instance.

M6 is highly effective in environments where one or more mission critical applications run by technologies such as J2EE, .NET, ESB, JMS, JMX, JMS Messaging, as well as homegrown applications.

M6 provides several types of monitoring services: *experts*, *managers*, *policies*, and *business views*. Experts are data collection components, which reside on CEP servers. Managers are distributed software components equipped with policies (components that proactively monitor and automate). Business views are top-level visual automation tools that can be deployed as and serve as policies.

M6 uses a collaboration model and collects all related system and application metrics (facts) from managed resources. It correlates the facts via intelligent business views, which can be shared among teams within the organization.

M6 provides security based on the mutual authentication between a client and a server, or between one server and another, before a network connection is opened between them.

It is important to note that understanding of M6 terms, concepts, and architecture is essential for M6 administrators. M6 introduces new terms that may not be commonly used in the systems and application management space. Understanding AutoPilot M6 architecture and organization becomes critical, since M6 employs a blend of *n*-tier and network architecture, which is contrary to traditional 3-tier application and system management platforms. This architecture allows M6 to scale beyond the limitations of standard 3-tier models and provide users with un-matched flexibility, scalability, and performance. Please review section 2.2, Terms, Concepts, and Architecture thoroughly before deploying M6 full scale.

2.2 Terms, Concepts, and Architecture

2.2.1 Concepts and Architecture

AutoPilot M6 SOA platform employs agent and agent-less technology with real-time, distributed event stream processing and correlation engine. M6 utilizes advanced data collection techniques and collects application component-level performance, availability, and operational metrics from any of the managed applications as well as user-defined components. *As a result, M6 is highly scalable; capable of processing millions of rules per second and monitoring small-, medium-, and large-scale infrastructures*. Capacity can be added on the fly by creating instances of CEP servers (event stream engines).

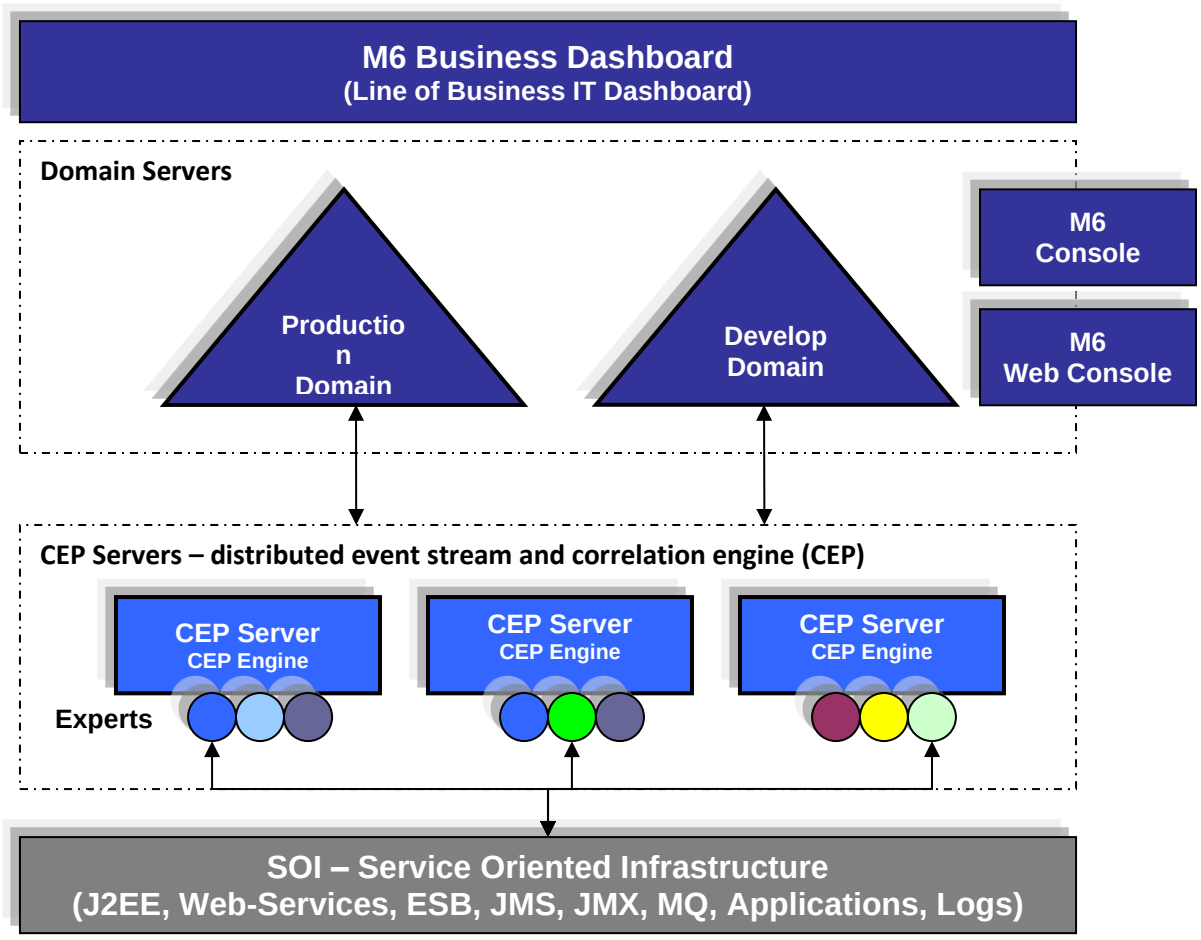


Figure 2-2. AutoPilot M6 Architecture

2.2.2 M6 Services

M6 services are monitoring agents, rules, policies automation scripts that monitor and act on a managed environment. (Example: an expert that monitors performance and health of a Windows platform.)

> **NOTE** Do not run M6 in a mixed JRE environment. Ensure all M6 services, Domain Server, CEP Servers and M6 User Console use a consistent JRE level. JRE 1.7 is recommended.

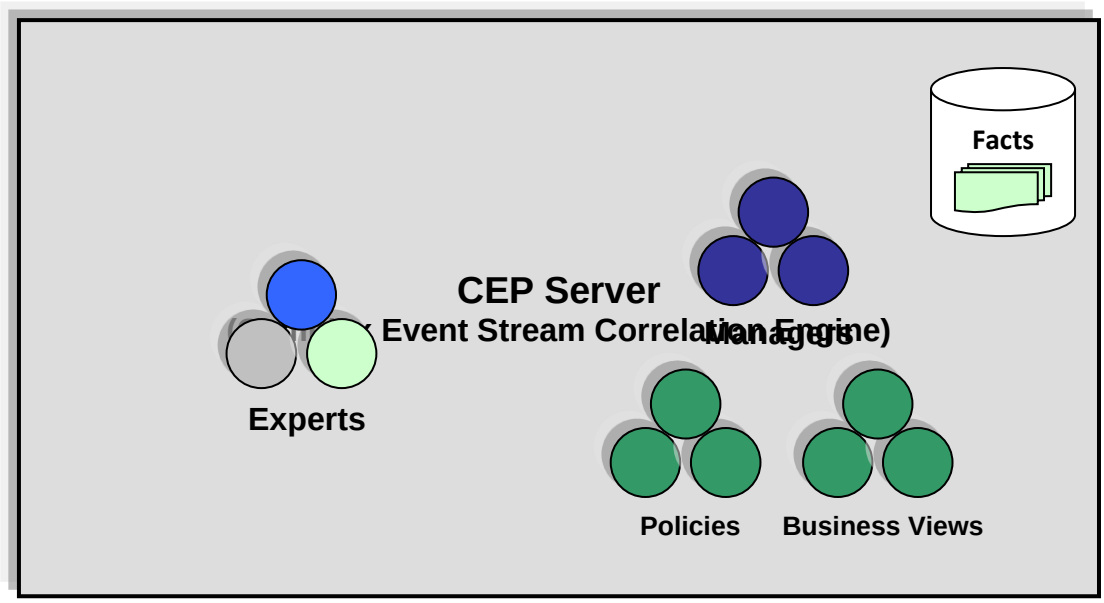


Figure 2-3. M6 Components

2.2.3 CEP Servers

CEP servers are a collection of containers that host management services. All CEP servers are registered within the domain server.

2.2.4 System Services

- **DOMAIN_SERVER:** Reserved M6 CEP server, which provides centralized directory and security services.
- **M6_WEB_SERVER:** Specialized CEP server that serves business views to web users using standard web browsers
- **SYSTEM:** There is a system folder under each CEP server.
    - Reserved and not to be modified.
    - System Services should never be deleted, moved, or modified.

2.2.4.1 Domain Server

- Domain Server is the main component of the M6 network.
- Domain server is a CEP server that consolidates security and directory services
- Domain Server is a primary CEP server and is capable of hosting management services
- There must be at least one domain server running on a network.

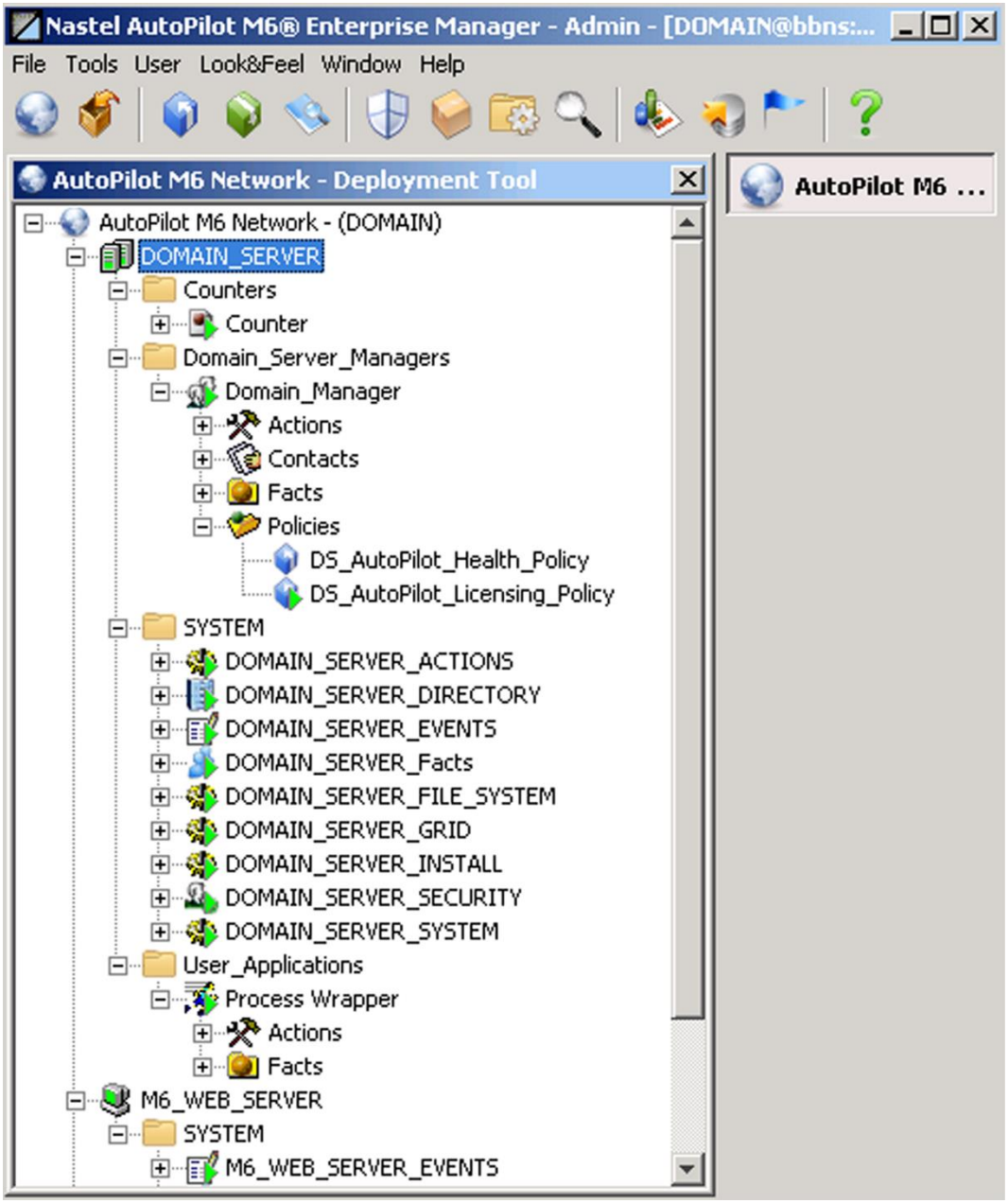


Figure 2-4. Domain Server View

2.2.4.2 M6 Web Server

M6 Web Server is a specialized CEP server that serves business views to web users, making management information available on the intra and/or extra-net. Requires JDK 1.5.x or higher

- M6 Web Server – CEP server acting as web server.
- Makes monitoring views available in the intra-net and /or extra-net.
- Usually installed on the same machine as the domain server.
- M6 Web Server requires no special set-up or software installation on remote machines, but Nastel recommends using Internet Explorer 5.5 or higher. Browsers and updates can be downloaded from the Microsoft web site http://www.microsoft.com/downloads
- Java Runtime Environment (JRE) 1.7 or higher is required to view/use M6 Web Console. JRE 1.7 is included with the M6 installations; however, if the remote user's machine does not have the current version of JRE, it will have to be updated or installed. Refer to the Sun Java web site for additional information: http://java.sun.com/j2se

M6 Web Console refers to a browser GUI interface that communicates with the M6 Web Server.

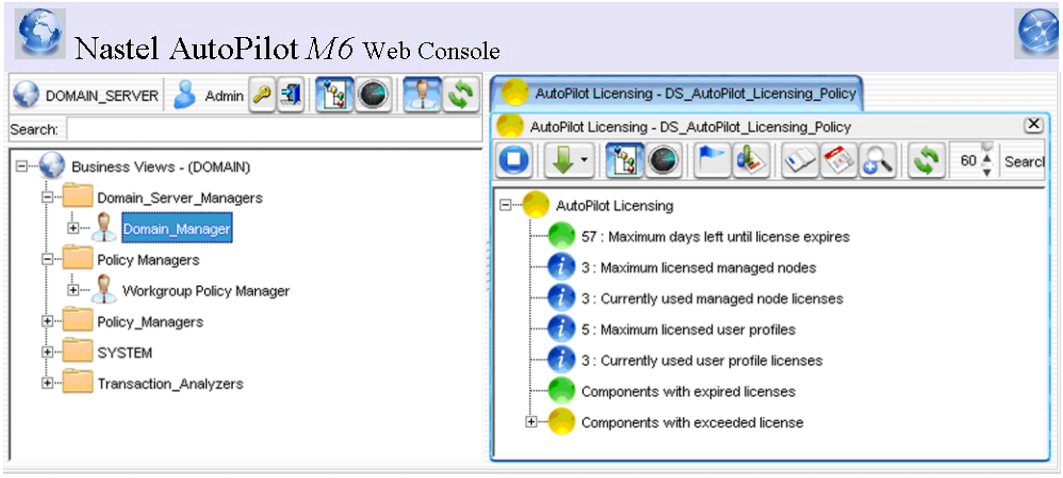


Figure 2-5. M6 Web Console View

M6 Web Console J2EE Deployment

M6 Web Console is a J2EE web application that can be deployed within any J2EE servlet engine. It is installed automatically with a Domain Server installation.

J2EE **autopilot.war** package is located in `[AUTOPILOT_HOME]/naming/j2ee-web`. Please refer to `[AUTOPILOT_HOME]/naming/j2ee-web/README.txt` for more details.

> **NOTE** AutoPilot M6 ships Jakarta Tomcat 4.1.36 servlet engine. It is installed automatically when Web Server is installed (part of the Domain Server installation option). M6 Web Console application is automatically configured and does not require separate deployment steps.

M6 Web Console can be accessed by pointing your browser to http://*webhost*:8080/m6console, where webhost is the host name of the M6 Web Server.

2.2.5 Monitoring Services

- **Experts and Agents:** components that collect data (performance, availability etc.) from underlying applications. The primary gateway between M6 and the target application.
- **Managers:** Managers are non-specific monitors that manage and dispatch policies and business views. Managers allow partitioning of policies and business views into groups based on purpose, security, and access control.
- **Policies:** Policies use proactive automation rules and procedures to perform actions on one or more management services on behalf of a user (for example, scheduling a policy).
- **Business Views:** Business views are a collection of rules that define a desired state of an *application* environment. Business views can be tailored to present the information in the form that best suits your needs. They represent a dependency model (tree form), where every element (sensor) describes how each application component should operate.

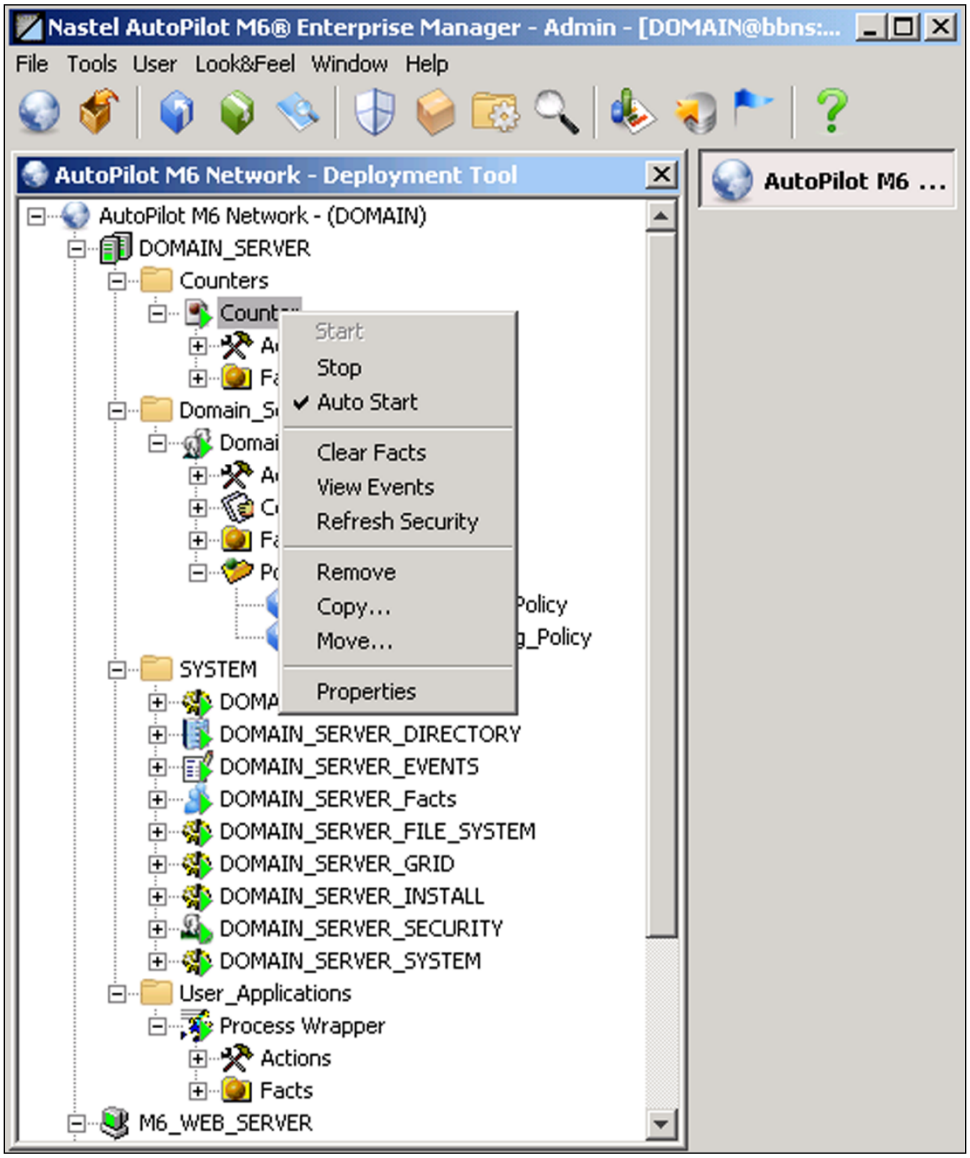


Figure 2-6. Management Services View

2.2.5.1 Experts and Agents

Experts and agents are interchangeable terms in M6. The official name for application specific module is Expert. Agent is a widely used term in system management that refers to a module that links a management platform to the managed resource.

- Experts monitor and control specific applications.
- Experts interface with managed applications using application specific interfaces (example: Experts for WebSphere Application Server, WebSphere MQ, Windows 200, 2003 and XP).
- Experts do not make judgments about the managed environment.
- Experts focus on data collection and execution of actions.

2.2.5.2 Managers

- Application independent monitoring and control agents
- Control and monitor multiple experts or managers
- Can be aggregated into *n*-tier management hierarchy
- Host and execute policies and business views

A manager has a set of policies and contacts. Policies are described in the following section. Contacts are other management services such as managers or experts that are associated with the given manager. Managers automatically subscribe to facts published by the services listed in the *Contacts* folder. These facts are routed to policies. The contact list is created dynamically and depends on the deployed policies and the services that these policies reference.

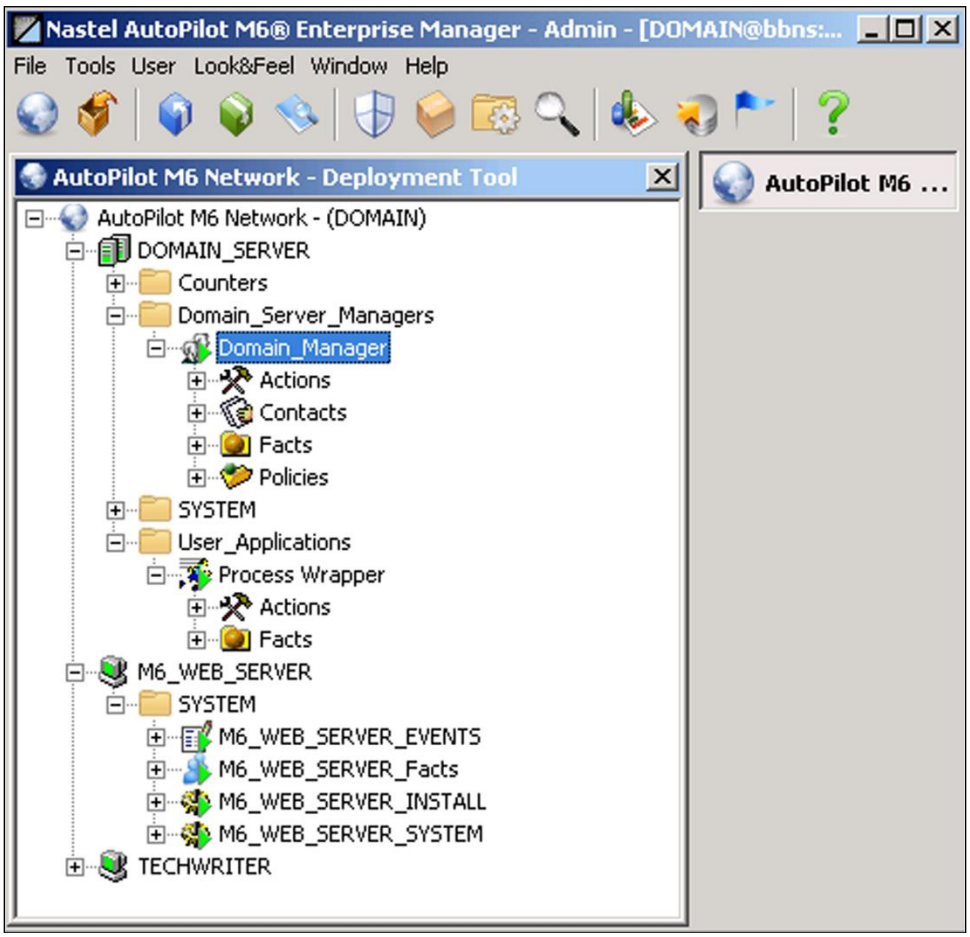


Figure 2-7. Manager View

2.2.5.3 Policies

- Perform actions on one or more management services
- Policies are capable of writing information to database, log files, and other data stores
- Policies subscribe to facts and act on fact changes, time events, or other conditions
- Usually specific to monitored applications
- Deployed within managers (example: Alert or execute an action based on a condition/event)

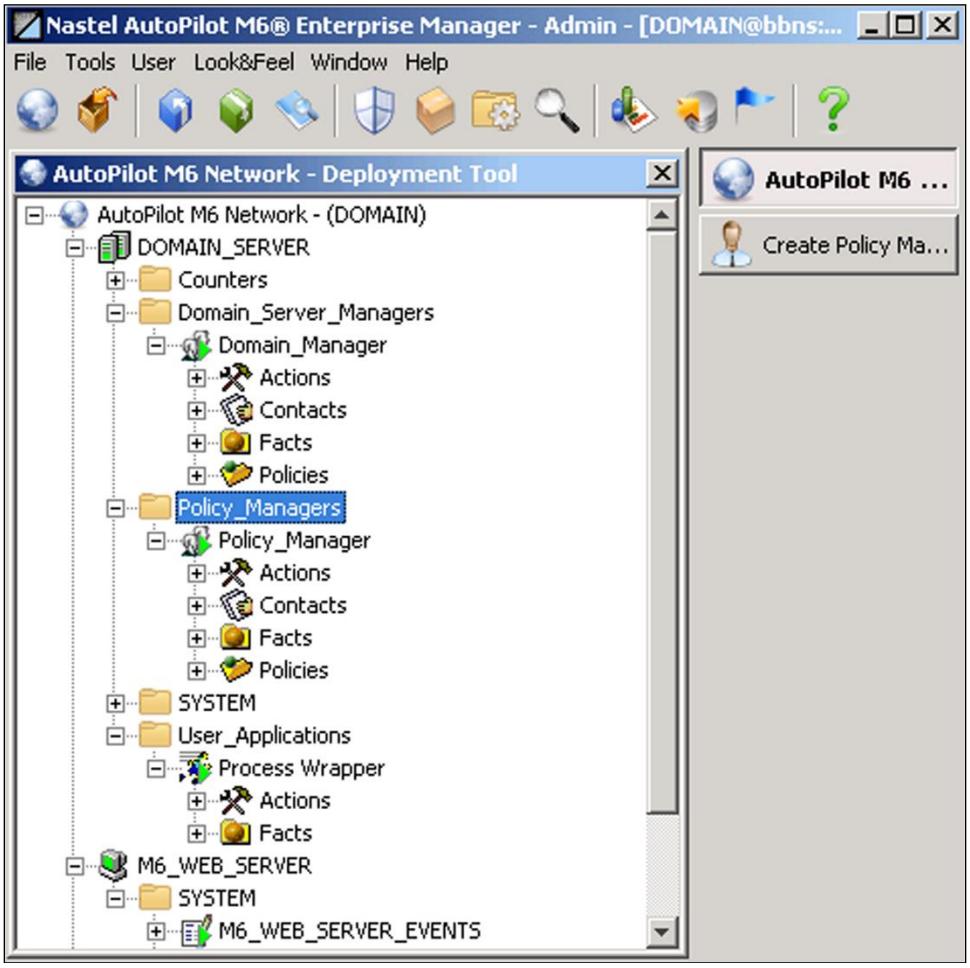


Figure 2-8. Policy View

2.2.5.4 Business Views

- Proactive user-defined policies that:
  - Correlate facts/events
  - Automate and alert
  - Generate user-defined events
  - Collect historical data for future analysis
- Business views are defined using M6 User Console
- Deployable as policies within any defined network of managers (running in the background)

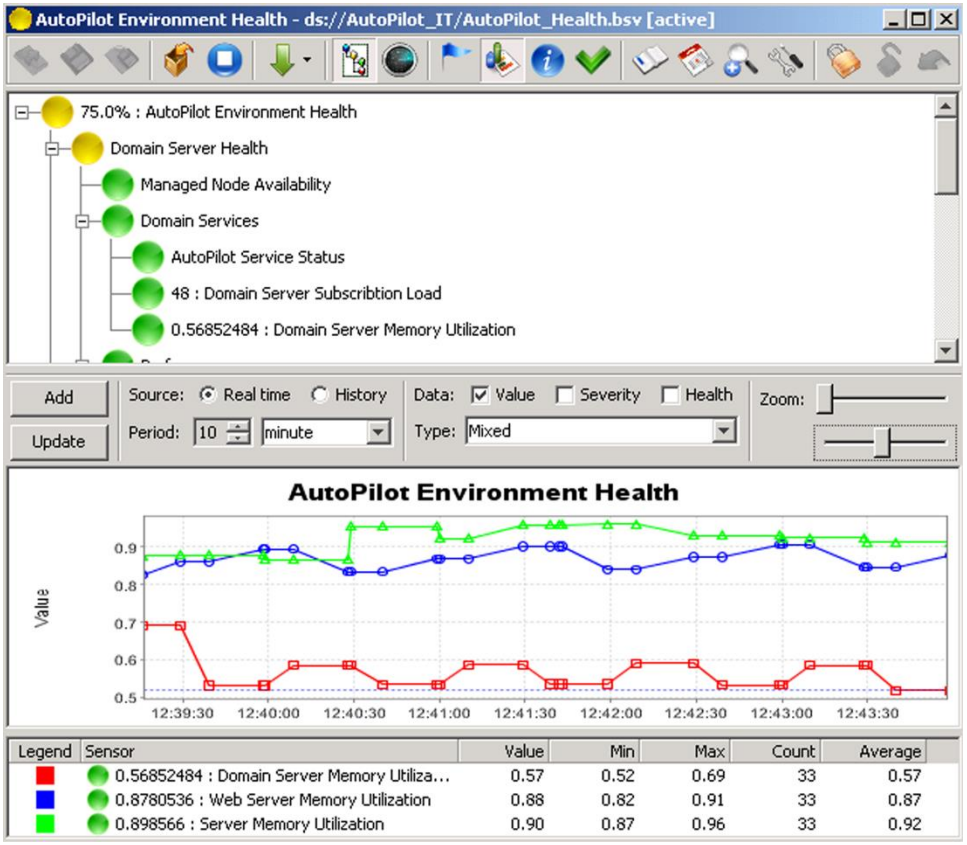


Figure 2-9. Business View Example

2.2.5.4.1 Version Control

Business views are guarded against overwriting by another user through a version control system. The version control system introduces a check-in and check-out model in which an individual user checks out a file, makes changes, and then checks the file back in. Other users are not able to make changes to a file while it is checked out.

If you have not been defined within VSS, the VSS module might hang as you perform a VSS operation because VSS SS.exe command prompts for a username/password and blocks the domain server. To mitigate this problem, ensure that you have provided valid VSS account information for every M6 user by configuring the *Misc* section in the User Manager. Refer to section 4-10 for information on configuring version control access.

If the domain server does hang, end all instances of SS.EXE executable. This procedure will need to be repeated for every instance of SS.EXE that is running.

2.2.5.5 AutoPilot M6 Facts – Metrics

- Facts are attributes or metrics published by management services about the target application, system, business process, or a system process. Examples:
  - Represents the state of an object or application
  - Performance data about applications, process, or resource
  - Metrics related to an application or business process
- Facts are variables that may change in real-time
- Facts follow this format: **Variable=Value/Method** where **Variable**, **Value**, and **Method** are objects (Example: *CPU\IDLE=90*)
- Fact names are broken down using "\" into groups of facts

- System metric example:
  - OS\Windows\Server1\CPU\IDLE=90
  - Applications\SAP_R3\Running=true
  - Web\WebSphere\Home_Page\Response=20
- Business metric example:
  - Inventory\Products\Machines=1020
  - Financial\DOW_Average=9065
  - Financial\NASDAQ_Average=1675

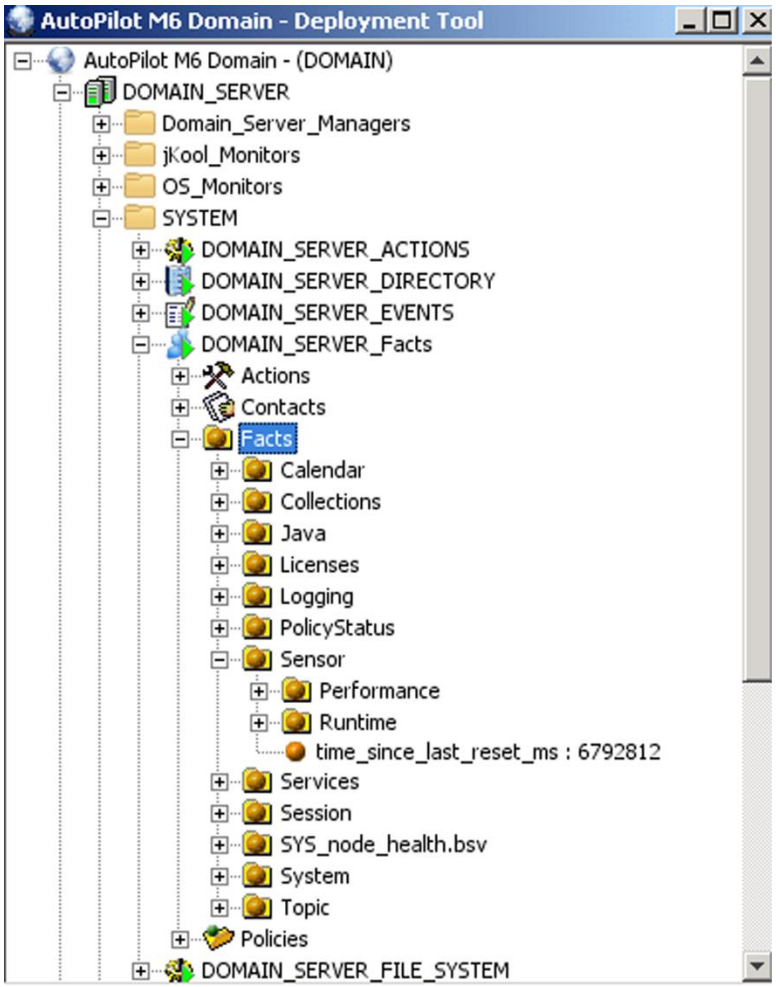


Figure 2-10. Facts View

2.2.6 N-Tier Service Oriented Architecture

The illustration below shows how Facts published by experts/agents flow into one or more subscribers. These subscribers are other management services such as managers, policies, and business views. Each management service will interpret the facts and produce:

- Other combined Facts
- Corrective or recovery actions
- Alerts and records information to database

Since CEP servers host management services, the architecture, in whole or in part, may be deployed into a single CEP server or can be split into multiple CEP servers.

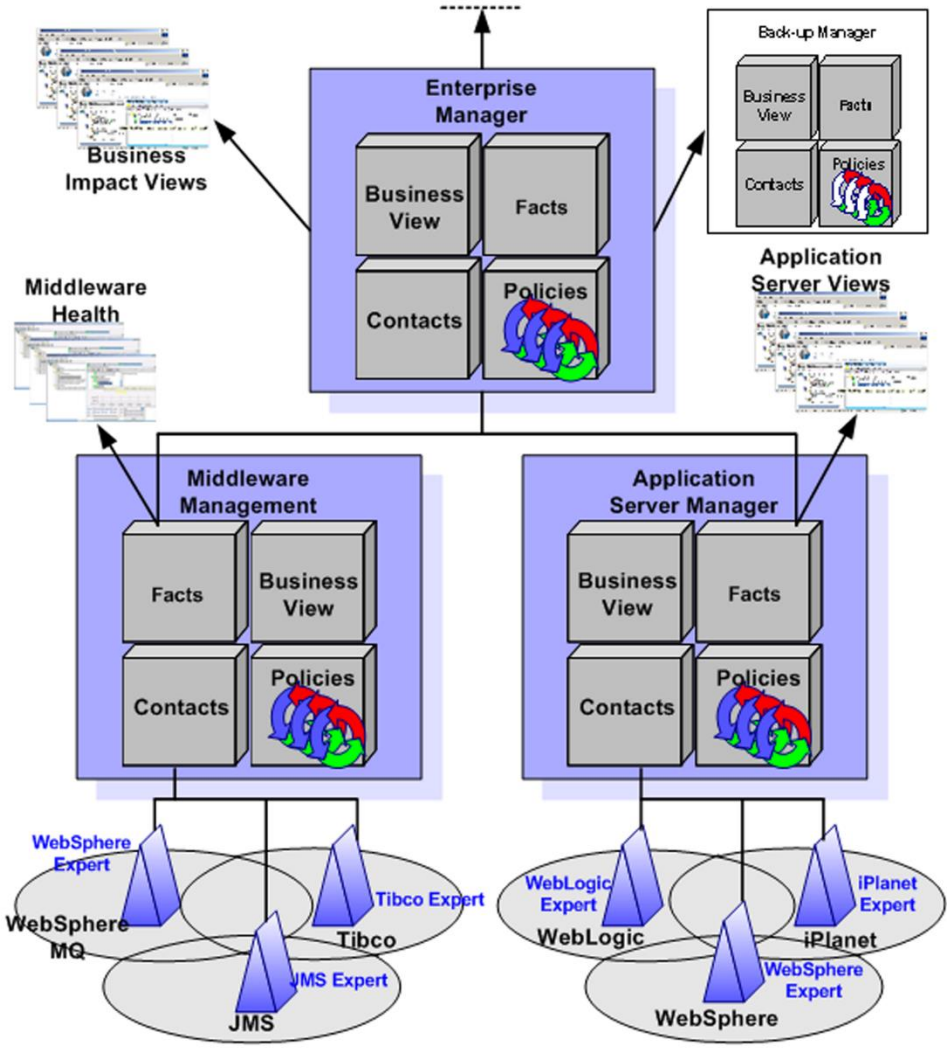


Figure 2-11. N-Tier Service Oriented Architecture

2.2.7 Typical AutoPilot M6 Installation

The following represents a typical AutoPilot M6 installation.

- Domain Server
  - Installed on a single machine
  - Should include M6 Web Server as well
- CEP Server
  - Installed on every machine where monitoring is required
- AutoPilot M6 User Console
  - Installed for every M6 administrator
  - Allows users to administer Nastel AutoPilot M6 domain
- **Installation Layout.** The illustration below reflects the follow conditions:
  - Domain Server is installed in a Windows 2000/XP environment. Only one domain server is required on the network.
  - CEP Servers are installed on other machines and are registered with the Domain Server.
  - M6 user consoles are installed on Windows/UNIX workstations
  - M6 Web Console is usually installed on the host running Domain Server. M6 Web Console users can use any machine with a web browser (requires browser Java plug-in)

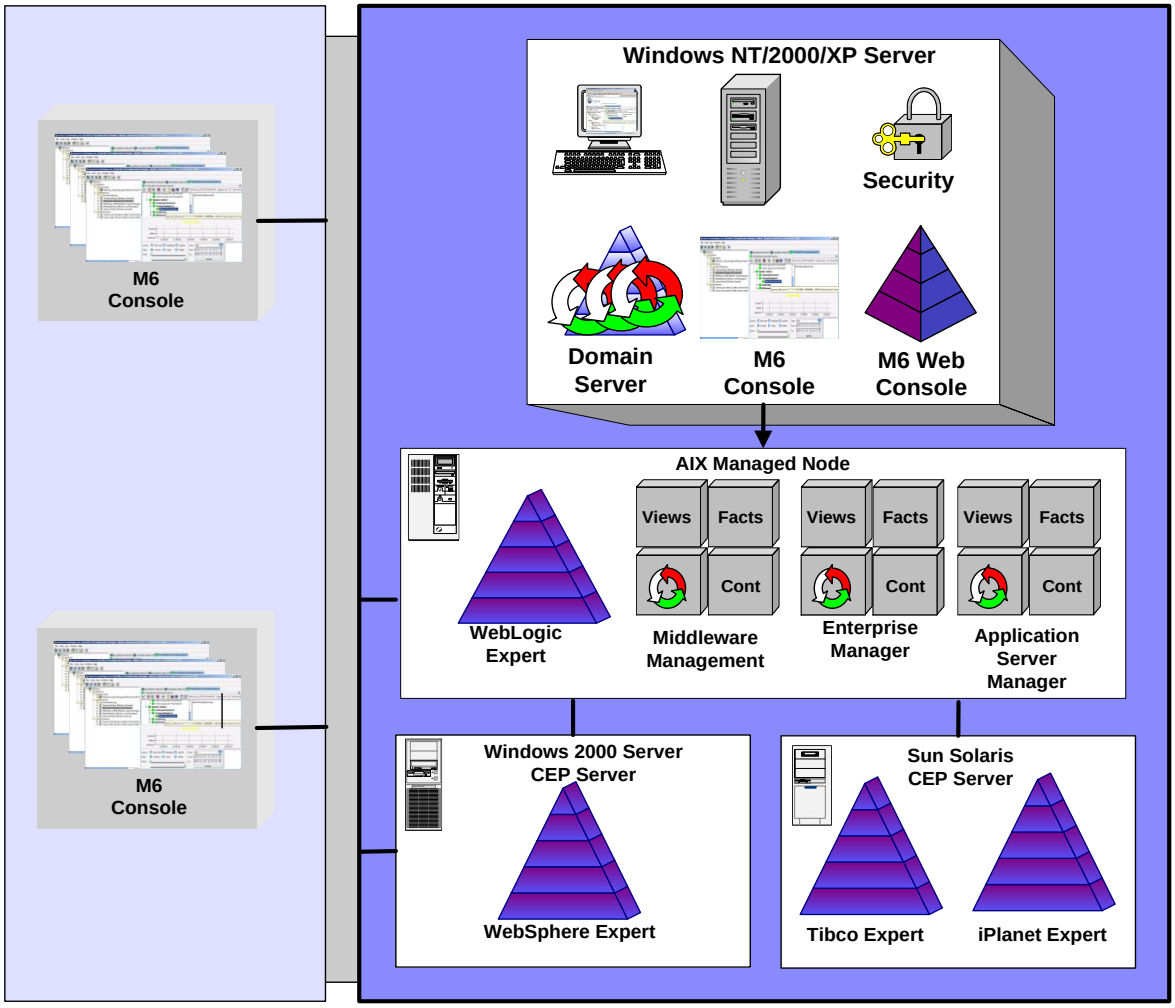


Figure 2-12. Typical Installation Layout

2.3 Native Security Model

AutoPilot M6 provides a role-based security model, where all users must be authenticated with the central trusted security authority – the Domain Server. The Domain Server assigns public and private keys for each created account including groups. Upon successful authentication, the domain server issues a digital signature that is signed with accounts public key. The issuing Domain Server verifies the digital signature, since it is the only authority that has the account's private key. The Domain Server must run on a trusted, secure host, which is guarded by sound security procedures.

All accounts are stored at domain server in the *security.dat* file under `[AUTOPILOT_HOME]\naming` directory. The file is stored in binary format and contains accounts, private-public keys, and encrypted passwords.

> **IMPORTANT!** ***Do not*** delete *security.dat* file, but **do** and back it up regularly. Loss or corruption of file will prevent users from logging on and may cause run-time problems. This file must also be protected against modification or any malicious activity.

2.3.1 Security Requirements

The security requirements for M6 vary depending on the extent of use. Adjustments have to be made depending on your specific use. For example, if you need to collect statistics to a database, then the relevant access is required to that database. Listed below are some common requirements:

- **Access to TCP/IP services on all systems**. The following are default listening ports (configurable) for each of the products:

Workgroup Server -	4010		
M6-WMQ Agent -	5000		
Domain Server -	2323	3000	8889
CEP Server -	2325	3005	
AutoPilot M6 Web -	8080	8007	

- **On MQ Servers**: The ID that runs the agent needs the following access to WebSphere MQ:

Connect	Get
Create	Put
Change	Alternate User
Inquire	Set
Display	

There may be others depending on the extent of use.

- **On MVS:** Read, Update, Alter and Control.
- **On WAS Servers**: The pmiexpert is used. If global security is turned *On*, the user id and password must be set in `sas.client.props`, `soap.client.props`. The passwords can then be encoded with [*wasroot*]`/bin/PropFilePasswordEncoder.bat` for enhanced security.
- **On HPOV SPI**: The SPI installation creates some Node Groups, Message Groups, Applications, User Profiles, and User. Authorized users must be given access to these Node and Message Groups.

2.3.2 Accounts and Passwords

User Groups

Group is a collection of user accounts that are logically related and share same rights. Groups may contain users or other groups. There are two default user groups, *Administrators* and *Operators*. Only the user with administrative privilege (member of *Administrators* group) can add, delete, and modify the users and groups. *Administrators* group is a system group and cannot be deleted.

User Accounts

Your local M6 administrator assigns user IDs and passwords. As a user, you cannot add or delete accounts, unless part of the *Administrators* group. M6 provides a *SYSTEM* account, which is reserved for system use and cannot be deleted.

Passwords

Your initial user password is issued by the system administrator, or as local policy dictates. The default password for accounts *Administrator* and *Admin* is *admin*, case sensitive. The passwords must be changed soon after the installation.

Password and password format policies should be adopted and enforced to prevent unauthorized system access. This is especially important if remote access (AutoPilot M6 Web Server) is to be authorized.

Please remember your password; passwords cannot be recovered due to one-way password encryption. User names and passwords are case sensitive.

Changing Passwords

Users must change their passwords after first logon in accordance with domain security policy. The default minimum character length is five which is set from domain server properties server.security.password.length=5. The changed password will be in effect the next time you logon. To change your password in the future, see your administrator.

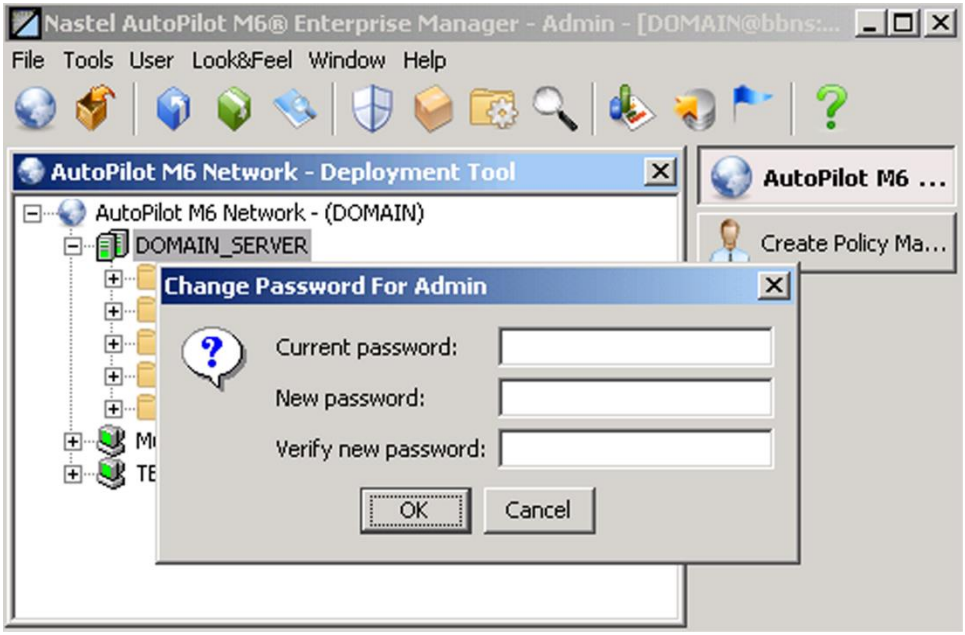


Figure 2-13. Changing User Password

2.3.3 Permissions and Ownership

Each management service has an owner – a user that deployed/created the service and a permission mask, which applies to all services that the user creates. Each user group has Administrator-defined access and limitations based on account policies. See your M6 Administrator for additional information.

The security tab for each expert, manager, policy, and business view identifies the owner, whether the service inherits the permissions from the originator/owner and permission settings. The permissions are inherited from owner's account "Security" settings and subdivided into three categories:

- **Group Permission** – the mask applies to all members of the account's group, which may contain other groups.
- **Others Permission** – the mask applies to a set of users that are not part of the group membership of a specific account.
- **Permissions (Supplementary)** – the mask applies to a group or individual user. It is used to grant additional permissions. Permissions granted here are added to the permissions granted by the **Group** or **Others** settings.

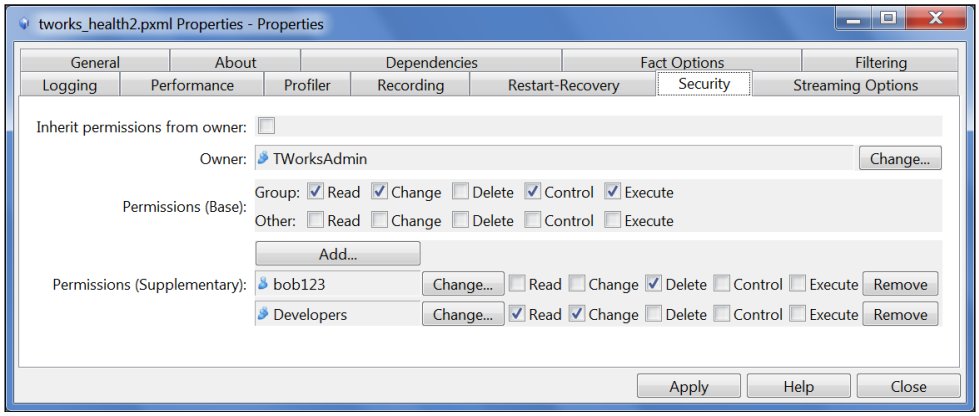


Figure 2-14. Security Tab

Changing the permission mask at the account level propagates the change to all owned services that inherit owner permissions. Propagation occurs only when the CEP server, hosting the services, is restarted.

2.3.4 Access Control

Members of the *Administrators* group have full access to all non-administrator owned objects within M6 even when owners deny access to all other accounts. Objects owned by members of the *Administrator* group follow standard access control procedures based on the object permission settings. By default, the following operations are reserved for members of the *Administrators* group only:

- **Domain Repository** – ability to save, change, upload business views
- **CEP Servers** – ability to stop and control CEP servers
- **Event Logs** – ability to delete/clear event log entries.
- **Security** – manage user account and security properties.
- **Deployment** – ability to deploy new experts and managers.

2.4 LDAP Integration

IMPORTANT!

LDAP is a separately licensed feature and must be unlocked by obtaining valid CPU-based license. The license must have "*KerberosLDAP*" feature enabled. (Use aplic command line to verify). Domain server will stop if "property server.domain.kerberos=true" and no valid license is present.

The LDAP authentication consists of a client-server model providing access to the entries in a directory structure via a directory service. The entries are referenced by their globally-unique Distinguished Name (DN). In LDAP, authentication is done by "binding" to the directory service. The data required depends on the type of authentication required, but generally requires the DN for a user and the user's password.

User/group membership can be derived from the appropriate server.

2.4.1 LDAP Requirements

To utilize strictly LDAP for authentication and group membership, the only requirements are:

- The ability for the Domain Server to connect to the LDAP service port. When configuring AutoPilot Domain Server LDAP configuration, you must specify the LDAP server URL as shown in these examples:
  - LDAP://myserver.example.com
  - LDAPS://myserver.example.com
  - LDAP://myserver.example.com:389
  - LDAPS://myserver.example.com:636
  - LDAP://myserver.example.com:3268
  - LDAPS://myserver.example.com:3269

  The default port for an LDAP connection is 389 and 636 for LDAPS. When you configure an LDAP connection to use port 389/636, you search for objects from this local domain controller only (replicated between domain controllers in the same domain). It has a complete set of all attributes each object contains. Alternatively, when configuring AutoPilot LDAP integration in environments with multiple domains in the forest, it is often required to use the Global Catalog in order to return objects from all domains in the forest.

The Global Catalog is a Read Only replica which contains a Partial Attribute Set (PAS) of objects within the forest, so it holds certain replicate objects from all domains. The default port for this is 3268 for LDAP and 3269 for LDAPS. When you configure the LDAP connection to use port 3268/3269, you search this Global Catalog (GC) to locate objects from any domain without having to know the domain name itself. This is often used in multi-domain forests where AutoPilot must pull users/groups from multiple domains.

To summarize:

- Default Ports: 389 (LDAP) / 636 (LDAPS): These ports are used for requesting information from the local domain controller. LDAP requests sent to port 389/636 can be used to search for objects only within the global catalog's home domain. However, the requesting application can obtain all of the attributes for those objects.
- Default Ports: 3268 (LDAP) / 3269 (LDAPS): These ports are used for queries specifically targeted for the Global Catalog. LDAP requests sent to port 3268/3269 can be used to search for objects in the entire forest. However, only the attributes marked for replication to the Global Catalog can be returned.

- Keystore file(s) containing certificate(s) for LDAP servers (if using SSL/TLS for LDAP server communication).

2.4.2 Configuring Domain Server for LDAP

Configuring Domain Server for LDAP requires:

- Defining necessary Domain Server LDAP properties
- Obtaining/generating a certificate for the LDAP server and adding it to keystore (for using SSL/TLS).

2.4.2.1 Configure Domain Server LDAP Properties

Configuring Domain Server for LDAP authentication consists of defining the appropriate properties in `[AUTOPILOT_HOME]/naming/node.properties`. The following properties are supported. The following template definitions are provided in `node.properties` (Table 2-1):

```
;Security LDAP Configuration Properties
;
; uncomment line below to enable LDAP authentication mode (default is false)
;property server.domain.ldap = true
;
; uncomment below to enable security trace. Enable for troubleshooting LDAP
;property com.nastel.nfc.security.trace = true
;
; defines a space-separated list of named LDAP servers. When authenticating users
; the servers are searched in the order specified.
;property server.domain.ldap.servers = ldapserver1
;
;LDAP Server Configuration
; Define the following properties for each named LDAP server above
; primary URL to use to connect to server
;property server.domain.ldap.<ldapserver1>.url = ldap://localhost:389
;
; failover URL to user for this server is unable to connect to primary URL - optional
;property server.domain.ldap.<ldapserver1>.failover_url = ldap://localhost:10389
;
; LDAP authentication model to use (simple|DIGEST5-MD|GSSAPI), default=simple
; - currently only simple is supported
```



```
;property server.domain.ldap.<ldapserver1>.auth = simple
;
; indicates whether or not SSL/TLS should be used to communicate with LDAP server
; only required when using SSL/TLS without ldaps
;property server.domain.ldap.<ldapserver1>.ssl = true
;
; if using SSL/TLS, defines keystore file containing SSL/TLS certificate for LDAP server
;property server.domain.ldap.<ldapserver1>.ssl.keystore = c:/temp/ApacheDS-trusted.ks
;
; defines keystore file containing SSL/TLS certificate for failover LDAP server
;property server.domain.ldap.<ldapserver1>.ssl.failover_keystore = c:/temp/ApacheDS-
trusted.ks
;
; LDAP user defined on server to use to query the server
;property server.domain.ldap.<ldapserver1>.searchuser = uid=admin,ou=system
;
; password for LDAP user used to query server
;property server.domain.ldap.<ldapserver1>.searchpwd = secret
;
;LDAP Server User Configuration
; base (root) entry in LDAP directory where users are defined
;property server.domain.ldap.<ldapserver1>.user.base = ou=users,dc=example,dc=com
;
; LDAP attribute whose value should be used as the user name in AutoPilot
;property server.domain.ldap.<ldapserver1>.user.name.attr = uid
;
; LDAP attribute whose value should be used as the description stored in AutoPilot security record
- optional
;property server.domain.ldap.<ldapserver1>.user.desc.attr = displayName
;
; LDAP class (objectclass) value that user entries must contain - optional
;property server.domain.ldap.<ldapserver1>.user.class = person
;
;LDAP Server Group Configuration
; base (root) entry in LDAP directory where groups are defined
;property server.domain.ldap.<ldapserver1>.group.base = ou=groups,dc=example,dc=com
;
; LDAP class (objectclass) value that group entries must contain - optional
;property server.domain.ldap.<ldapserver1>.group.class = groupOfUniqueNames
;
; LDAP attribute that defines group members
;property server.domain.ldap.<ldapserver1>.group.members.attr = memberOf
;
; LDAP attribute whose value should be used as the group name in AutoPilot
;property server.domain.ldap.<ldapserver1>.group.name.attr = cn
;
; LDAP attribute whose value should be used as the description stored in AutoPilot
; security record - optional
;property server.domain.ldap.<ldapserver1>.group.desc.attr = description
;
; defines time in seconds after which a user's group membership should be refreshed
; from LDAP server
; (0 implies never refresh - just use membership list received during login), default=0
```


;property server.domain.ldap.<ldapserver1>.group.refresh_time = 3600

Table 2-1. node.properties	
Property	**Description**
server.domain.ldap = [`true`|`false`]	Set to `true` to enable LDAP authentication support.
server.domain.ldap.servers = *<ldapserver1>* [*<ldapserver2>*] […]	Space-separated list of named LDAP servers to support.

The following properties (Table 2-2) are defined for each server named in server.domain.ldap.servers, replacing <ldapserver1> with the appropriate name from list.

Table 2-2. LDAP Server node.properties	
Property	**Description**
NOTE: Special characters are not allowed in LDAP attributes.	
server.domain.ldap.*<ldapserver1>*.url = *<URL for LDAP server>*	Primary URL used to connect to LDAP server, e.g., `ldap://localhost:389`
server.domain.ldap.*<ldapserver1>*.failover_url = *<URL for LDAP server>*	URL for failover server to use when this primary URL cannot be connected to. The failover LDAP server must have the exact same directory structure and entries as server at primary URL.
server.domain.ldap.*<ldapserver1>*.auth = [`simple`|`DIGEST5-MD`|`GSSAPI`]	LDAP authentication model to use. Default = `simple`
server.domain.ldap.*<ldapserver1>*.ssl = [`true`|`false`]	Indicates whether SSL/TLS is used to communicate with LDAP server. This is only required to use SSL/TLS when URL is not defined using "ldaps", since with ldaps, SSL/TLS is used independent of this setting.
server.domain.ldap.*<ldapserver1>*.ssl.keystore = *<path to keystore>*	When using SSL/TLS, defines path to the keystore file containing the security certificate for the LDAP server. Only required if certificate for LDAP server is not signed from a known Certificate Authority.
server.domain.ldap.<ldapserver1>.ssl.keystore.pwd = <password for keystore>	Password for accessing SSL/TLS keystore
server.domain.ldap.*<ldapserver1>*.ssl.failover_keystore = *<path to keystore>*	When using SSL/TLS, defines path to keystore file containing security certificate for the failover LDAP server. Only required if certificate for LDAP server is not signed from a known Certificate Authority.
server.domain.ldap.<ldapserver1>.ssl.failover_keystore.pwd = <password for keystore>	Password for accessing SSL/TLS keystore for failover server
server.domain.ldap.*<ldapserver1>*.searchuser = *<user DN>*	DN of a user defined on LDAP server for Domain Server to use to connect to LDAP in order to execute searches. This use requires read access to the subtrees containing user and group definitions, e.g., `ou=users,dc=example,dc=com`
server.domain.ldap.*<ldapserver1>*.searchpwd = *<user password>*	Password for searchuser defined above
server.domain.ldap.*<ldapserver1>*.searchscope = [`subtree`|`onelevel`|`object`]	Search scope to use when looking up users and groups. One of:

Table 2-2. LDAP Server node.properties	
Property	**Description**
	-subtree – search entire subtree rooted at user.base and group.base (default) **-onelevel** – search just the node at user.base and group.base and the level immediately below it **-object** – search just the node at user.base and group.base
server.domain.ldap.*<ldapserver1>*.user.base = *<root node DN>*	DN for base (root) entry in LDAP directory under which users are defined, e.g., `ou=users,dc=example,dc=com`
server.domain.ldap.*<ldapserver1>*.user.name.attr = *<user LDAP attribute>*	Attribute from LDAP user entries whose value should be used as the user's name in AutoPilot. This represents the name under which the user will log into AutoPilot. Default = `uid`
server.domain.ldap.*<ldapserver1>*.user.desc.attr = *<user LDAP attribute>*	Attribute from LDAP user entries whose value will be used as the description stored in AutoPilot security record for this user. This is optional and is only utilized during LDAP user import.
server.domain.ldap.*<ldapserver1>*.user.class = *<user LDAP class>*	Defines an LDAP objectclass that user entries must contain. This is optional and is only utilized during LDAP user import.
server.domain.ldap.<ldapserver1>.user.filter = <LDAP user filter>	Filter to apply when querying for users. This is optional, but when specified, must be a properly formatted LDAP filter string.
server.domain.ldap.*<ldapserver1>*.group.base = *<root node DN>*	DN for base (root) entry in LDAP directory under which groups are defined, e.g., `ou=groups,dc=example,dc=com`
server.domain.ldap.*<ldapserver1>*.group.name.attr = *< group LDAP attribute>*	Attribute from LDAP user entries whose value should be used as the group's name in AutoPilot.
server.domain.ldap.*<ldapserver1>*.group.desc.attr = *<group LDAP attribute>*	Attribute from LDAP group entries whose value will be used as the description stored in AutoPilot security record for this group. This is optional and is only utilized during LDAP group import.
server.domain.ldap.*<ldapserver1>*.group.class = *<group LDAP class>*	Defines and LDAP objectclass that group entries much contain. This is optional and is only utilized during LDAP group import.
server.domain.ldap.<ldapserver1>.group.filter = <LDAP group filter>	Filter to apply when querying for groups. This is optional, but when specified, must be a properly formatted LDAP filter string
server.domain.ldap.*<ldapserver1>*.group.members.attr = *<group LDAP attribute>*	Attribute from LDAP group entries whose values contain the user members of the group.
server.domain.ldap.*<ldapserver1>*.group.refresh_time = *<refresh interval>*	Defines the interval that group membership retrieved from LDAP server is considered stale and should be retrieved from LDAP server. A value of 0 indicates that membership list should never be refreshed. Note that this list is always retrieved when user logs in.

2.4.2.2 Defining Keystore

In order for the Domain Server to communicate with LDAP service using SSL/TLS, it requires the LDAP service to supply a trusted certificate. A trusted certificate is one that is either signed by a known Certificate Authority (e.g., Verisign), or one that exists in a store of trusted certificates. If you already have a properly signed certificate for your LDAP service, then no further configuration is required.

On the other hand, if you do not, then you will have to obtain a certificate and store it in a keystore accessible to the Domain Server. For the LDAP service to use SSL/TLS, it must have a public/private key pair defined and must be configured to use it. How this is done is different for each LDAP service. At this point, we're going to assume that this key pair already exists. If it does not, consult the documentation for your LDAP service.

The first thing that needs to be done is to export the LDAP service certificate from the LDAP server. This can be done with the java utility **`keytool`**, using options similar to the following:

```
keytool –export –keystore <keystore-file> -alias <keystore-alias>
-file ldap_serv.cer
```

You will be prompted for the keystore password. It will store the keystore in the file named `ldap_serv.cer`. The LDAP administrator will have to run this command for you.

Once you have the certificate file, you need to import it into a new or existing keystore. This should be done on the server running the Domain Server. Again, we use the keytool utility to import the certificate. The command below will import the certificate exported above into a keystore named ldap_serv.ks, creating it if it does not exist:

```
keytool –import –file ldap_serv.cer –alias <keystore-alias>
 -keystore ldap_serv.ks -storepass <keystore-password>
```

If the keystore exists, then *<keystore-password>* must match the expected one.

Java Keytool Commands: If you need to check the information within a certificate, or Java keystore, use the following commands:

- Check which certificates are in a Java keystore:
  `keytool -list -v -keystore keystore.jks`
- Check a particular keystore entry using an alias:
  `keytool -list -v -keystore keystore.jks -alias mydomain`
- Check a stand-alone certificate:
  `keytool -printcert -v -file mydomain.crt`

Now that we have our certificate in the keystore, we put the keystore file in a directory on the Domain Server system and set the property **server.domain.ldap.*<ldapserver1>*.ssl.keystore** in node.properties to the full path of this file.

2.4.3 Configuring Domain Server for LDAP (Windows Active Directory)


NOTE

The following applies to domain server installation only.

M6 domain server is capable of looking up existing users within LDAP directory service (such as Windows Active Directory). To configure domain server with LDAP, do the following:

- **Configure Domain Server properties** (LDAP)-
  `[AUTOPILOT_HOME]/naming/node.properties`
- **Configure JAAS**: Java Authentication and Authorization Service configuration (*jaas.conf*)

2.4.4 Accounts

User Groups

Group is a collection of user accounts that are logically related and share same rights. Groups may contain users or other groups. There are two default user groups, *Administrators* and *Operators*. Only the user with administrative privilege (member of *Administrators* group) can add, delete, and modify the users and groups. *Administrators* group is a system group and cannot be deleted.

Groups are defined as one of 2 types:

- **Native** – these groups are defined in AutoPilot with no assumed relationship to groups defined elsewhere (like operation system groups or LDAP groups)
- **LDAP** – these groups are assumed to match groups defined in an LDAP server. To define groups of this type, they must be imported from an LDAP server. The main use for these types of groups is to derive the group membership from the LDAP server instead of defining the memberships in AutoPilot.

User Accounts

Your local M6 administrator assigns user Ids. As a user, you cannot add or delete accounts, unless part of the *Administrators* group. AutoPilot M6 provides a *SYSTEM* account, which is reserved for system use and cannot be deleted.

Users are defined as one of 3 types:

- **Native** – for these users, all authentication information is stored within AutoPilot, and when these users log in, the specified credentials are compared against these stored values.
- **LDAP** – for these users, the authentication information is stored in an LDAP server, and when these users log in, they are authenticated against the LDAP server.

User/Group Membership

Users and Groups can be associated either explicitly, where each user is specifically assigned to one or more user groups, or derived, where the user's group membership is retrieved from an external (e.g., LDAP) server. In order to derive a user's group membership, the user must be defined as a non-Native user and the user's security record must be configured to retrieve the group membership from the external server. The explicit user/group memberships can be defined between all types of users and groups, with the following exceptions:

- For **LDAP groups**, only Native users can be explicitly added to the group. Which LDAP users are members of this group are obtained from the LDAP server.
- For **LDAP users**, they can only be explicitly added to Native groups, since again, which LDAP groups they belong to is obtained from the LDAP server.

2.4.5 Permissions and Ownership

Each management service has an owner – a user that deployed/created the service and a permission mask, which applies to all services that the user creates. Each user group has Administrator-defined access and limitations based on account policies. See your M6 Administrator for additional information. The security tab (Figure 2-14) for each expert, manager, policy, and business view identifies the owner, whether the service inherits the permissions from the originator/owner and permission settings. The permissions are inherited from owner's account "Security" settings and subdivided into two categories:

- **Group Permission** – the mask applies to all members of the account's group, which may contain other groups.
- **Others Permission** – the mask applies to a set of users that are not part of the group membership of a specific account.

- **Permissions (Supplementary)** – the mask applies to a group or individual user. It is used to grant additional permissions. Permissions granted here are added to the permissions granted by the **Group** or **Others** settings.

Changing the permission mask at the account level propagates the change to all owned services that inherit owner permissions. Propagation occurs only when the CEP server, hosting the services, is restarted.

2.5 User Interface

AutoPilot M6 provides two types of user interfaces. M6 User Console is used for administration and M6 Web Console for operator access. Each of these interfaces allows users to access business views that:

- Monitor collect data based on defined requirements.
- Access management views across intra- or extranets via M6 Web Server.
- Graphically see the status of applications and impact of failures on overall environment.
- Receive alerts when failure and/or recovery occur.
- View system performance and status statistics.

Only M6 User Console provides users with full capability to administer M6 domain and to develop and test business views.

2.5.1 M6 User Console

IMPORTANT!

Do **not** run M6 in a mixed JRE environment. Ensure all M6 services including M6 User Console use a consistent JRE level. JRE 1.7 or higher is required.

The M6 users have full access to all operational functions within M6 User Console, including tools to load and create business views. The user will be able to review properties of all services, monitor facts and events. Individual access to M6 and ownerships are as specified by your local M6 administrator.

2.5.2 M6 Web Console

M6 Web Console is a browser-based interface that provides monitoring and operational control over managed resources and applications. It allows users to:

- Monitor health and status of all deployed business views (subject to security)
- Perform operator-initiated actions to recover from a problem or failure
- View events and historical performance graphs
- Visualize impact of a failure on various parts of infrastructure.

2.6 Nastel AutoPilot M6 Business Dashboard

The M6 Dashboard is a web-based graphical user interface that is deployed as a component of the web portal. It was developed for Line of Business (LOB) users as a front-end for all plug-ins and monitors and checks for problems in all business services. It also checks Key Performance Indicators (KPIs) as defined by the sensors in M6. M6 Dashboard receives all of its information directly from M6 User Console and M6 Web. It imports data from M6 (historical) and from M6 Web (real time).

M6 database schema has been adjusted to support M6 Dashboard. The Historical Sensor Table [*db_sensor_name*] is shown in Appendix F.

All real-time data will be stored in a table maintained by M6 bsv and bsp policies. If the policy is configured to record to a table, for example, db_sensor; its real-time data will be recorded into *db_sensor_rt* (_rt suffix). The Real-Time Sensor Table [*db_sensor_name*] is shown in Appendix F.

Real-time state of every sensor is recorded to the database for logging. Only the most current status needs to be maintained for each marked sensor. Tables are created for real-time db logging which have the same schema as the historical sensor table. Sensors log all the sensor details including the creation of db_services table that contains a list of all manager/policy/domain pairs and their associated Service IDs (SIDs). The Policy Table is shown in Appendix F.

When the sensor state changes, a new record is inserted if one does not already exist; otherwise, the existing record is updated. If the sensor gets deleted (temporary sensor), the record is deleted from the database. When the BSV is stopped or started, all records are deleted pertaining to manager/policy name in the real-time table. This ensures that the table contains only newest records for the BSV. Multiple BSVs/BSPs record real-time data to the same table, *db_sensor_rt* table. The current status of a sensor can be looked up by its SID and sensor name. When a business view is stopped, all real-time status records of its sensors are deleted.

Chapter 3: Getting Started

3.1 AutoPilot M6 Basics

AutoPilot M6 User Console is a Graphical User Interface (GUI) application that gives users access to functionality and information provided by M6. It is thin client (clients are any component that uses the M6 infrastructure: such as M6 User Console). As a rule, all functions, and properties are visible to all users, however, based on permissions and security, changing, editing properties, and creating new managers, business views, expert is subject to M6 domain security policy.

NOTE

If you are running M6 on a Linux platform, refer to Appendix D for Linux specific configurations.

3.1.1 Launching AutoPilot M6 in Windows Environment

NOTE

Domain Server must be running and highly available.

1. Click **Start > Programs > Nastel AutoPilot M6** to open M6 Enterprise Manager program menu.

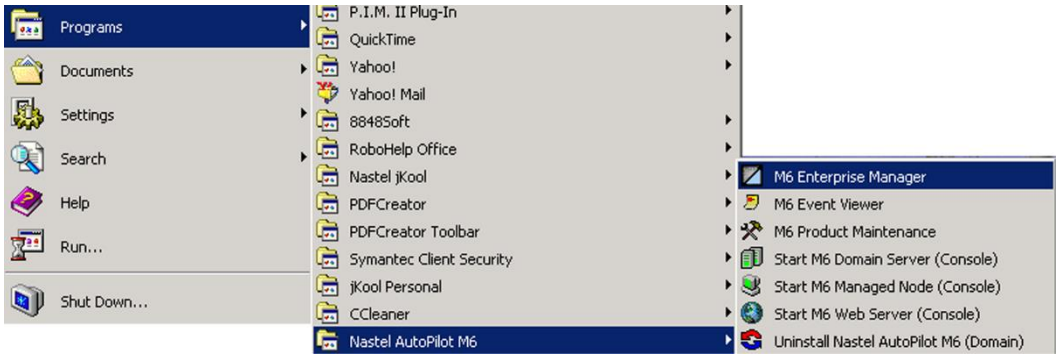


Figure 3-1. Windows Program Group

2. Click **M6 Enterprise Manager.** The *About AutoPilot M6 Enterprise Manager* screen is displayed.

Figure 3-2. AutoPilot M6 User Console -- Splash Screen

You can log onto M6 using either Native Authentication or by selecting the appropriate Security Realm.

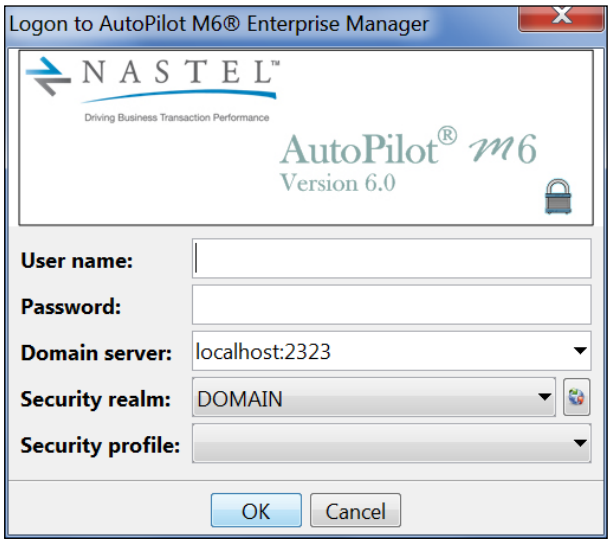


Figure 3-3. Logon to AutoPilot M6 User Console

3. When the *Logon* screen is displayed:

   a. Click the refresh  icon to obtain the Native domains.

   b. Select the *Security realm* you will be using to log in by clicking the down arrow in Security Realm field to display the pull-down menu.

4. If configured by your AutoPilot administrator, the *Security profile* drop-down list provides the name of a specific security profile used for logon. If not configured, this field should be left blank. If an invalid field is provided, the logon may be unsuccessful.

5. Enter *User name* and *Password*.

6 Enter *Domain server* hostname and port.

The *Domain server* field includes the domain server name and port in the format <host:port>. The default *Domain Server* is *localhost*; if you want to access other domains within the AutoPilot M6 network, enter or select the Domain Server from the pull-down menu.

The *Domain Port* is set during installation either by the installer or by default. If you want to change it to another, enter a new port or select the domain port from the pull-down menu.

7 Click **OK**.

3.1.2 Launching AutoPilot M6 from the Command Prompt

All M6 components can be started from a command prompt or UNIX shell. All commands and command line options are platform independent and work on all platforms. The exact path and calling procedures are platform specific.

> **CAUTION!** TCP ports *must not* be shared among M6 services and any other TCP server/application on the same machine. Please refer to component specific node.properties for port configuration.

3.1.2.1 Starting Domain Server

1. From Command Prompt, **cd** [*AUTOPILOT_HOME*]
2. From [*AUTOPILOT_HOME*] directory type: **cd naming**.
3. From [*AUTOPILOT_HOME*]\naming type: **ATPNAMES**, to start the domain server.

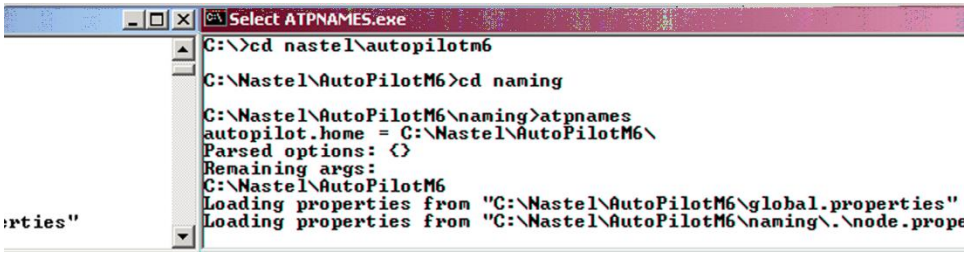


Figure 3-4. Starting a Domain Server

4. You may start the domain server in console mode, where all output is redirected to the user's console, type: **ATPNAMES -console**. All relevant data about the server and its host will be displayed. The "**AutoPilot M6 [Domain] is ready!**" statement will be posted along with information and status display options.

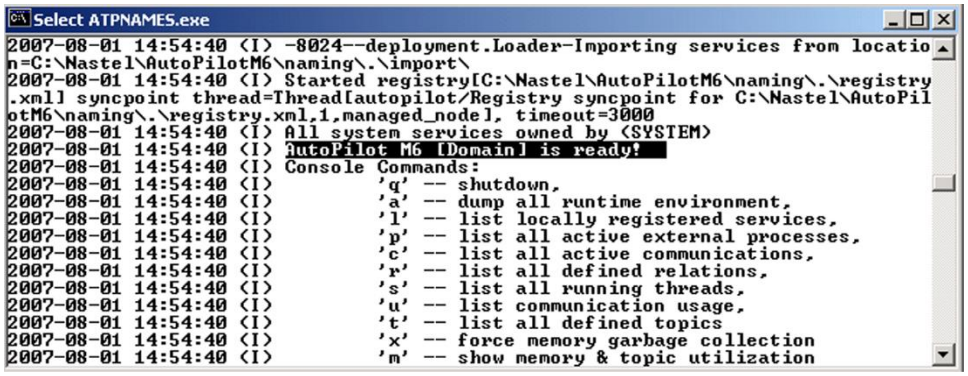


Figure 3-5. ATPNAMES Console Mode

2.4.1.1 Starting CEP Servers

NOTE

1. Domain Server (ATPNAMES) must be running for the AutoPilot M6 environment to work.
2. The following procedure is typical of starting M6 in a Windows environment; however, it is typical of start procedures on all supported platforms.

1. From Command Prompt, type: **cd** [*AUTOPILOT_HOME*]
2. From [*AUTOPILOT_HOME*] directory type: **cd localhost**
3. From [*AUTOPILOT_HOME*]\localhost type: **ATPNODE -console** . Wait for node to load.

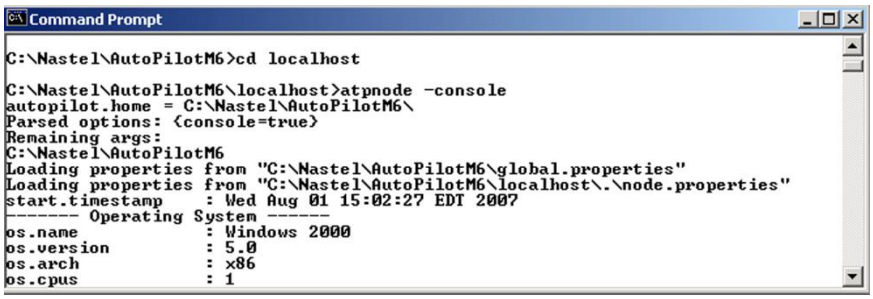


Figure 3-6. Starting CEP Server

3.1.2.3 Starting AutoPilot M6 Web Server

NOTE

Domain Server (ATPNAMES) must be running and highly available. The domain server must be running for AutoPilot M6/Web to start properly.

1. From Command Prompt, type: **cd** [*AUTOPILOT_HOME*]
2. From [*AUTOPILOT_HOME*] directory type: **cd jakarta-tomcat\bin**
3. From [*AUTOPILOT_HOME*]\jakarta-tomcat\bin, type: **catalina run**.
   (On UNIX systems: **sh ./catalina.sh run**)

3.1.2.4 Starting AutoPilot M6 User Console

1. From [*AUTOPILOT_HOME*]\mconsole type: **ATPCONS -console**. *AutoPilot M6 Enterprise Manager splash* screen is displayed. Click **OK**.

Figure 3-7. AutoPilot M6 User Console and Splash Screen

NOTE

1. The following procedure is typical of starting M6 in a Windows environment; however, it is typical of start procedures on all supported platforms.
2. Your local M6 Administrator assigns Usernames and Passwords. See the System Administrator for Username and Password assignments.

2. When the *Logon* screen is displayed:

   a. Click the refresh icon to obtain the Native domains.

   b. Select the *Security realm* you will be using to log in by clicking the down arrow in Security Realm field to display the pull-down menu.

3. If configured by your AutoPilot administrator, the *Security profile* drop-down list provides the name of a specific security profile used for logon. If not configured, this field should be left blank. If an invalid field is provided, the logon may be unsuccessful.

4. Enter *User name* and *Password*.

5. Enter *Domain server* hostname and port.

   The *Domain server* field includes the domain server name and port in the format `<host:port>`. The default *Domain Server* is *localhost*; if you want to access other domains within the AutoPilot M6 network, enter or select the Domain Server from the pull-down menu.

   The *Domain Port* is set during installation either by the installer or by default. If you want to change it to another, enter a new port or select the domain port from the pull-down menu.

6. Click **OK**.

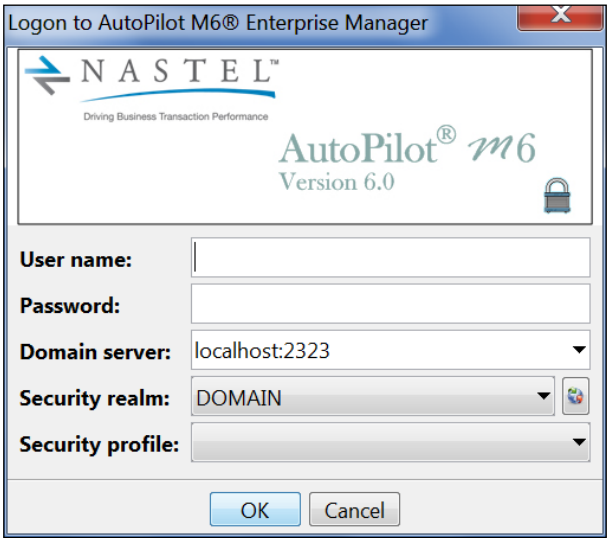


Figure 3-8. Logon to M6 User Console

7. When *Management Console* is displayed, M6 User Console is running.

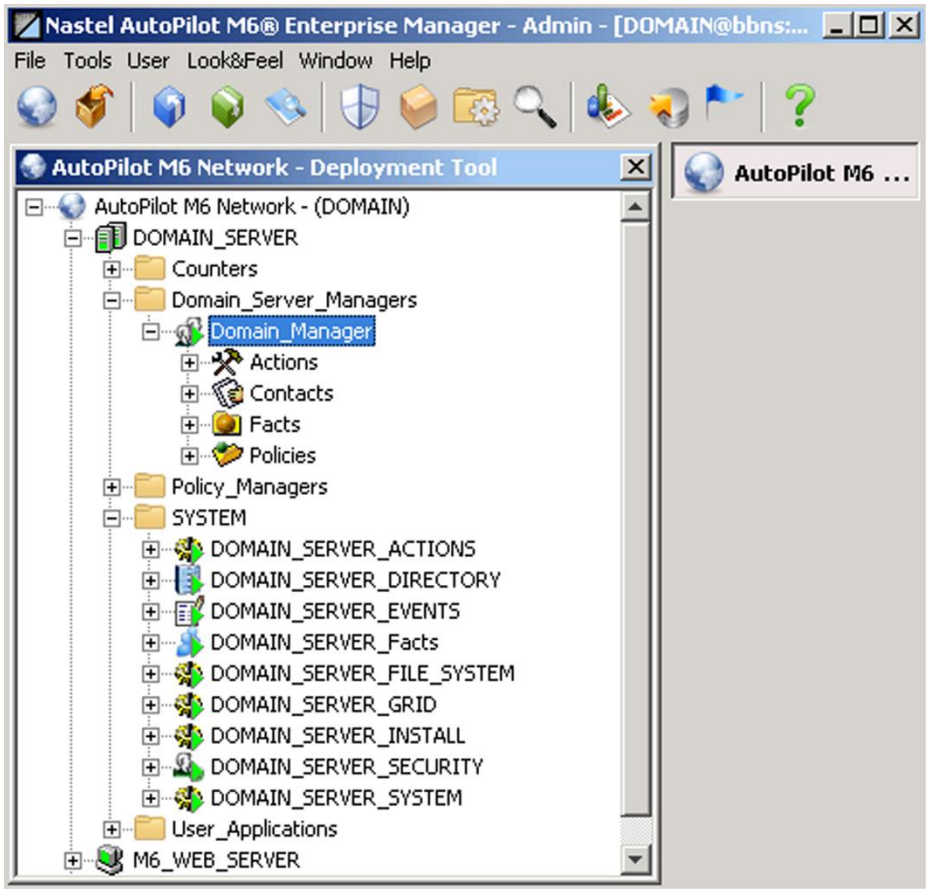


Figure 3-9. User Console

2.4.2 Stopping AutoPilot M6 Services

NOTE There are no specific logoff procedures required in M6. The following is a typical example.

1. From the *Deployment Tool* screen, right click **DOMAIN_SERVER** or the CEP server you want to stop, sub-menu will be displayed. Click **Stop Node**. The icon of the node you stopped will switch to red and the status will be displayed as *Off-Line* (e.g., icon switches to grey) when screen is refreshed.

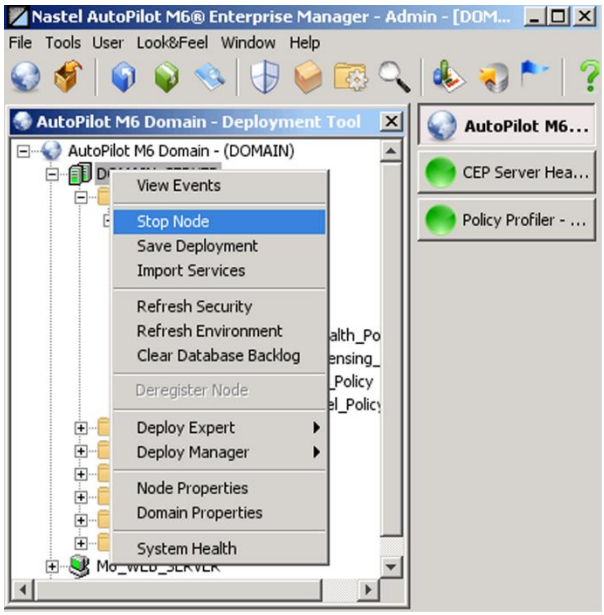


Figure 3-10. Stopping Domain Server or CEP Server (Typical)

2. Exit Consoles by clicking **File** to open pull-down menu, click **Exit.** Screen will close; you will be logged off.

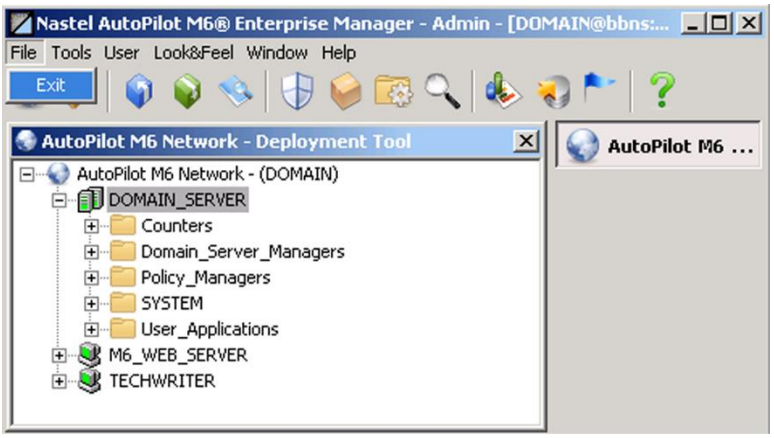


Figure 3-11. Exiting M6 User Console, User Logoff

NOTE After exiting from the M6 User Console, the domain server and CEP servers remain running in the background. If it is necessary to stop these components, they must be shutdown manually. This can be accomplished from the command prompt or in Services (Windows).

3. Shut down servers in Command Prompt by typing: "q" [Enter]. Repeat this for each running service. This procedure can be used only if the service is started with "-console" mode. Use **apnet** utility to shutdown M6 services regardless of the location or start procedure.

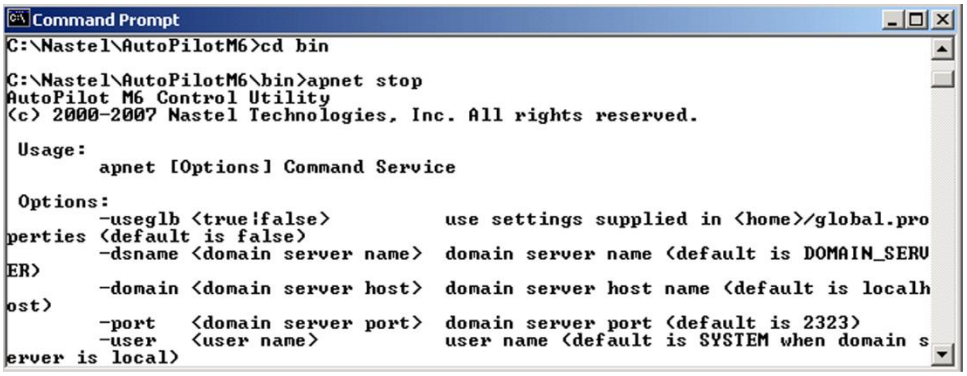


Figure 3-12. Command Prompt: Domain Server

3.1.4 Starting and Stopping AutoPilot M6 on UNIX

3.1.4.1 Starting AutoPilot M6 Servers

1. To start M6 in an UNIX environment use the following commands:
   - **Domain Server:** *[AUTOPILOT_HOME]*/naming/ATPNAMES
   - **CEP Server:** *[AUTOPILOT_HOME]*/localhost/ATPNODE
   - **M6 Web Server:** *[AUTOPILOT_HOME]*/jakarta-tomcat/catalina.sh run
2. To start the console, start an X-Server on your workstation (if using a PC) set the DISPLAY environment variable to your PC

   Example: **ksh export DISPLAY=your_host:0**

   Example: **csh setenv DISPLAY your_host:0**
3. Start: [*AUTOPILOT_HOME*]/mconsole/ATPCONS -console

   where:

   [*AUTOPILOT_HOME*] is the M6 installation directory.

3.1.4.2 Stopping AutoPilot M6 Servers

Stop the CEP servers, M6 Web Server, and domain server from within M6 by right clicking on the node or server, and then clicking on **Stop Node**. Stop the nodes first, then the domain server.

Alternately, you can use *apnet* or UNIX *kill* command to stop processes ATPNAMES, ATPNODE. It is highly recommended *not* to use the UNIX "kill -9" command.

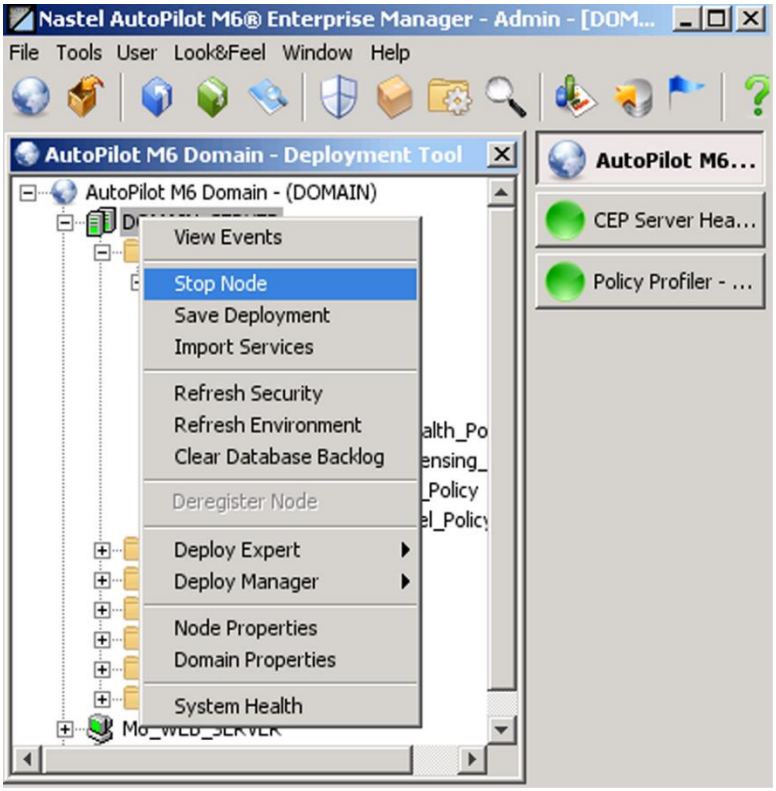


Figure 3-13. Stopping M6 on any Platform

3.1.5 Start and Stop Services Using apnet

APNET can be used to start or stop M6 services and stop nodes. Please refer to C.2 APNET – Control Utility for more information.

3.1.5.1 Stopping Services and CEP Servers

IMPORTANT!

CEP Servers stopped using *apnet* or the *Console* can be restarted from the command line or Windows Services. They cannot be re-started using apnet or console.

1. CD to: *[AUTOPILOT_HOME]*/bin.
2. Type: **apnet**. The *apnet options* will be displayed. The commands displayed can be used to start and stop any M6 service or node.
3. To stop the M6 CEP Server type:
   apnet -domain localhost stop <NODENAME>_SYSTEM
4. To stop the M6 domain server type:
   apnet -domain localhost stop DOMAIN_SERVER_SYSTEM

To stop the AutoPilot M6 Web Server type:

For Windows:

1. CD to: [*AUTOPILOT_HOME*]
2. From [*AUTOPILOT_HOME*] directory type: **cd jakarta-tomcat**
3. From [*AUTOPILOT_HOME*]\jakarta-tomcat directory type: **cd bin**
4. From [*AUTOPILOT_HOME*]\jakarta-tomcat\bin directory type: **\shutdown.bat**

For UNIX:

1. CD to: [*AUTOPILOT_HOME*]
2. From [*AUTOPILOT_HOME*] directory type: **cd jakarta-tomcat**
3. From [*AUTOPILOT_HOME*]\jakarta-tomcat directory type: **cd bin**
4. From [*AUTOPILOT_HOME*]\jakarta-tomcat\bin directory type: **\shutdown.sh**

You can also use Windows Services to shut down all services.

3.1.5.2 Starting Services

APNET command line utility can start M6 services; however, it cannot start domain server, CEP Server or AutoPilot M6 Web Server. These services must be started using platform specific start up procedures. APNET can, however, start management services such as experts, managers, and policies within CEP Servers.

IMPORTANT!

CEP Servers stopped using *APNET or the Console* can be restarted from the command line or Windows Services. They cannot be re-started using *APNET*.

To start a management service:

```
apnet -domain localhost start Service
```

3.1.6 Getting Started with AutoPilot M6 User Console

M6 User Console provides your user interface for monitoring of M6 domain, to create business views and monitor applications. Local security policy will dictate the specific level of access and abilities each user groups actually has been granted. All the details should be available from your Administrator. Domain security policy is managed using *User Manager* tool within AutoPilot M6 User Console.

3.1.6.1 AutoPilot M6 User Console Navigation

Application Management Console:

- **Title bar:** Displays domain name and domain URL location.
- **Pull-Down Menus:** Menus provide navigation and access to standard and high-level functions within AutoPilot M6.
- **Main Toolbar:** Toolbar provides quick access to most frequently used components of AutoPilot M6.
- **Tab Selection:** Click on tab (right-hand side of screen) to open that window. As a new window is opened, its tab is added to the right-hand side of the screen.

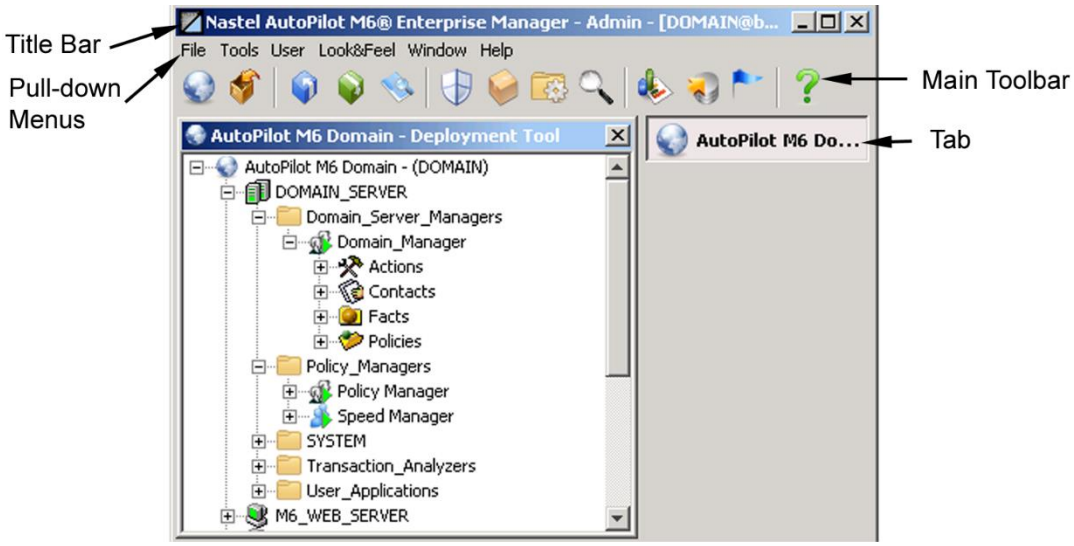


Figure 3-14. Menu and Tool Bars

Pull-Down Menus

This section provides instructions and definition for functions, links, and commands related to components of Pull-Down Menu. Pull-down menu consists of the following:

1. **File:** Click **File,** the *File* menu is displayed.
   - **Exit:** Click **Exit** to close AutoPilot M6 User Console and log off the AutoPilot M6 network.

> **NOTE** The following menu items are only accessible if you are displaying your open views (tabs) across the top of your screen. (Select Tabs from the Windows menu.) Exit is selectable from both.

   - **New:** Click **New** to display untitled window at left of screen.
   - **Open:** Click **Open** to display *Open Favorite View* menu. Select the desired file. Click Open to initiate the selected favorite view. Favorites are saved with a `.cvt` extension and are located in the `[AUTOPILOT_HOME]\mconsole\views\autopilot\custom` directory.
   - **Save:** Click **Save** to save the current favorite view.
   - **Save As:** Click **Save As** to save a modified Favorite View. It is recommended that the built-in Favorite view remain unaltered. Once modified, rename the new view.
   - **Built-In Favorite Views:** As part of the file pull-down menu the built-in favorite views are listed for quick reference. The views listed will vary with installation options exercised. The Customized feature and personal views will not be listed, and will require access through *Open,* above.

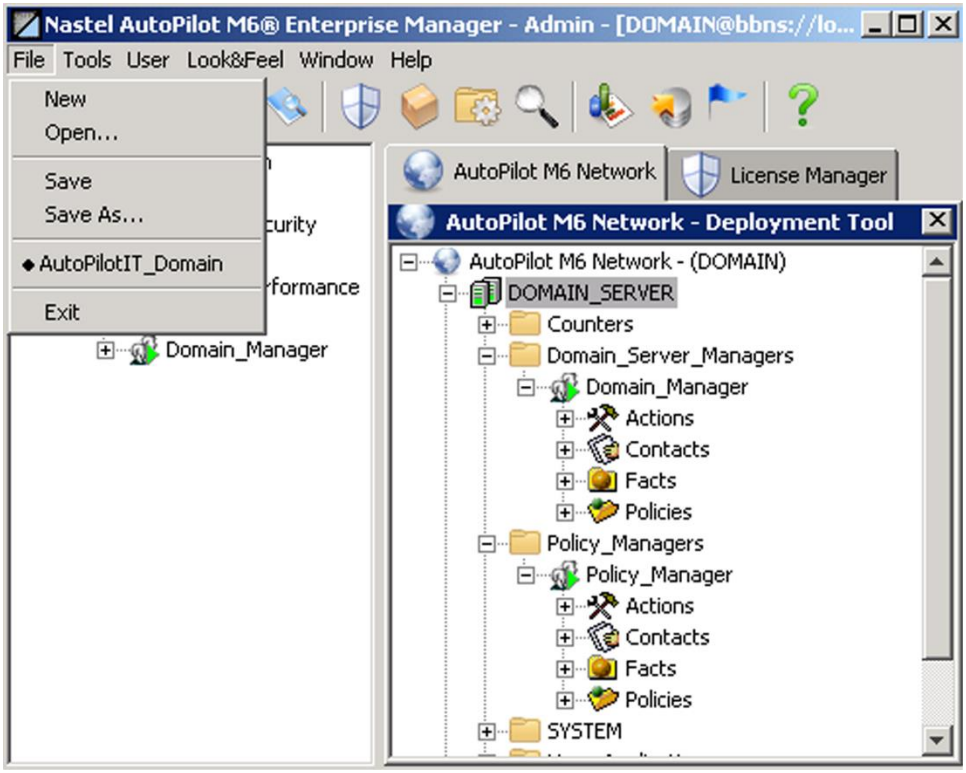


Figure 3-15. M6 User Console - File Menu

2. **Tools:** Click **Tools**, *Tool Menu* is displayed. Clicking on items listed in Tools Menu will open corresponding screen/view. Tool menu contents are same as tool bar, with exception of help screen, which is included on the toolbar. Click required tool to bring it into view. Each screen has all required links, menus, and toggles. As each screen is displayed, it is also listed in the Windows menu.

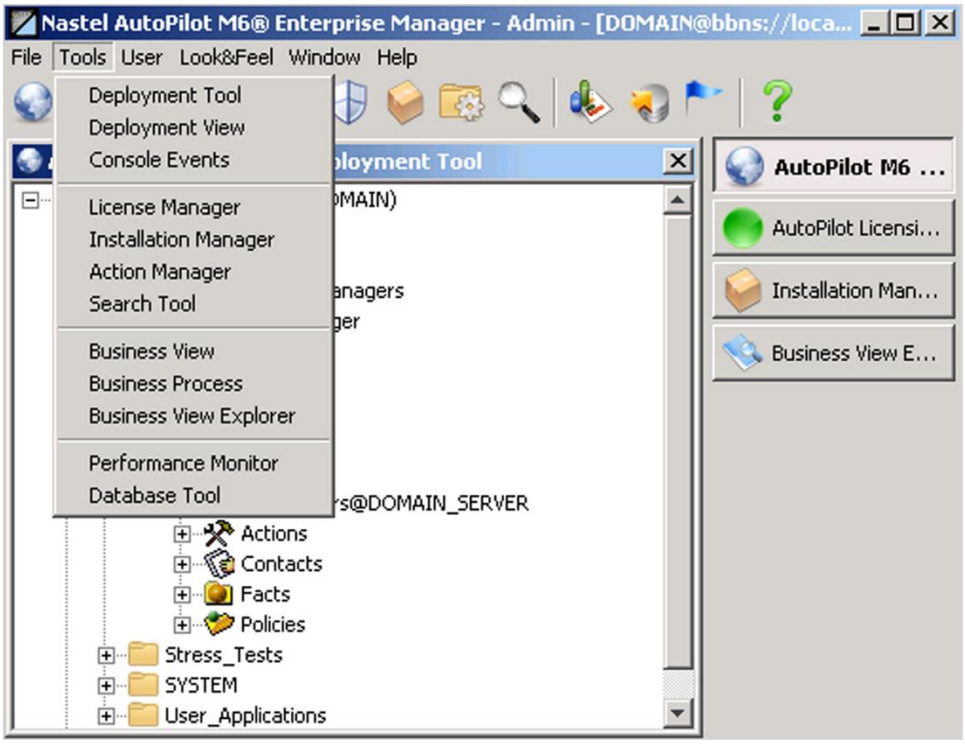


Figure 3-16. M6 User Console - Tools Menu

3. **User:** Click **User** to display pull-down menu.

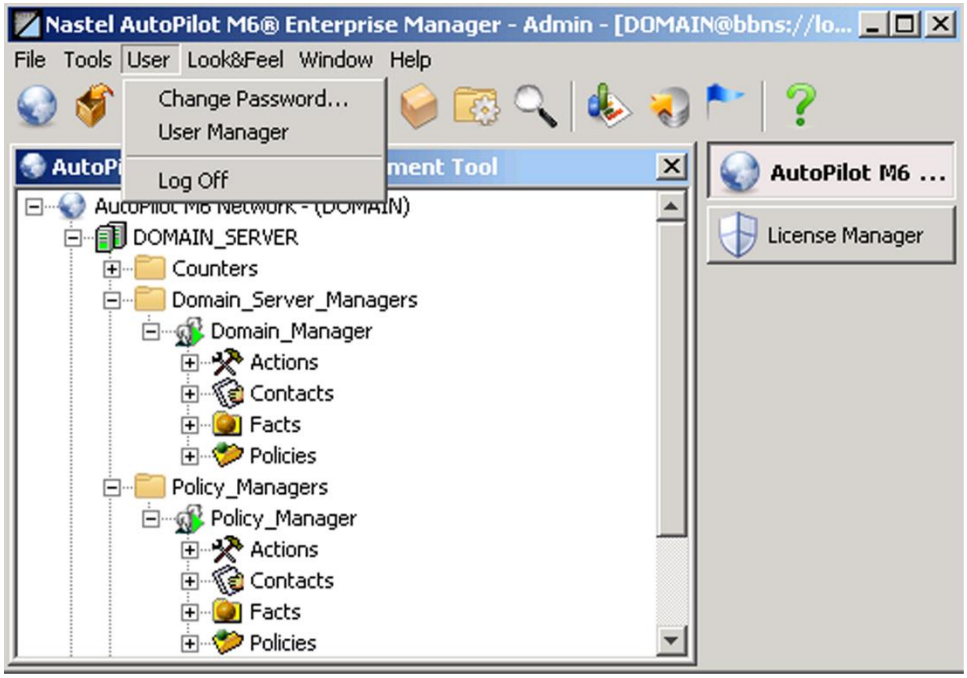


Figure 3-17. M6 User Console - User Menu

- Click **Change Password** to display *Changing Password* dialog box. Users can only change passwords if your admin enables password security permission for that user.

Figure 3-18. Changing Passwords

- Click **User Manager** to display the *User Manager* screen. The **Users** list is displayed by default. Click **Groups** tab to display the list of groups.

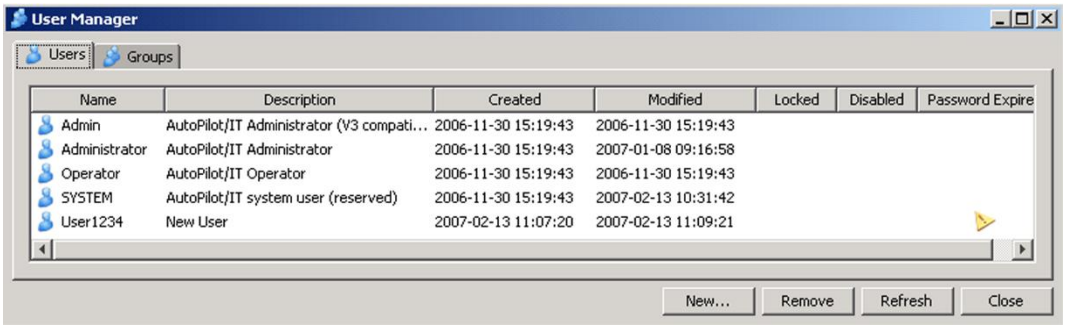


Figure 3-19. User Manager

- Click the **Groups** tab to display the list of user groups. Double-click a group or right-click and choose **Properties** to display the details for that group. There are four tabs, *General*, *Member of*, *Permissions*, and *Misc*. The admin manages the settings in these screens.

NOTE

If the user belongs to an LDAP group, the Member of tab will show the group.

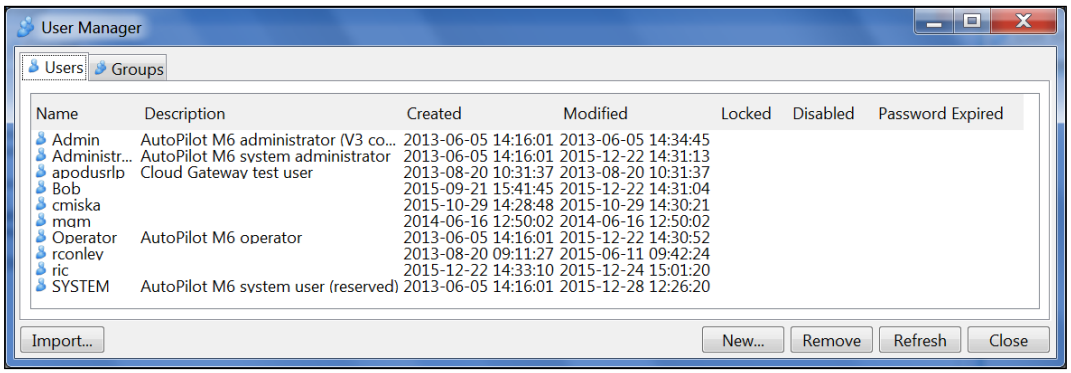


Figure 3-20. User Groups

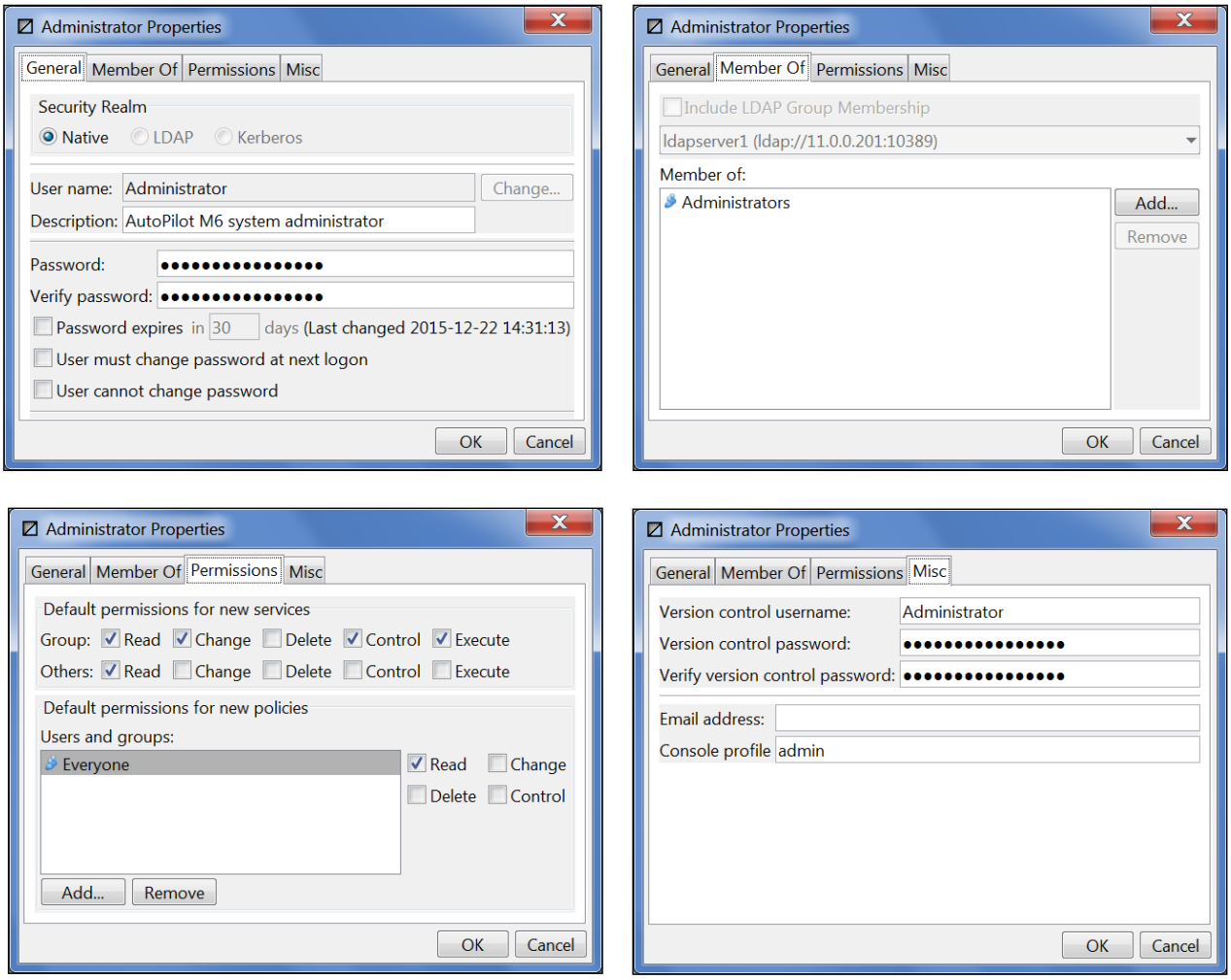


Figure 3-21. User Group Screen Details

4. Look&Feel: Click **Look&Feel** to display a pull-down menu of various ways to display M6 User Console.

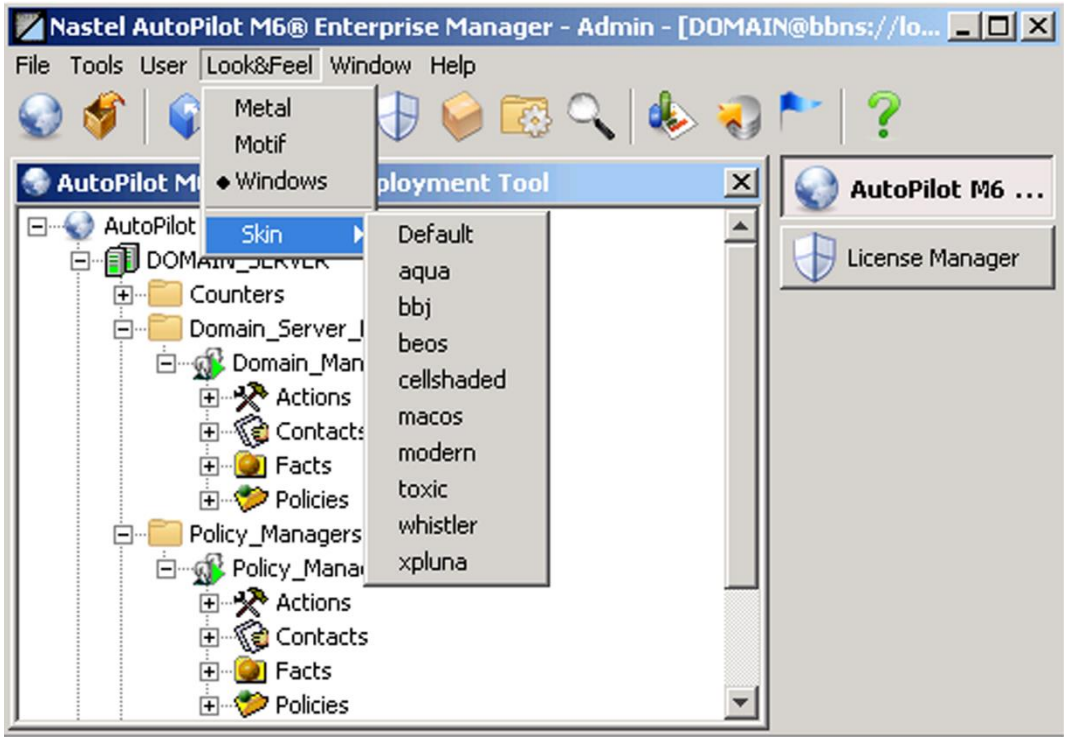


Figure 3-22. Look & Feel Menu

5. **Window:** Click **Window** to display pull-down menu of open windows on right side of screen. Clicking on a listed window will bring that window to the top of open screens.

Figure 3-23. Window Menu

- Click **Tabs** to toggle open window names between top of screen and right side of screen.
- When open windows are separated from the console (drag and drop title bar to remove), the **Cascade** and **Tile** options are activated. Click the option you need to arrange the open windows as indicated.

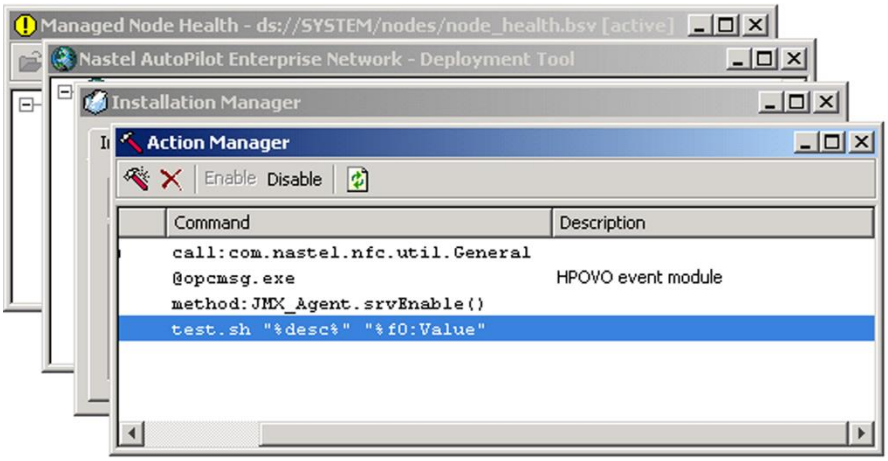


Figure 3-24. Cascading Open Windows

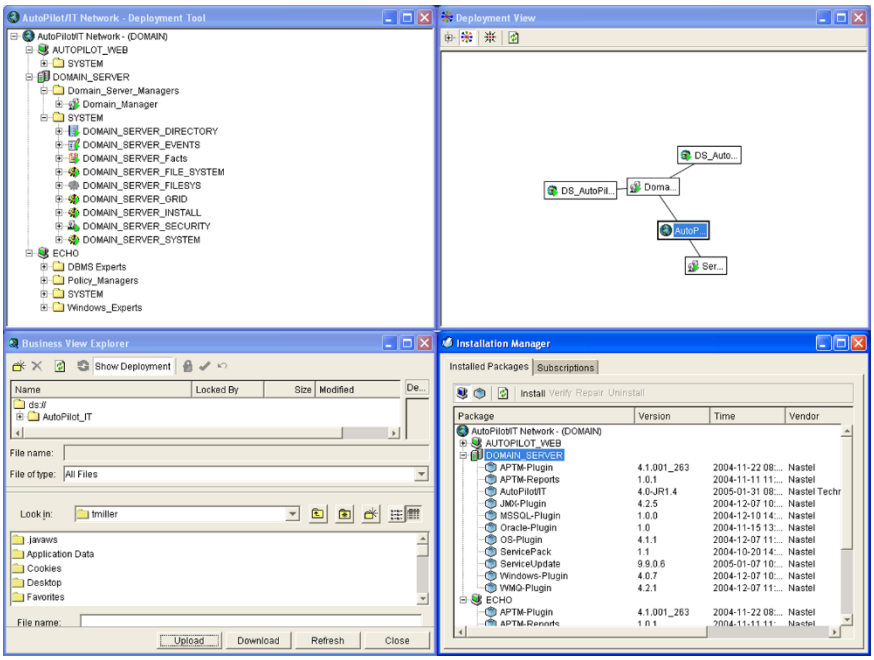


Figure 3-25. Tiled Open Windows

6. **Help:** Click **Help** to display pull-down menu of support topics.

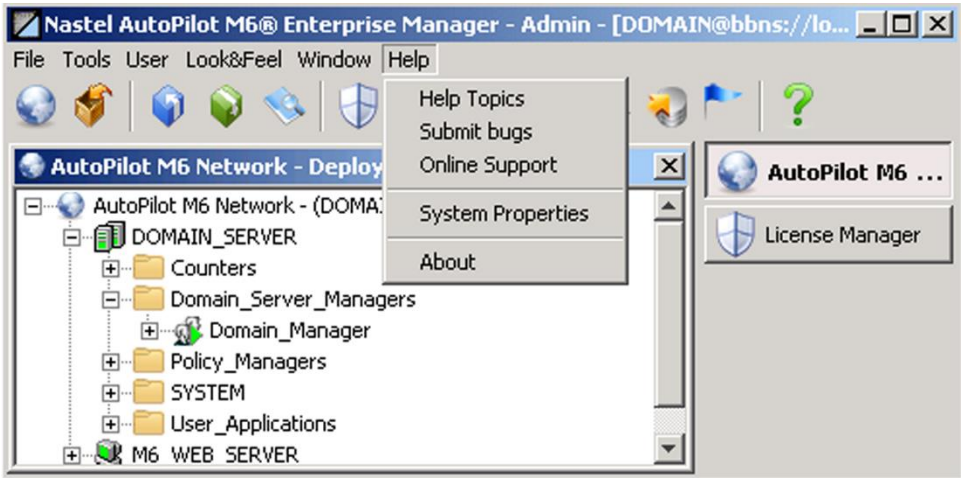


Figure 3-26. Help Menu

NOTE

The HTML page contains links to all available documents and reference material loaded at the time of installation. If the documents were not loaded during installation, they are available on the M6 installation media and at the Nastel Support web site at http://support.nastel.com/

- Click **Help Topics** to access the Nastel Resource Center.
- Click **Submit bugs** to open the Nastel bug reporting screen. Username and Password are required.
- Click **Online Support** to access the Web based Nastel support system. Your *User ID* and *Password* login must be obtained directly from Nastel. Please contact Nastel support at support@nastel.com for more information. Check with your administrator about access privileges.

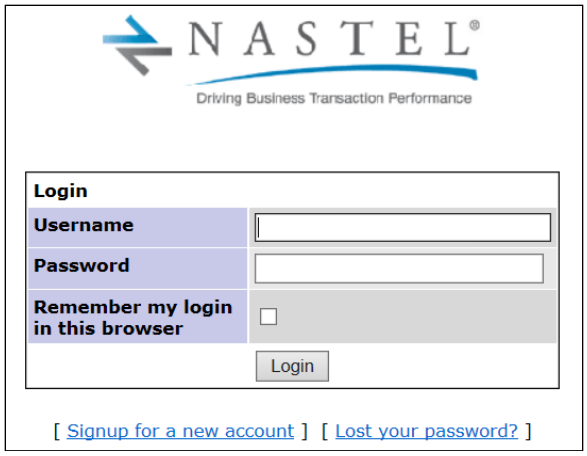


Figure 3-27. Nastel Online Support

- Click **System Properties** to display the system properties screen. It contains all relevant Java system information. Scroll down to review all available system properties. This information will be useful for Nastel support personnel in case of a problem.

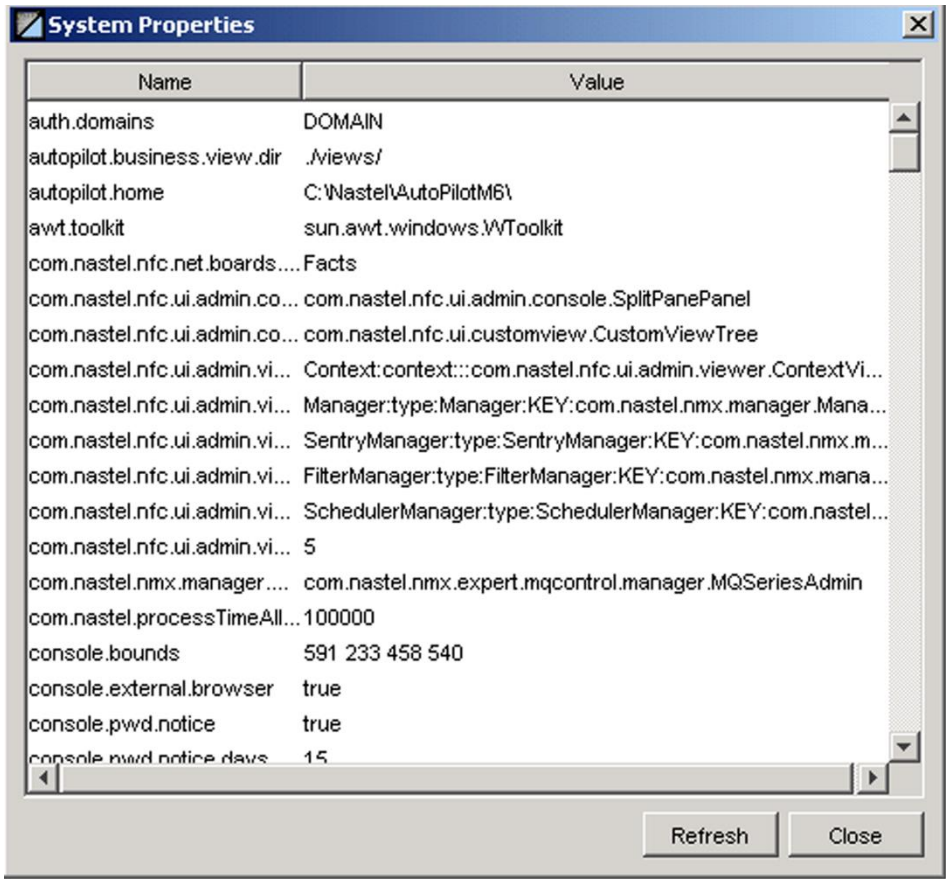


Figure 3-28. Console Java System Properties

- Click **About** to display M6 version information. The information includes the version you have installed as well as the build number. This information is important when contacting Nastel Support.

Figure 3-29. About AutoPilot M6

Main Toolbar

The main toolbar icons correspond to the pull-down menu items.

Figure 3-30. Main Toolbar

Tab Selection

Easily navigate to previously opened screens by clicking on the tab for the screen you want to view. If you would like to view the tabs above the screen, as in earlier versions of AutoPilot, click **Window** to display the pull-down menu and then click **Tabs**.

3.2 Getting Started with M6 Web Console

M6 Web Console is a web-based interface to your M6 domain that enables you to monitor your vital information and processes from anywhere you have Internet or Intranet access. There are no special requirements, just that you have a current browser java plug-in (JRE 1.7 or higher) installed on the machine you are using.

TIP

Nastel recommends the use of Internet Explorer 5.5 with IE Java plug-in 1.7 or higher for use with M6 Web Console.

1. To access the M6 Web Console: from an open web browser such as Internet Explorer, open `http://[server]:8080/m6console/logon.jsp`, where [*server*] is the host name of the M6 Web server.
2. Enter your M6 *User ID* and *Password.*

3. If configured by your AutoPilot administrator, the *Security Profile* field provides the name of a specific security profile used for logon. If not configured, this field should be left blank. If an invalid field is provided, the logon may be unsuccessful.
4. Click **Logon**.

Figure 3-31. AutoPilot M6 Web Console

3.2.1 M6 Web Console Overview

M6 Web Console allows users to access application availability, performance, and health from any location. It allows users to:

- Check the status of application environment
- Acknowledge problems and conditions
- Perform corrective actions
- Lookup history and impact of the failure on other applications or environments

3.2.2 M6 Web Console Functionality

The following describes the functions and settings of the M6 Web Console.

- **Change Password** to open user *Change Password* screen.
- **Log Off** to exit M6 Web Console.
- **Map/Tree Views** toggles between tree view and map view of business views, managers, and domain.
- **Show Managers** opens or closes the active managers.
- Manually **Refresh** to update the business view hierarchal menu display. In addition, the menu is routinely updated by the user-defined refresh cycle rate.
- Set the automatic *Refresh* rate in seconds as needed
- Enter Search parameters in the *Search* dialog box Search: , press **Enter**.

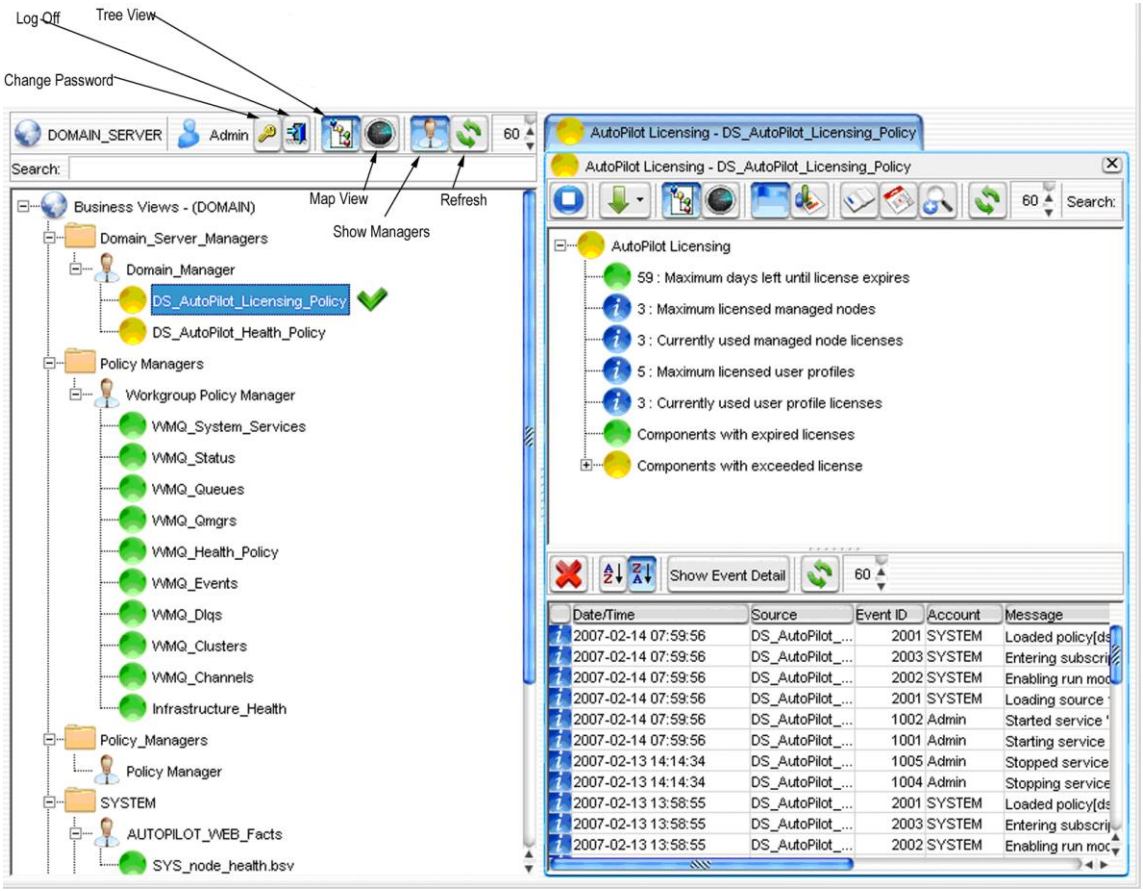


Figure 3-32. Typical M6 Web Console Screen

- **Stop Service** to close the active managers.
- **Arrange Dynamic Sensors** to sort temporary sensors by Name, Severity, Value, Health and in Descending and Ascending order. Only sensors selected as temporary from the sensor wizard can be sorted.
- **Map/Tree Views** toggles between tree view and map view of the business view.
- Click **Show Events** to display table of events.
- Click **Show Chart** to display or close charts. Charts can only be shown for sensors that have logging enabled (see Maintaining Sensor History for details).
- Click **Show Descriptions** to display the description of the selected object.
- **Show Facts** used in a given sensor to determine health.
- **Show Related Views** displays all active views related to the current view.
- Manually **Refresh** to update the business view hierarchal menu display. The menu is routinely updated by the user-defined refresh cycle rate.
- Set the *Refresh* rate in seconds as needed 60.
- Enter Search parameters in the *Search* dialog box Search:, press **Enter**. Enter the word, letter, or acronym required in the search parameter. The corresponding findings will be highlighted in gray in the menu field.

CAUTION!

Using the **Clear Events** button deletes all events. Specific events cannot be selected for deletion in AutoPilot M6 Web Console.

- **Clear Events** . Once deleted the events are not recoverable.
- **Sort** in ascending or descending order by age of the events.
- **Show Event Detail** Show Event Detail displays the selected event detail in the dialog box at the bottom of the screen.
- Manually refresh to update the business view hierarchal menu display. The menu is routinely updated by the user-defined refresh cycle rate.
- Set the *Refresh* cycle rate in seconds as needed .

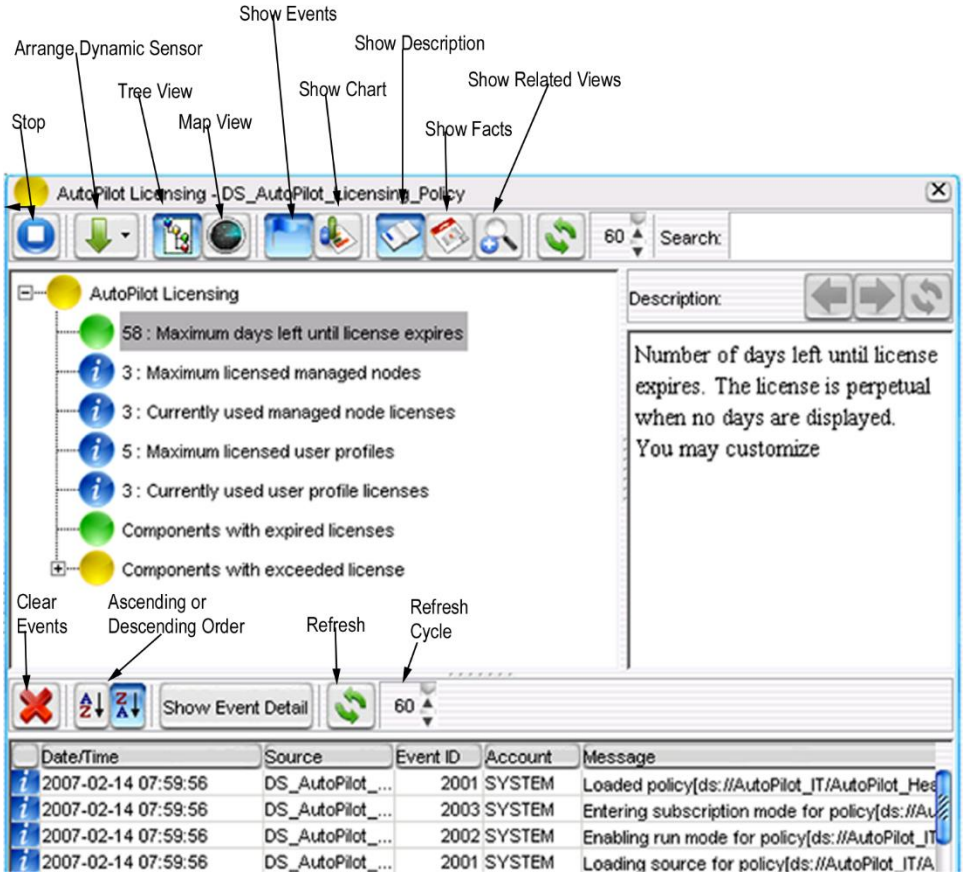


Figure 3-33. Getting Around M6 Web Console

Business View Policy Details

1. Right-click a given Business View or Policy to display Health with the listing as shown in the figure below.

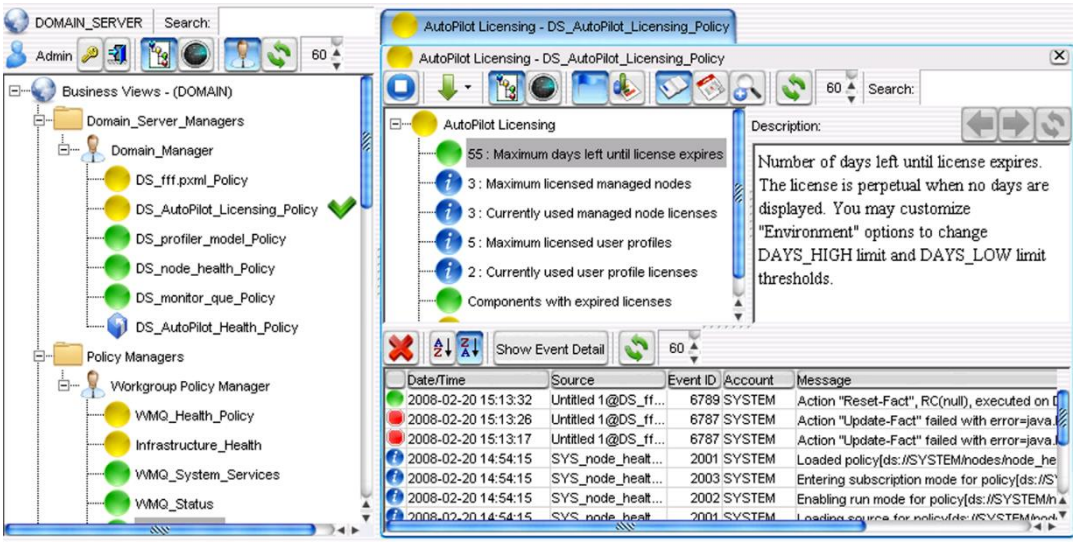


Figure 3-34. Policy and Business View Details

2. Unique to M6 Web Console, selected sensors can be copied to the system clipboard from the status menu. Right-click the sensors to be copied. After clicking **Copy to clipboard**, each sensor is converted into a string with one sensor per line as follows:

   severity health:value:sensor_name

3 As in M6 User Console, the expanded windows for policies and business views can be separated from the primary screen to allow you to expand for detail. Click any fact to get detailed information. Right-click to display the status menu, the listed status reflects your settings from AutoPilot M6 User Console and cannot be changed in M6 Web Console. Click **Action** to access the sub-menu and click the function you require.

NOTE

Clear-Related-Facts action only clears facts under the same folder

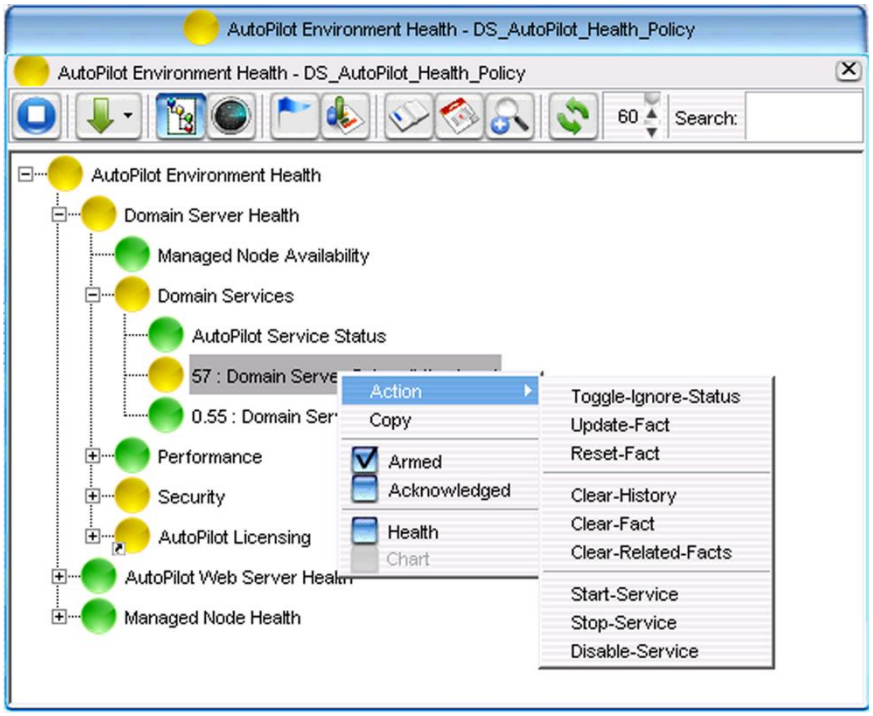


Figure 3-35. M6 Web Console Operator Actions

4. To view a business view chart, the business view must be logged to either the database or file.
5. To improve performance and response time, a business view's cache timeout can be adjusted using the following property:

   servlet.cache.timeout=5

   where 5 is the default value in seconds. The business view is cached for 5 seconds until it is refreshed from the source server.

   This property can be added to one of the following:

   [AUTOPILOT_HOME]/global.properties

   or

   [jakarta-tomcat]/webapps/autopilot/node.properties

   This is helpful when many users are using the same business views.

3.3 Installation Manager

The Installation Manager, which is available through the M6 User Console, allows users to manage M6 software from one centralized point. A user can view, install, verify, and repair all packages from one screen. All packages are located under ds://software folder on the domain server.

Under the *Installed Packages* tab, there are two view screens:

- **View by Server** – what nodes within a domain the packages are installed on
- View by package name – name of package

Both screens display Version Number, Time package was installed, Vendor name, File Name given to package, and Size of package.

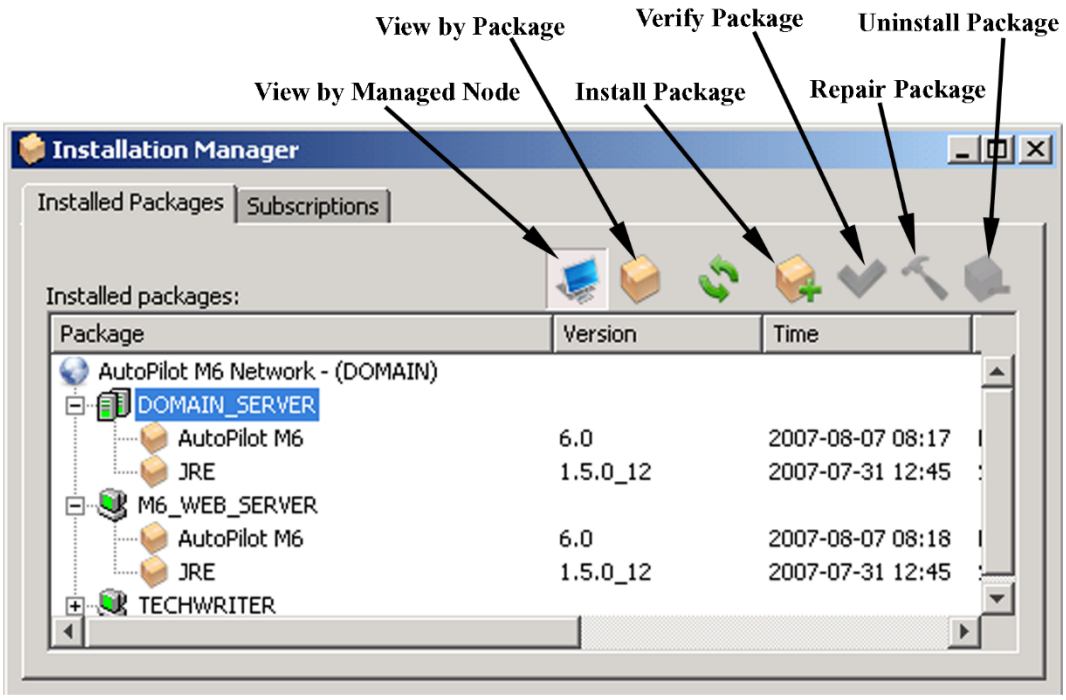


Figure 3-36. Installation Manager – CEP Server View

The *Subscriptions* tab allows the user to create update groups. These groups are a collection of updates that are automatically pushed out and installed on all active subscribers. Three update groups are provided. The user must uniquely configure each group by placing the appropriate files in each group.

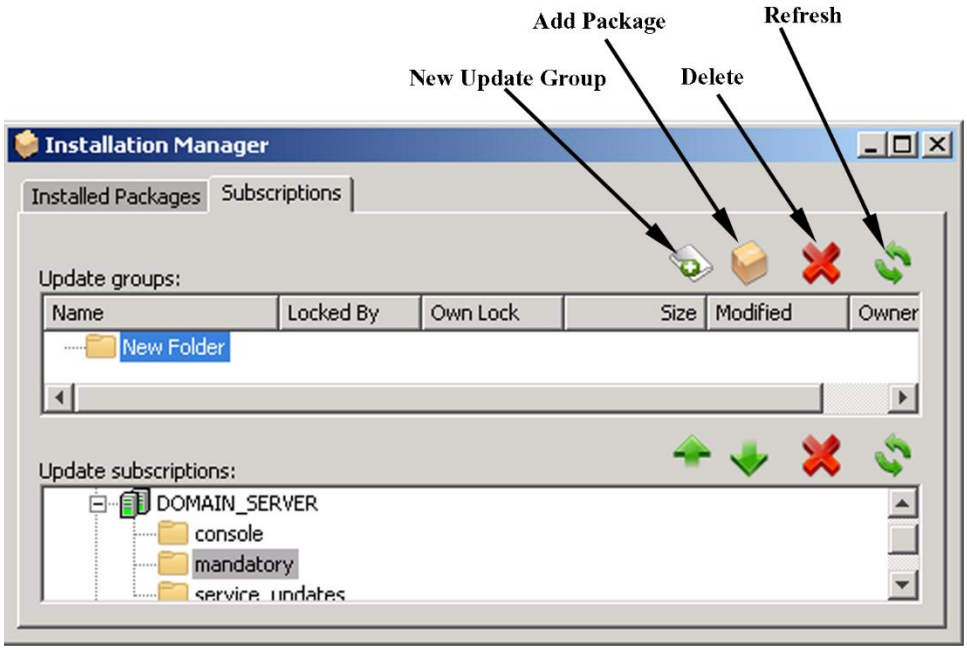


Figure 3-37. Installation Manager – Subscriptions Tab

Table 3-1. Update Groups

Update Group	Description
console	Maintains updates designed for Nastel AutoPilot M6 User Console installations only.
mandatory	Maintains updates that are mandatory for all AutoPilot M6 installations.
service_updates	Maintains all Service Updates.

After startup, every M6 user console, CEP server and domain automatically checks its list of subscriptions and downloads and installs all new and updated packages.

This Page Intentionally Left Blank

Chapter 4: Administering AutoPilot M6

This chapter describes M6 administrative tasks and services. The steps described in this chapter are based on default M6 configuration; your configuration may vary based on installation, platform, and customization.


TIP

The "/" character is a reserved symbol and cannot be used as part of any name in AutoPilot M6.

4.1 Licensing

M6 has introduced a new licensing model that includes a single license file and is based on the maximum number of running CPUs. The standard evaluation license is issued for 15 days. It will run the domain server, CEP server and web server. After 15 days, a new license key must be installed. (Refer to section 4.1.3 for obtaining a trial license.) If the Domain Server and the CEP server are on the same machine, then the CPU count is based on the CEP server only. Domain Server CPU count is not used in order to avoid double counting of CPUs.


NOTE

Current users of AutoPilot can still use their existing license to run M6 but must migrate to the new licensing model within 60 days by contacting Nastel Support.

4.1.1 Obtaining Licenses


NOTE

If a firewall is present, automated license activation (step 2 below) may fail. In this case, the user can obtain a license at the following:
http://www.nastel.com/license/activation.jsp.

To obtain and install an M6 license, do the following:

1. Click the license manager icon from the main toolbar to display the *License Manager* screen.

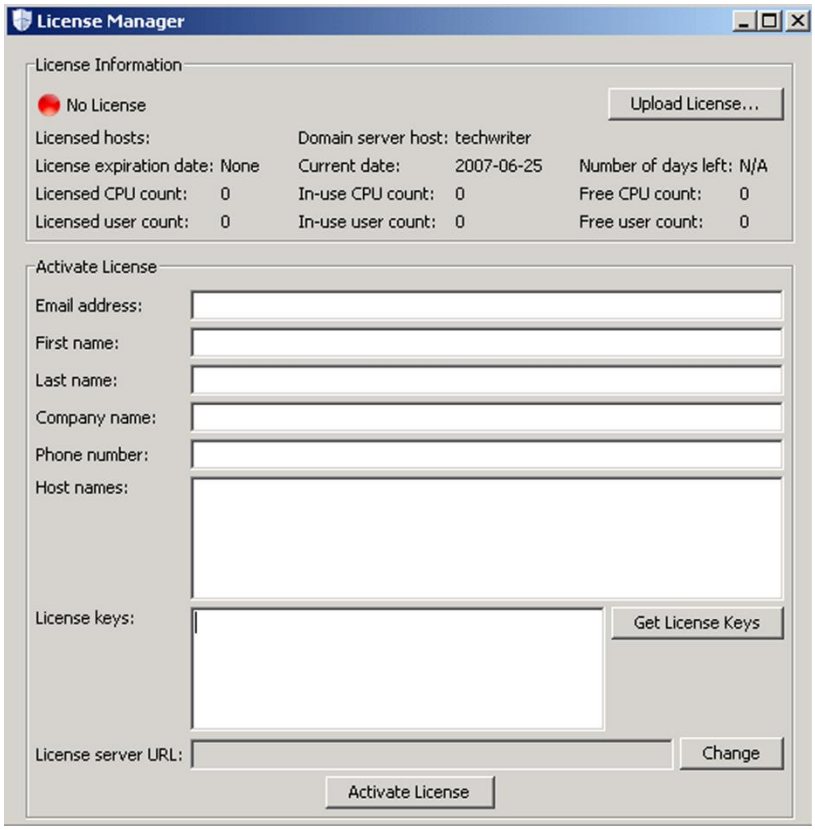


Figure 4-1. License Manager

2. Click **Get License Keys**. The following screen is displayed.

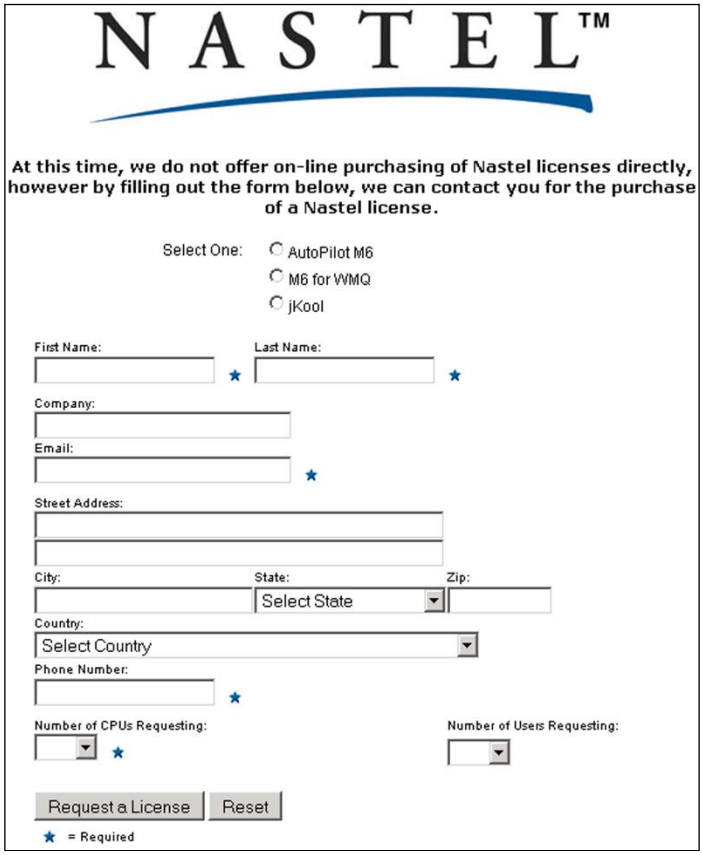


Figure 4-2. Purchasing an AutoPilot M6 License

3. Select *AutoPilot M6* and fill in the form ensuring that all fields with an asterisk are completed.
4. Click **Request a License**.

   An e-mail containing a license key(s) for the number of CPUs and Users purchased will be returned.
5. After the e-mail is received, enter your information in the provided fields including the Host name(s) and license key(s) you received in the e-mail. Click **Activate License.**
6. Verify license status as described in Checking License Status section below.

All license related events are recorded in "License" log located on the Domain Server. This log can be accessed using M6 "Event Viewer". License usage information can be accessed via a built-in business view shown below. *DS_AutoPilot_Licensing_Policy* by default is deployed within *Domain_Manager*.

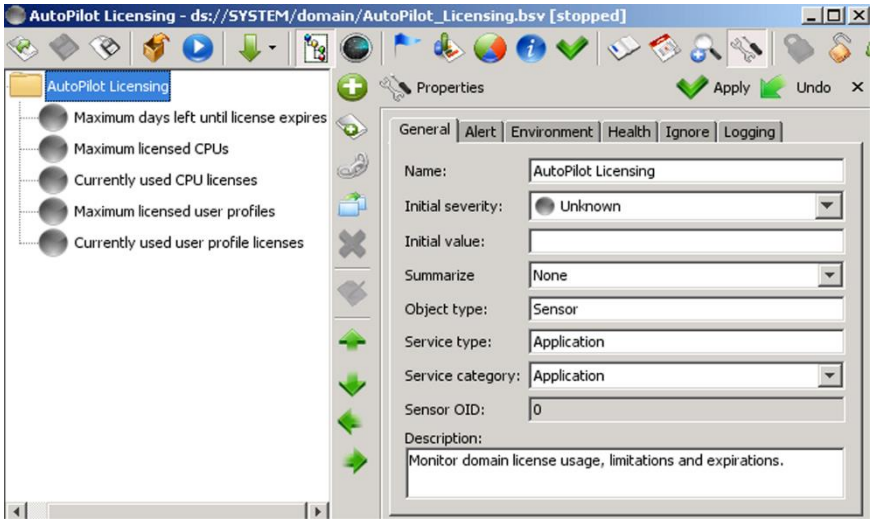


Figure 4-3. License Usage Monitor

4.1.2 Checking License Status

License status, limitations, and conditions can be reviewed using the *aplic* command. You should check the status after license updates are installed. From the command prompt type **aplic**, the current license information will be displayed (see C.3 APLIC – License Manager for more details).

Table 4-1. License Status (APLIC)

Property	Description	
scope=	Name of localhost scope. Scope is defined as a list of servers where license is valid. By default scope represents the name of the local server. The scope can be overridden using -s command line option, to test license for a specific host or a list of hosts (see C.3 APLIC – License Manager for more details).	
localhost=	Host name of the local node.	
license key=	Location of license file.	
license url=	License file URL.	
License Details	*component*=Licensed component licensed_hosts=Names of hosts where license is valid *max_usage*=maximum instances allowed	*expiration_date*= current expiration date *expired*=whether the license is expired *inscope*=whether the license is in scope
Summary	*total*=total number of licenses components *expired*=total number of expired licenses *out-of-scope*=total number of licenses out of scope	

4.1.3 Obtaining a 30-Day Trial License

> **NOTE** If a firewall is present, automated license activation (step 2 below) may fail. In this case, the user can obtain a license at the following: http://www.nastel.com/license/activation.jsp.

To obtain a trial license, do the following:

1. Click the license manager icon from the main toolbar to display the *License Manager* screen.
2. Enter your information leaving the *License keys* fields blank. Click **Activate License**. An evaluation license will immediately be sent to the e-mail address you supplied on this screen.
3. Save the license file at `[AUTOPILOT_HOME]\naming`.
4. Return to license screen and click **Upload License**.

5. Start the Domain Server.

4.2 Managing Users and Groups

4.2.1 User Manager

M6 provides two built-in user groups and four built-in users. The *Admin* and *Administrator* (or those with Admin privileges) perform all management of the users and groups. *SYSTEM* account is a reserved account and used by M6 system services and servers.

User Manager: Open the *User Manager* by opening the **User** menu, then selecting **User Manager**.

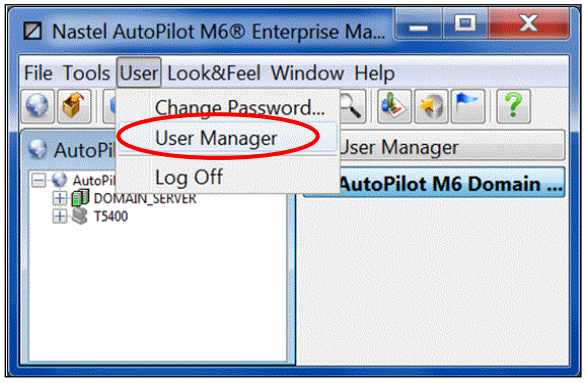


Figure 4-4. Opening User Manager

NOTE When creating a new user or making changes to an existing user, the changes are logged to a file in the Domain Server. These audit records are viewed by right-clicking the Domain Server and selecting **View Events** from the popup menu.

The **User Manager** displays two folders, **Users** and **Groups**.

Users: The **Users** folder (Figure 4-5) is the default view. It lists all the registered users in your M6 Network. It provides information about the user (Table 4-3) and links you to individual user profiles and status. Click a user record to view the user properties.

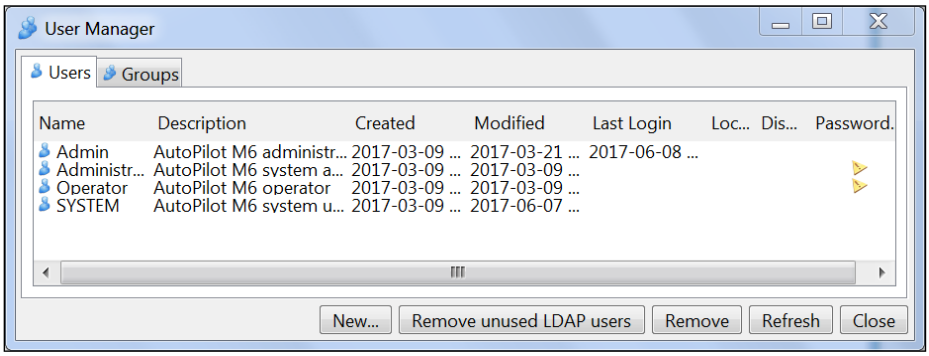


Figure 4-5. User Manager: Users Folder

Table 4-2. User Manager

Category	Description
Users folder	Displays a listing of all users registered in the M6 domain
Group folder	Displays a list of all user groups in the M6 domain
Import	Allows user or group definitions to be imported from an external security server (e.g., LDAP). Only available if LDAP support is enabled.

New	Allows you to add a new user or group
Remove unused LDAP users	Allows you to delete unused LDAP users
Remove	Removes a selected (highlighted) user or group
Refresh	Refreshes the User Manager
Close	Closes the User Manager

Table 4-3. User Manager: Users

Category	Description
Name	Default or Admin assigned user name.
Description	Descriptive information entered at the time the user was created.
Created	Time and date the user was created.
Modified	Time and date of the last modification to this user's profile.
Locked	The user is locked by the administrator or due to too many logon attempts. An exclamation point icon under the locked column indicates a lockout.
Disabled	An exclamation point icon indicates the user account has been disabled.
Password Expired	An exclamation point icon indicates the user password has expired.

Groups: The *Group* folder (Figure 4-6) lists all users groups in your M6 network. There are two default user groups in the installation of M6: *Administrators* and *Operators*. Only the Admin (or those with Admin privileges) can add, delete, and modify the groups. The *Group* folder provides information about the user group and links you to individual group profiles. The group list provides the following information listed in Table 4-4.

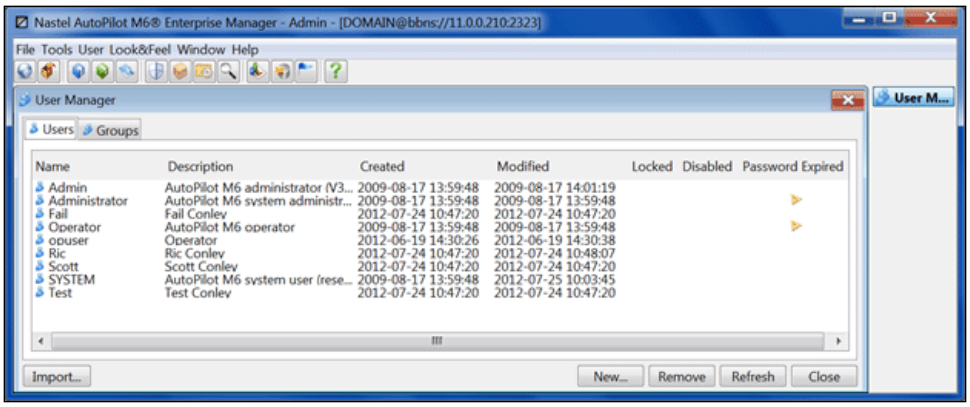


Figure 4-6. User Manager: Groups Folder

Table 4-4. User Manager: Groups

Category	Description
Name	Default or Admin assigned user names.
Description	Descriptive information entered at the time the user was created.
Created	Time and date the user group was created.
Modified	Time and date of the last modification to this group's profile.
Locked	Not used for user groups.

Disabled	Not used for user groups.
Password Expired	Not used for user groups.

User Sub-menu

Right-click any user to open the sub-menu for that user.

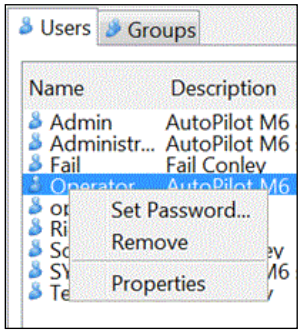


Figure 4-7. User Sub-Menu

Table 4-5. User Manager: User Options

Category	Description
Set Password	Opens *Set Password* screen (Figure 4-8). Enter user's new password in the *New Password* and *Confirm Password* dialog boxes. Password not available when opening group sub-menu. (Not enabled for LDAP.)
Remove	Deletes the selected user or group. The *Confirm Remove Account* screen is displayed, click **Yes** or **No** as needed.
Properties	Displays the selected user or group *Properties* screen (Figure 4-9).

Users with password update/change authority can change their own password. Restricted users must contact the M6 Admin for password or username assistance.

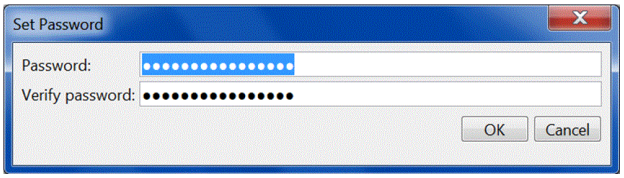


Figure 4-8. Set User Password

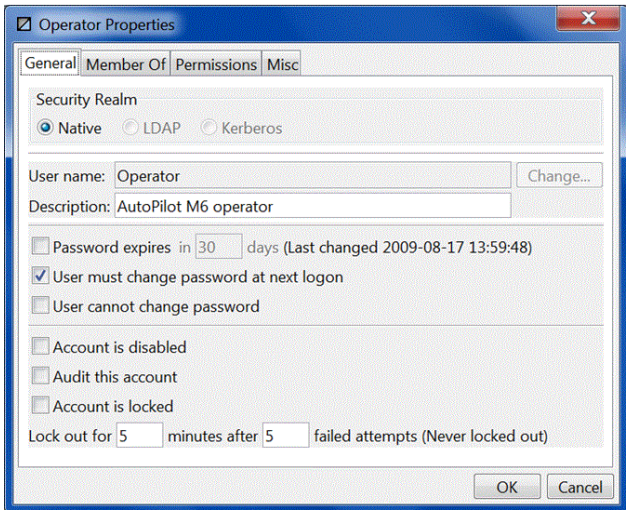


Figure 4-9. User Properties

	TIPS!	1. The number of users that can be added and maintained is governed by your license. Consult your M6 administrator or contact your Nastel representative for further assistance. 2. The **"/"** character (forward slash) is a reserved symbol and cannot be used as part of any name in M6.

Group Sub-menu

Right-click any listed group to open the sub-menu (Figure 4-10). The sub-menu allows you to remove groups or access the group properties.

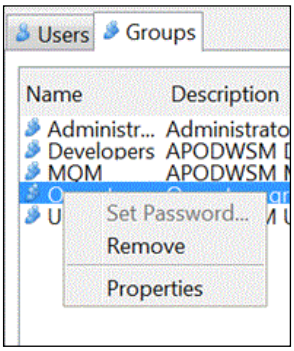
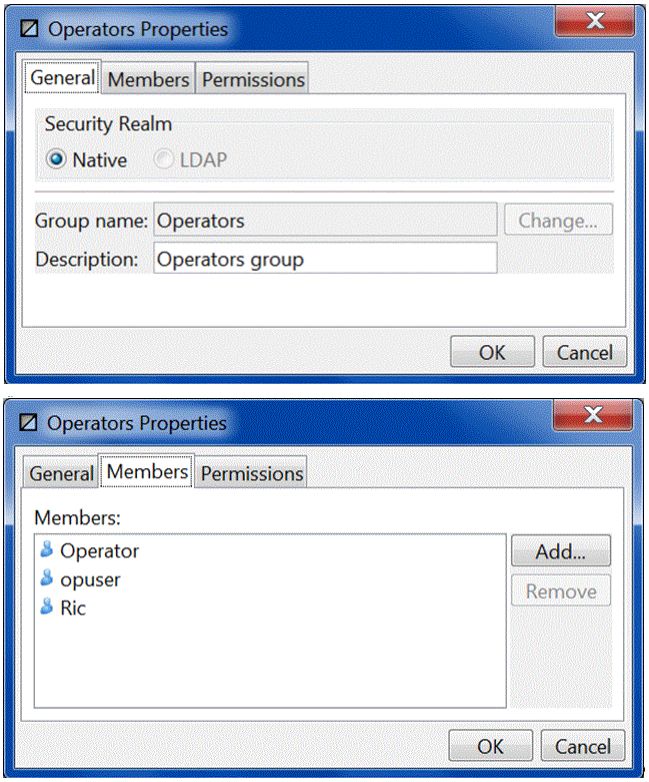


Figure 4-10. Groups Sub-menu



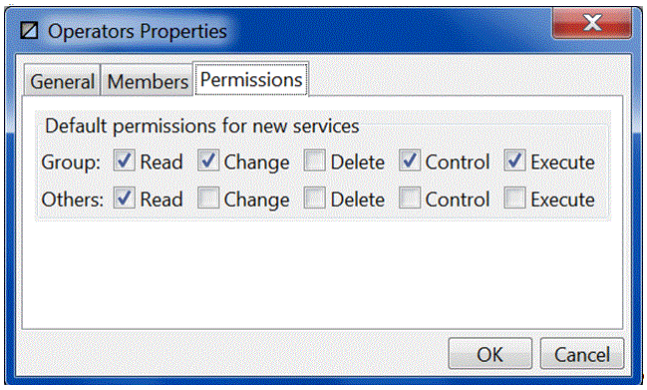


Figure 4-11. Group Properties

4.2.2 Adding Users

1. From the *User* folder (Figure 4-5) in the *User Manager* click **New**. The *New User* screen is displayed.

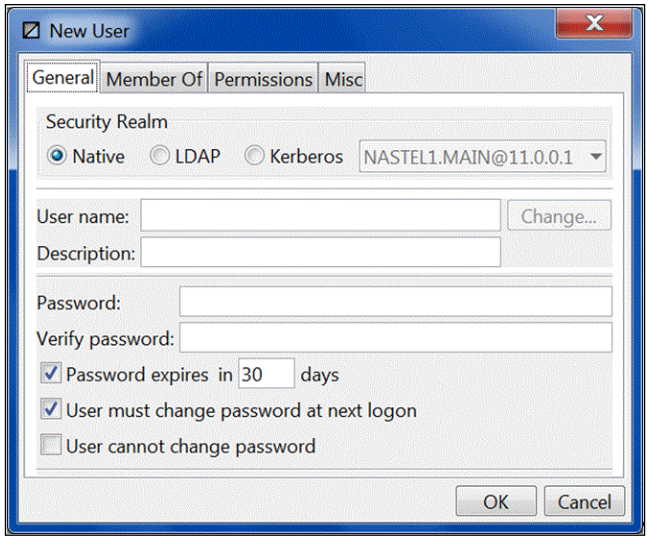


Figure 4-12. Adding New Users

2. Select either Native or LDAP.

   Native: proceed to step 3.

   LDAP: proceed to step 3.

3. **Native:** Enter the user properties as explained in the table in step 4 below.

   LDAP: Specify user name and description.

4. Fill in the following user properties:

Table 4-6. Adding New User: General

Property	Description
Security Realm	Select the type of security model to be used when the new user is logging in.
User Name	User name in accordance with local policy.
Description	Logical user description based on local policy.
Change button	Click to display list of user names added when configuring LDAP.
Password/Verify password	Enter passwords. Password is Admin defined. User can change as needed. See changing passwords in Chapter 4. (Not enabled for LDAP.)
User must change password at next logon	Click **User must change password at next logon** to enable/disable user password change at next logon. For Administrator Use Only. (Not enabled for LDAP.)

User cannot change password	Click **User cannot change password** to enable/disable user changing of password. For Administrator Use Only. (Not enabled for LDAP.)
Password expires in	Default is 30 days. Admin can define per local security policy. (Not enabled for LDAP.)
Account is disabled	Click **Account is disabled** to enable/disable user privileges. The check signifies that the account is disabled.
Audit this account	Click **Audit this account** to enable/disable user audit functions. The Audit function records all actions by the user on the server in the M6 log file. The check signifies that the account audit is enabled.
Account is locked	Click **Account is locked** to enable/disable the user account. The check signifies that the account is locked.
Lock out user for	Locks out the user for an *Admin defined* number of minutes after a defined number of attempts to login have failed.

5. Click **OK**. Right-click newly created user and select *Properties*. Select the **Member Of** tab and click the **Add** button. The *Add User to Groups* screen is displayed.

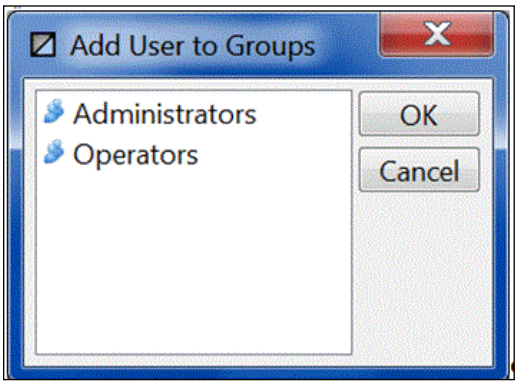


Figure 4-13. Assigning Users to Groups

6. Click the groups the new user is being assigned to. Click **OK**. Users can be assigned to multiple groups as needed. To select multiple groups hold the **Ctrl** key and click the groups desired.

Table 4-7. Adding New User: Member of (Group)

Group	Description
Administrators	System group that has administrative privileges
Operators	Group for the M6 users privileges
Others	Others are groups and users that do not belong to the accounts list of groups.

To have the user's group membership read from the LDAP Server, check the **Include LDAP Group Membership** box.

7. Click **OK**. Right-click newly created user and select *Properties*. Select the **Permissions** tab.

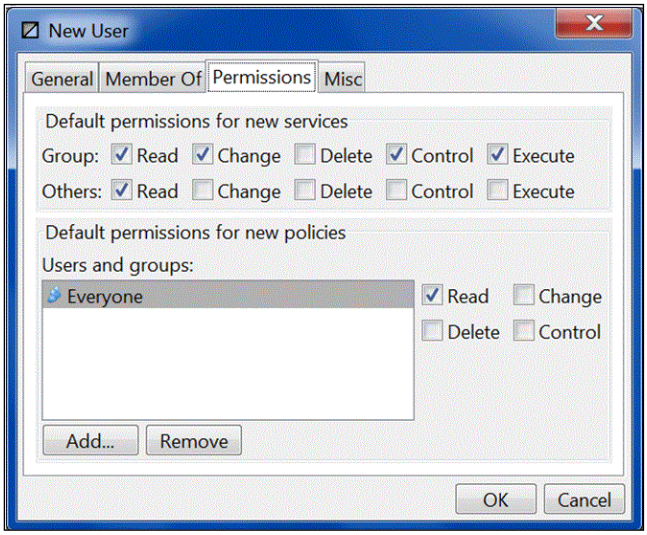


Figure 4-14. User Security Properties

8. Define the user permissions in accordance with local policy. Group permissions extend the checked privilege to all members of the specified user group. When the property is checked for others, users from other groups are extended the same privilege. All objects created by this user will automatically inherit its permission settings.

Table 4-8. Adding New User: Permissions

Permission	Description	
	Group	**Others**
Read	Group members may read/view attributes of an object	Others may read/view attributes of an object
Change	Group members may change the attributes of an object	Others may change the attributes of an object
Delete	Group members may delete the object	Others may delete the object
Control	Group members may execute control actions such as start, stop, disable	Others may execute control actions such as start, stop, disable
Execute	Group members may execute operational commands on the object	Others may execute operational commands on the object

9. Click **OK**. Right-click newly created user and select *Properties*. Select the **Misc** tab.

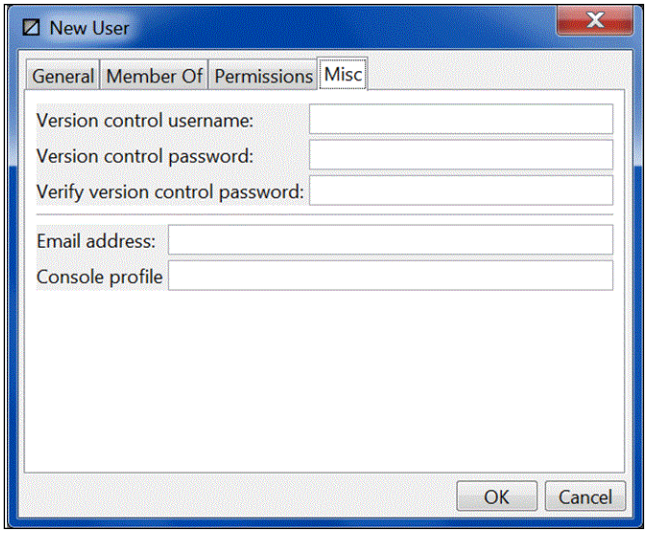


Figure 4-15. User's Business View Version Access

10. Define the new user's ability to update, add, or delete business views.

Table 4-9. Adding New User: Misc (Business Views)

Permission	Description
Version control username	Username assigned to those users requiring access to, and tracking of, business view history and revision.
Version control password	User password to authorize access and to add, delete, or change business views.
Verify version control password	Retype version control password.
Email address	User's email address.
Console profile	Specifies the name of the profile that console loads based on user-defined profiles configured for each user to allow console to be centrally managed. Maps to Domain Server profile directory which is stored under `\naming\profiles\<prof_name>`. Profile must match `\naming\profiles\<prof_name>` directory. Each profile has `profile.properties` file which determines how console is configured. A default profile is loaded for users whose console profile is empty.

4.2.3 Adding Groups

1. From the *Groups* folder in the *User Manager* click **New**.

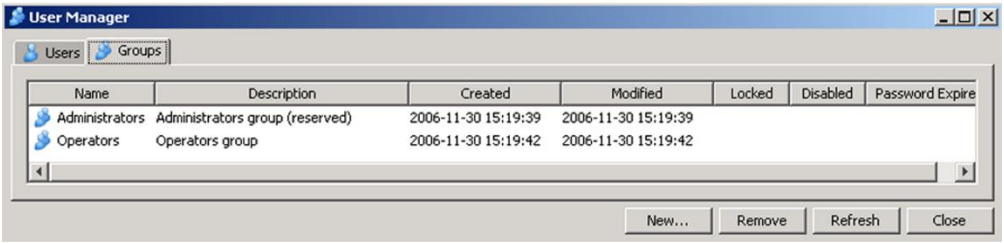


Figure 4-16. Adding New Groups

The *New Group* screen is displayed.

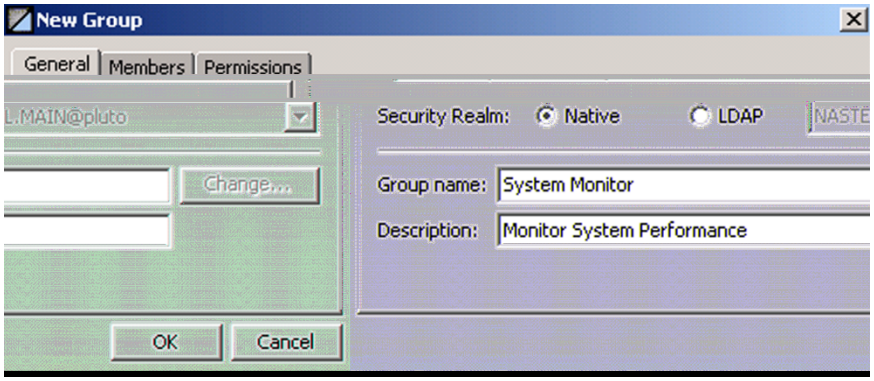


Figure 4-17. New Group General

2. Fill in the group name and description:

Table 4-10. Adding New Group: General	
Property	**Description**
Group Name	Group name in accordance with local policy
Description	Logical user description based on local policy.

3. Select the **Members** tab on the *New Group* screen.
4. Click **Add**. The *Add Group Members* screen will be displayed. Users can be assigned to multiple groups as needed. To select multiple members hold **Ctrl** and click the groups desired. Click **OK** when all members have been added.

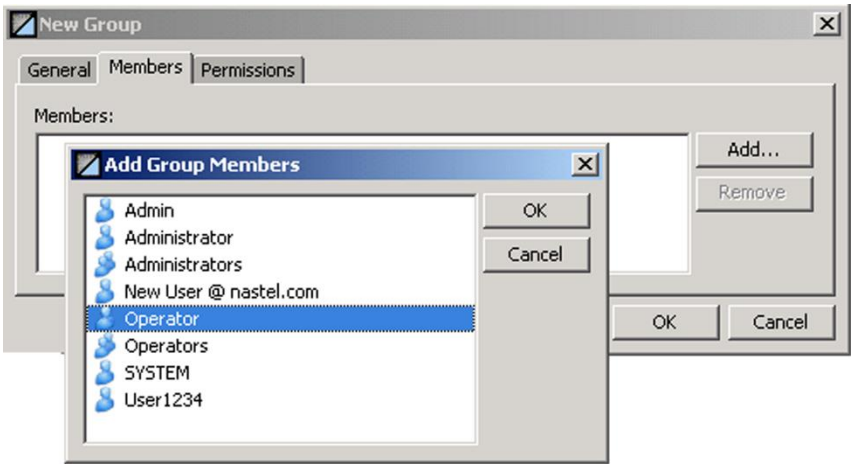


Figure 4-18. Assigning Members to New Group

5. Define the user permissions in accordance with local policy. Group permissions extend the checked privilege to all members of the specified user group. When the property is check for others, users from other groups are extended the same privilege. All objects created by this user will automatically inherit its permission settings.

Table 4-11. Adding Group: Permissions		
Permission	**Description**	
	Group permissions	**Others permissions**
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.

Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, disable.	Others may execute control actions such as start, stop, disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.

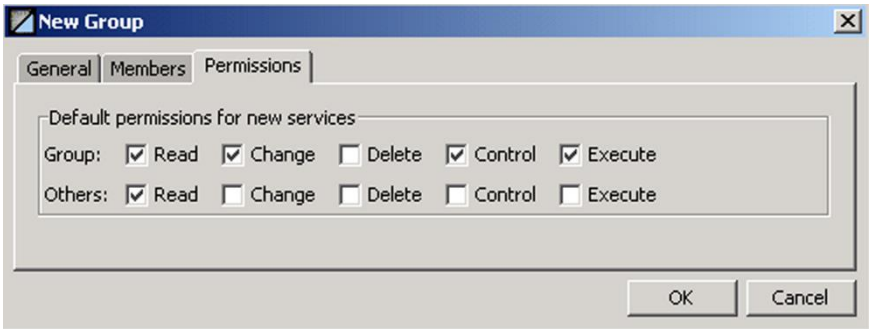


Figure 4-19. User Security Properties

4.2.4 Removing Users

The M6 administrator can remove users as follows:

1. In the *User Manager, User* folder, click the user to be removed. Click **Remove**.
2. In the *Confirm Remove Account* screen click **Yes**. The user account will be permanently removed.

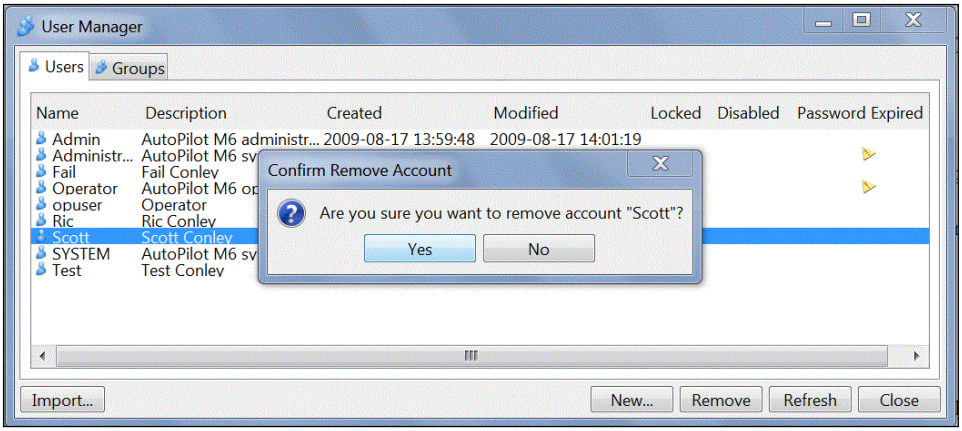


Figure 4-20. Removing Users

4.2.5 Removing Groups

The M6 administrator can remove users as follows.

1. In the *User Manager, Groups* folder, click the group to be removed. Click **Remove**.
2. In the *Confirm Remove Account* screen click **Yes**. The user group account will be permanently removed.

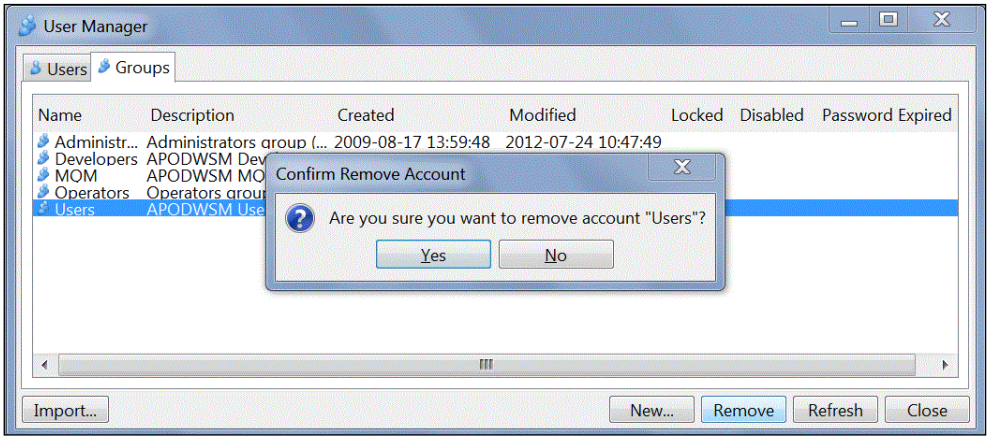


Figure 4-21. Removing Groups

4.2.6 Changing Passwords

> **NOTE** Does not apply to LDAP users.

Password and password format policies should be adopted and observed to prevent unauthorized system access. Default passwords should be changed as soon as possible to prevent unauthorized access. Both the user and the administrator can change user names and passwords. However, the admin can limit which users, if any, are permitted to change their passwords. The permission for users to access and change passwords is determined when the user is added and can be changed any time thereafter. Any password can be changed as needed or as dictated by local policy. Individual users can change their passwords as policy and existing permissions dictate.

> **NOTE** Password length must be at least five characters long (by default) or as defined by *server.security.password.length=5* property configured at the domain server's *node.properties* file.

There are several methods to change passwords based on your M6 authority and permissions.

Changing Your Password

1. From the *User* menu, select *Change Password*. The *Change Password For <user>* screen is displayed.
2. Enter your current password in the *Current password* field.
3. Enter your new password in both the *New password* and *Verify new password* fields.
4. Click **OK**.

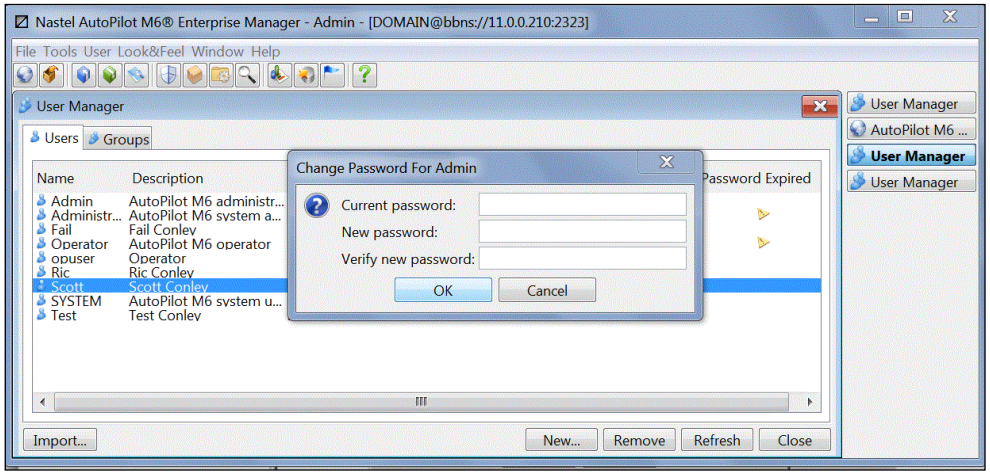


Figure 4-22. Changing User Passwords

Changing User Passwords

1. From the *User* menu select *User Manager.*

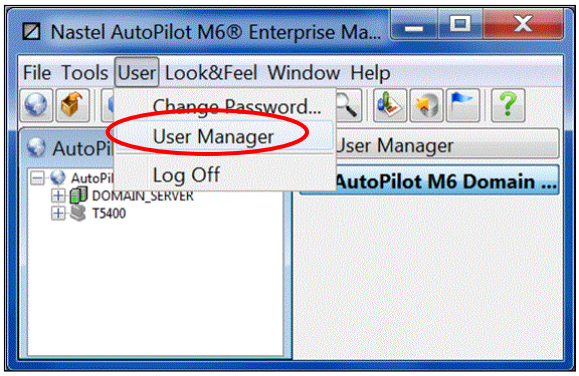


Figure 4-23. Open User Manager

2. From the *User Manager* menu right-click the target user. Click **Set Password** on the sub-menu. The *Set Password* screen is displayed.

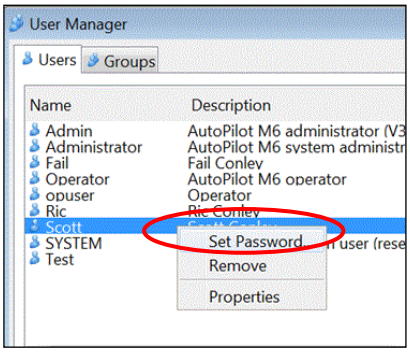


Figure 4-24. Set User Password

3. Type new user password in the *password* and *Verify password* boxes. Click **OK**.

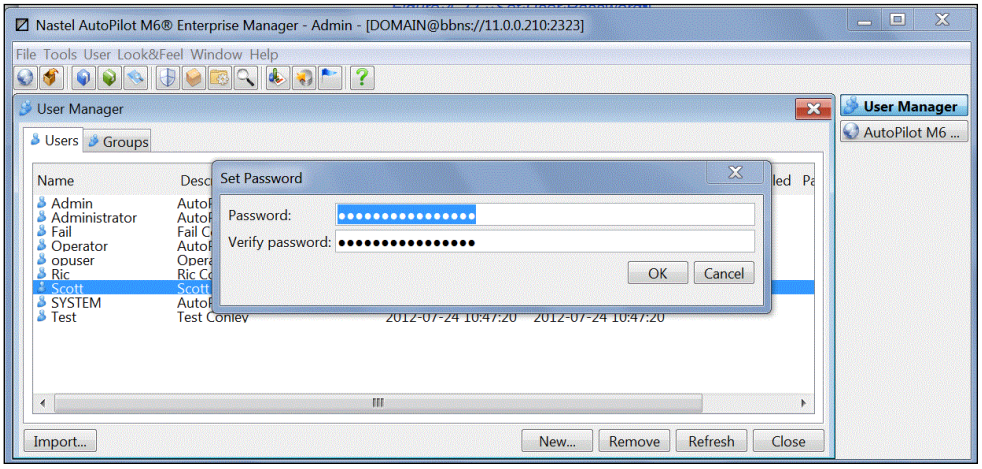


Figure 4-25. Change User Passwords

4.2.7 Importing Users and Groups

If LDAP support is enabled, an **Import** button is available which will allow user or group (based on current view) definitions to be imported from the configured LDAP servers.

To import users and/or groups, click the **Import** button to open the *Import Users or Import Groups* screen.

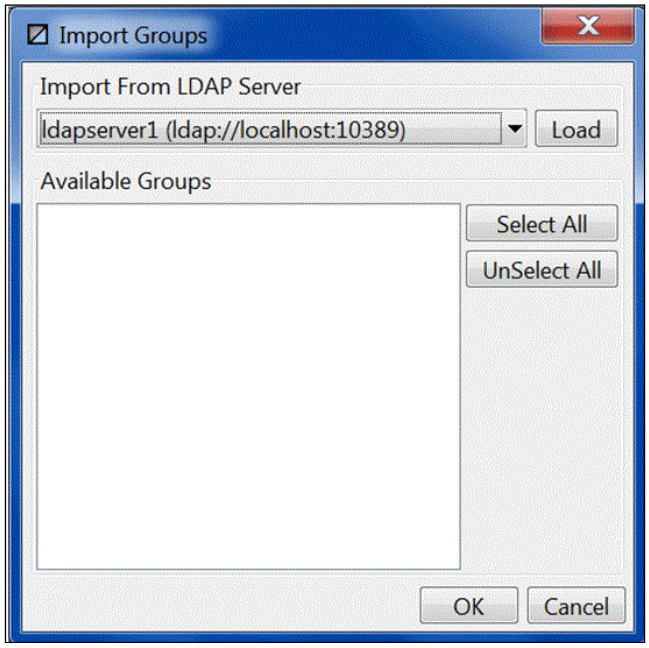


Figure 4-26. Import Users or Import Groups

1. At the top, pick from the configured LDAP servers and click **Load**.
2. The **Available Groups** list will then be populated with all users or groups found at the selected server whose names do not match those of existing users or groups.
3. Select the users or groups to import and click **OK**.

4.3 Service and Account Permissions

4.3.1 Service Permission Mask

The security settings are common for all management services. However, the actual settings may vary depending on owner's permission mask. By default, all services inherit permissions from the owner account.

Table 4-12. Service Permission Settings	
Inherit permissions from owner	Clicking the check box enables or disables the transfer of permissions from the original owner to the owned object. The default is enabled (checked).
Owner	Name of the account that owns the object. The owner can be changed by clicking **Change**.
Permissions	The default mask is defined in the group and other permission settings in your user properties. (Refer to section 4.2.2, Adding Users.) The owner or anyone who has change permission can change security settings.

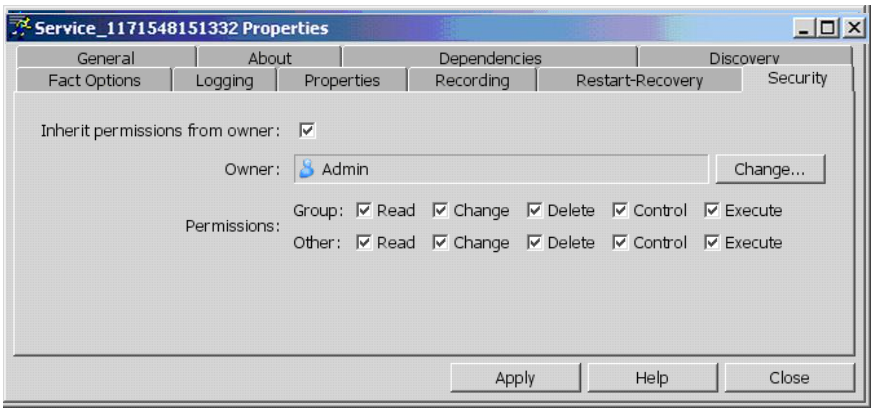


Figure 4-27. Service Permission Mask

4.3.2 Account Permission Mask

Permissions specify access control to objects by group members and those who are not part of the group (others). Owners always have full access to all owned objects. Permissions are inherited from the owner to the owned service such as expert, manager, and policy.

1. The default settings are the same for all new users and groups. The M6 administrator must define the security settings for each group, and if needed, each account.
2. Permissions in the Group and Other categories are enabled or disabled by clicking on the appropriate button. The check mark indicates that the permission is enabled.

Table 4-13. Group and User Permissions		
Permission	**Description**	
	Group	**Other**
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.

Control	Group members may execute control actions such as start, stop, disable.	Others may execute control actions such as start, stop, disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.

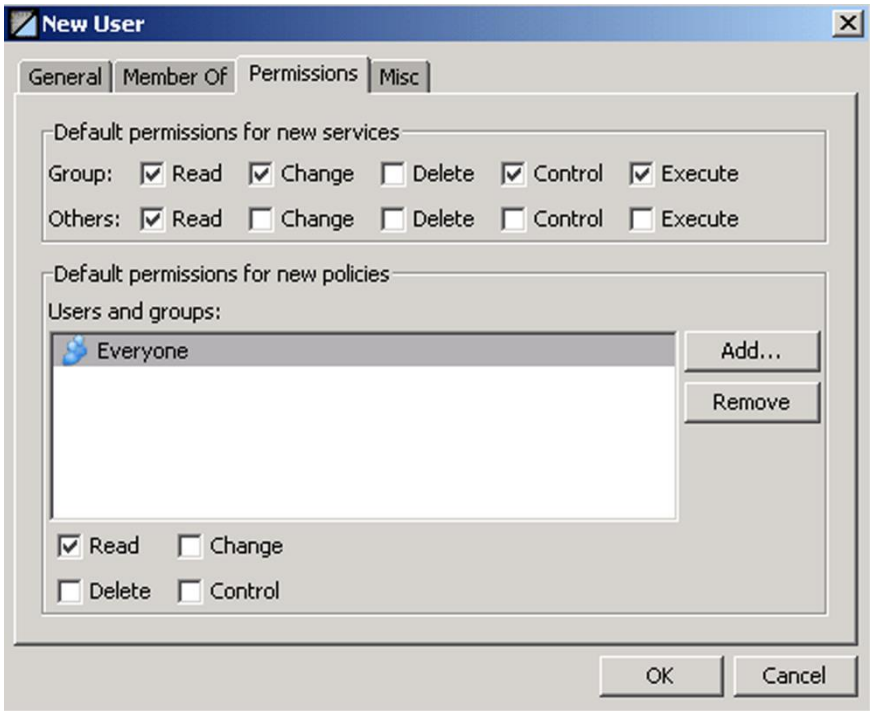


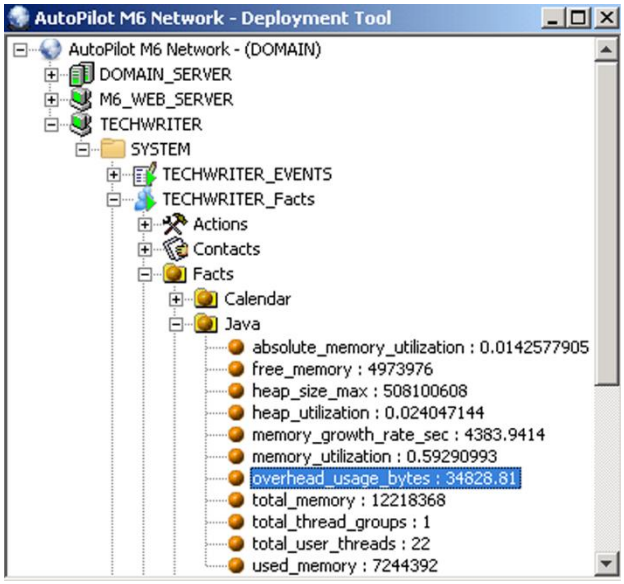
Figure 4-28. Default User Permission Mask

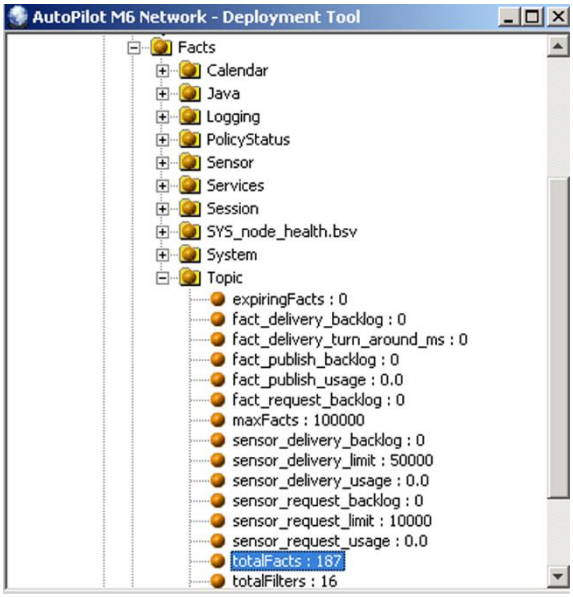
4.4 CEP Servers

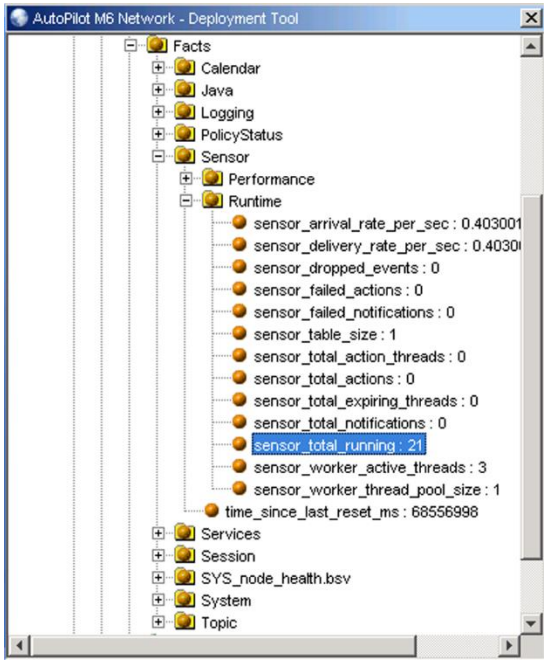
A CEP server is an AutoPilot M6 target-managed resource that holds services such as experts, managers, policies, and sensors. There are also built-in CEP servers that are available immediately after installation such as the Domain Server. From a CEP server you can view events, import services, deploy experts and managers and refresh security (permission settings) to corresponding services. The CEP server icon reflects the status of node_health.bsv to help users spot CEP servers that have performance or resource issues. The icon will remain green unless the CEP server falls into a warning (yellow) or higher state. Stopping a CEP server stops all services associated with it.

4.4.1 Performance Guidelines

Every CEP server publishes information about its memory and fact utilization. This information is published under [*NODE_NAME*]_Facts\Java and [*NODE_NAME*]_Facts\Topic. These facts can be included into business views and can be monitored and alerted on.







CPU Utilization and Estimation

CPU utilization is proportional to:

- number and scope of the business views deployed with the CEP server
- discovery intervals and scope of poll-based experts. For example, WebSphere MQ Experts.

Estimate CPU utilization and load by using the following:

- (Topic\totalFacts\(Average Sampling Rate)) * Sensor\Runtime\sensor_total_running
  Example: (100000/100 (seconds)) * 1000 = 1,000,000 logical operations per second.
- Average sampling rate can be estimated based on the sum of all sampling intervals for all monitors divided by the total number of monitors.

Memory Utilization and Estimation

Memory utilization, *memory_utilization*, is proportional to:

- number of experts and managers deployed
- number of facts published within the CEP server
  (*totalFacts* should be less than *maxFacts*)
- number and scope of business views deployed with the CEP server
  (estimated total_memory~totalSensors * totalFacts * overhead_usage_bytes)
- For example, average memory utilization for a CEP server with 10-15 experts and 50,000-75,000 facts may require 60-90 MB of virtual memory (platform dependent).

Estimate CEP server memory utilization by using the following:

- Java\overhead_usage_bytes * (Topic\totalFacts + Sensor\Runtime\sensor_total_running)

When evaluating required memory for a CEP server:

- Java\overhead_usage_bytes * (planned_total_facts + planned_total_sensors)

To calculate the operations per second of a CEP server, use the following formula:
*((Total number of facts) /(Discovery period of the polling interval of the expert)) * (Total number of sensor rules)*.

4.4.2 CEP Server Communication

The [*server*]_Facts monitor publishes new metrics about CEP server communications as follows:

Table 4-14. CEP Server Communication Metrics

Metric	Description
Session\total_bytes_sent	Total bytes sent across all pipes.
Session\total_objects_recvd	Total received objects across all pipes.
Session\total_objects_sent	Total sent objects across all pipes.
Session\total_objects_dropped	Total dropped objects due to one or more of the pipes being disabled.
Session\total_pipes	Total number of allocated pipes.
Session\total_pipes_disabled	Total number of pipes that no longer accept outbound traffic.
Session\total_pipes_full	Total pipes are filled to 100% capacity (max_send_queue_limit).
Session\total_pipes_idle	Total pipes that are currently idle.
Session\total_send_queue_depth	Total number of objects queued for outbound communication across all pipes.
Session\max_send_queue_limit	Maximum capacity of each pipe.
Session\avg_pipe_usage	Average usage for all pipes.
Session\avg_pipe_arrival_rate_per_sec	Average number of objects being queued for outbound communication.
Session\avg_pipe_delivery_rate_per_sec	Average number of objects being sent out per second.

IMPORTANT!

1. It is important to monitor Session\total_send_queue_depth and Session\total_pipes_disabled.
2. Session\total_send_queue_depth is an indication of slow connections that cannot keep up with the rate at which facts are generated.
3. A pipe is disabled once it reaches 100% capacity utilization. Once a pipe is disabled, it can no longer accept any new traffic and begins to drop any new activity (total_objects_dropped).

4.4.3 Using CEP Servers

Right-clicking on CEP server displays following sub menu. CEP server options are described below.

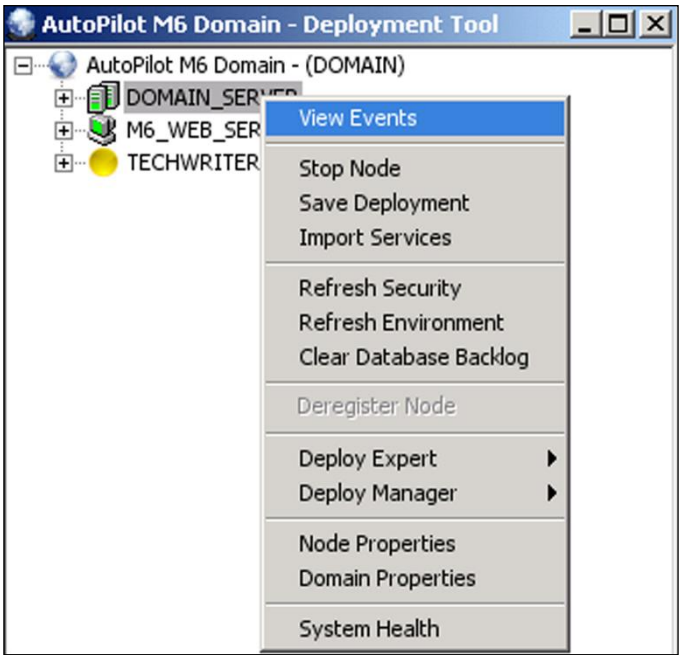


Figure 4-29. CEP Server Options

Table 4-15. CEP Server Options

Option	Description
View Events	Displays the Event Viewer and all events associated with the selected node.
Stop Node	Turns off the CEP server and all of its services.
Save Deployment	Saves the configuration of the CEP server selected.
Import Services	Processes import files. Details are contained in the Event Log.
Refresh Security	Refreshes security (permission settings) to corresponding services. The settings are applied from the owner's permission mask. This option removes the need to restart the CEP server when account security mask is changed.
Refresh Environment	Reloads all previously loaded property files used by business views (for example, global.properties and node.properties).
Clear Database Backlog	Clears database backlog to free up memory. This option should only be used when <*managed_node*>_Facts\Topic\sensor_dbpool_backlog fact is growing too quickly due to persistent database failures. Backlog can also be cleared from the APNET command line as follows: `apnet invoke [managed_node_name]_SYSTEM.clearDBBackLog( )`
Deregister Node	Removes node from database. Only enabled after node has been stopped.

Deploy Expert	Makes Expert (Samples, Probes, or Wrappers) ready for use.
Deploy Manager	Makes default policy manager ready for use.
Node Properties	Displays node property configurations.
Domain Properties	Displays domain property configurations.
System Health	Displays node_health.bsv which monitors CEP server health and performance.

4.4.3.1 Grid Support

M6 provides grid support for all CEP servers. The Grid is a collection of highly available services that allow the user to provide uninterrupted management in the event of server failures. To implement a grid, do the following:

1. On the server running the Domain Server, create a folder named grid under *$AUTOPILOT_HOME*/naming. (*$AUTOPILOT_HOME* is the install directory of AutoPilot M6.)
2. Create a folder for the grid you are defining under naming/grid. (The folder name is user-defined and should accurately represent the collection of services you will be placing in the grid.)
3. Create all the required services on a dummy CEP server.
4. Copy registry.xml to the naming/grid/[*grid_name*].
5. Create a file named server.properties and enter all the required properties to be used by the business views. (For example, required properties could be environment variables.)

   registry.xml – defines shared cluster services

   server.properties – defines properties required by cluster services

To be selected as the primary server by the grid manager, each CEP server must declare a priority. The higher the priority, the more likely it will be selected as the primary server.

6. Start all participating nodes using the following parameters:

   ATPNODE –console –join *[grid_name]* –i *[number]*

   (*<number>* is the priority.)

   You may also provide these parameters through the CEP server's node.properties file. For example, for CEP servers running as Window Services:

   property server.grid=[grid_name]

   property sever.grid_enabled=true

   property server.grid.pri=0

The Grid Manager is always running within the domain server, and it must be highly available. When the domain server is started, grid manager waits for CEP server registrations with a specific grid and then begins the primary election logic based on CEP server priority.

Each grid has only one primary server. Grid manager votes on the primary server based on the CEP server priority when the primary server becomes unavailable for any reason. When the primary server becomes available and joins the grid, it will not become primary until the current primary server is taken offline. The following properties control primary voting logic:

Grace period in ms before electing primary – gives a primary server a chance to rejoin and regain primary status.

server.grid.vote.delay where default is 1500 – delay vote in ms.

Each grid must have its own primary server and should be checked periodically by using:

server.grid.vote.sample where default = 120000.

4.5 Experts

In M6, experts are agents that are knowledgeable about the subjects they are assigned to monitor. Experts are mobile and act as connectors to other M6 experts. They are contained in M6 target managed resources (nodes) and collect system's status information, which is analyzed by assigned policies and business views: they are filtered by M6 managers to be viewed in a business view.
Experts publish facts. Facts are basic true statements about current run-time values, indicating an objects' state in a managed resource.

1. By default, users in the Operator Group have read-only and operational access to expert properties. If you require additional access privileges contact your local M6 Admin.
2. The number and type of built-in experts will vary based on options exercised at installation as well as license limitations.

M6 experts perform the following functions:

- Experts serve as liaisons between managed resource and M6
- Understand events, information, and protocols of managed resources
- Translate unique "languages" of managed systems to M6 facts
- Collect and publish facts about managed resources to all interested subscribers
- Carry out actions requested by users via M6 User Console or M6 Web Console.
- Carry out actions requested by managers or policies.

4.5.1 Built-in Experts

The Application Instrumentation Module (AIM) which is described in this section is a licensed function. When used for data collected by AutoPilot or by one of the licensed experts, this function is an extension of that license. Additional licenses are required when AIM is used to collect data from other products or components. Examples include using the SQL expert to collect data from a database created by another product or using APFACT to send data to the process monitor. M6 includes the following built-in experts:

- Samples
- Probes
- RSS News Feed
- Wrappers

There are many additional built-in experts that are supplied with various plug-ins. See the guide for the appropriate plug-in for additional expert details.

4.5.1.1 Samples

The following sample experts are provided as references and can be used by developers to create their own experts.

- **Counter:** Increments an integer based on a user defined timer.
- **Car:** Demonstrates a simple car simulation.
- The source code is available in `[AUTOPILOT_HOME]\examples` directory and is available when "*SDK Samples*" option is installed.

NOTE A limitation in the 1.x versions of the libsafe library can cause Java 2 SDK tools to fail with the message "Invalid initial heap size: -Xms8m". Please download the most recent release of the libsafe library, version 2.0, from http://www.research.avayalabs.com/project/libsafe/.

4.5.1.2 Probes

M6 has three built-in probes as follows:

- URL Monitor
- Stress Test
- SQL Query Monitor

URL Monitor

The URL Monitor measures the availability and performance of one or more URL addresses such as files, web pages, and web applications.

The URL Monitor accepts one or more URLs, such as http://www.nastel.com and https://www.google.com and can include parameters as appropriate (?id=9999). If required, a proxy can be configured to access the URL.

Based on the sampling rate, the URL Monitor will attempt to connect to each URL configured (as a synthetic transaction). If successful, it will return details about the response packet. If unsuccessful, it will return details about the results and the exception. (See figure below.)

The facts vary based on the response from the expert. Descriptions of the various response facts can be found at the following web pages:

- http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html
- http://en.wikipedia.org/wiki/List_of_HTTP_header_fields

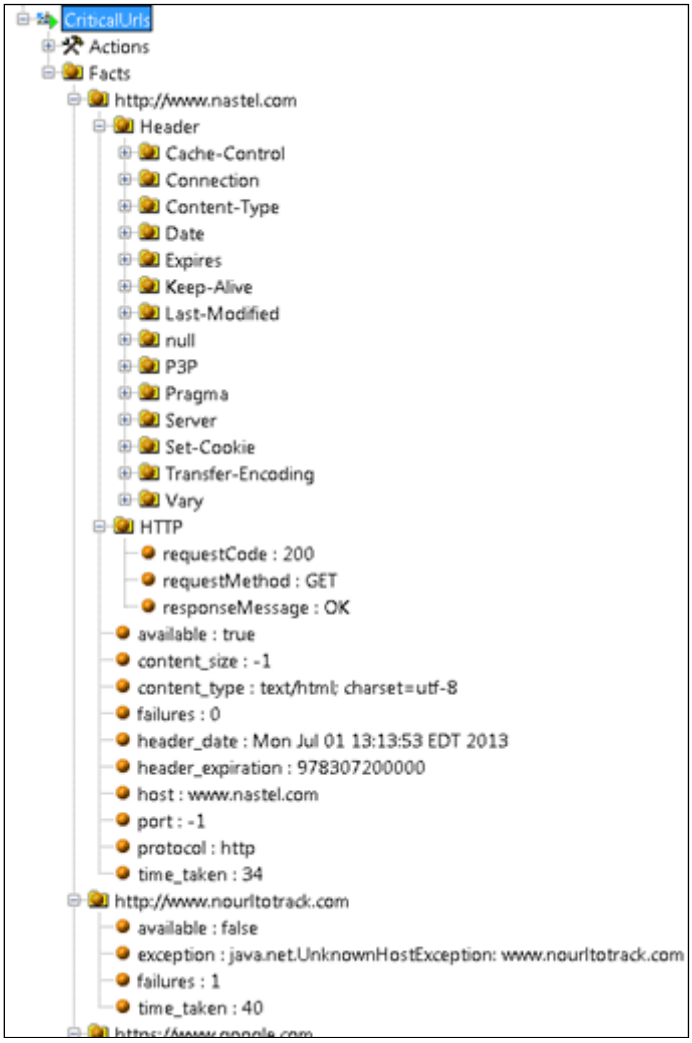


Figure 4-30. URL Monitor Success Data

The user can specify content string(s) that must be present in content by selecting **Load content** and specifying the content sting(s) in the **Search Content** field in the figure below.

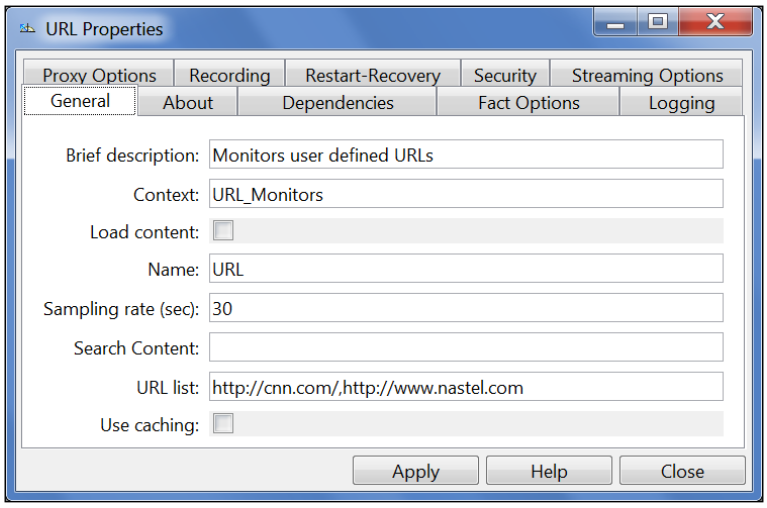


Figure 4-30A. URL Properties – General

Table 4-15A. URL Properties – General	
Property	**Description**
Brief description	A short, user defined, description of the service
Context	A user defined category that will be registered with the domain server. The default is: *URL_Monitors*. Context can be changed as needed.
Load content	Load URL content during the scan (used in conjunction with **Search Content** field).
Name	Name that uniquely identifies the service in the domain
Sampling rate (sec)	Sampling rate in seconds
Search Content	Comma separated list of content to search for. **Load content** field must be selected. One or two facts are created (Figure 4-30B): • **content_match** – true/false • **content_line** – data from url that matched (not present if content_match = false)
URL list	Comma separated list of URLs to be scanned
Use caching	Document cache when sampling URLs (protocol dependent)

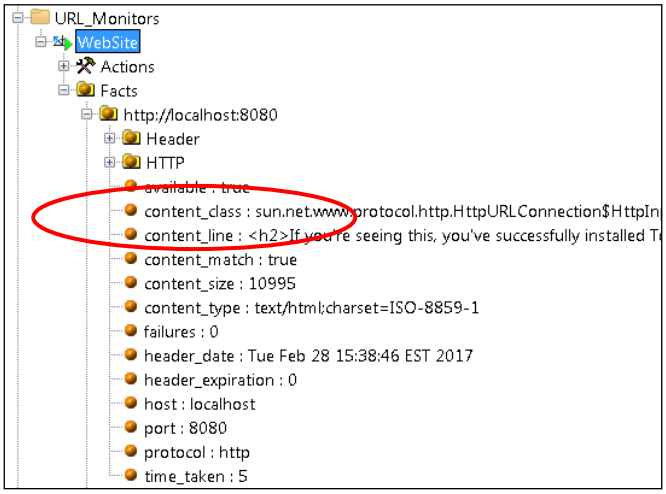


Figure 4-30B. New Facts Created

Stress Test

The stress test allows users to generate hundreds or thousands of facts to stress test CPU and memory performance of a CEP server. This probe can be used to test performance and scalability of a CEP server. Flow control is provided for all socket communications by preventing network delays from degrading CEP server CPU and memory performance using the following properties:

- **server.pipe.processor.limit=20000**. When the maximum number of queued objects for each socket connection reaches its limit, the connection is disabled, and objects are dropped.
- **server.pipe.delivery.flowpct=60**. Setting the value between 0 and 100 temporarily suspends the flow when the usage reaches 100%. The flow will remain suspended until the usage drops below the value specified by the variable.

  Setting value to 0 or below disables this property, which means flow will always stay enabled. Enabling flow control puts a maximum limit on number of outstanding events for each connection and limits memory growth associated with unlimited growth of queued objects.

SQL Query Monitor

The SQL Query Monitor allows users to define an SQL query and publish the results to Auto Pilot. By default, it is configured to sample MySQL STATISTICS table with a sample query. This probe is only available when AIM 6.0.6 or higher is installed.

When specified, the **Sort By Column(s)** field under *SQL Query* tab controls the instance creation. The SQL Query creates a set of facts for each row. The facts represent the row returned. For example: If select a,b,c from db.table where a >100, a set of facts is created for each row that matches a >100 and 3 facts (a, b, and c) if **Sort By Columns** was specified.

To create a SQL query, the expert must be deployed.

1. Right click the desired CEP server to receive the SQL Query.
2. Select **Deploy Expert > Probes > SQL Query Monitor**.

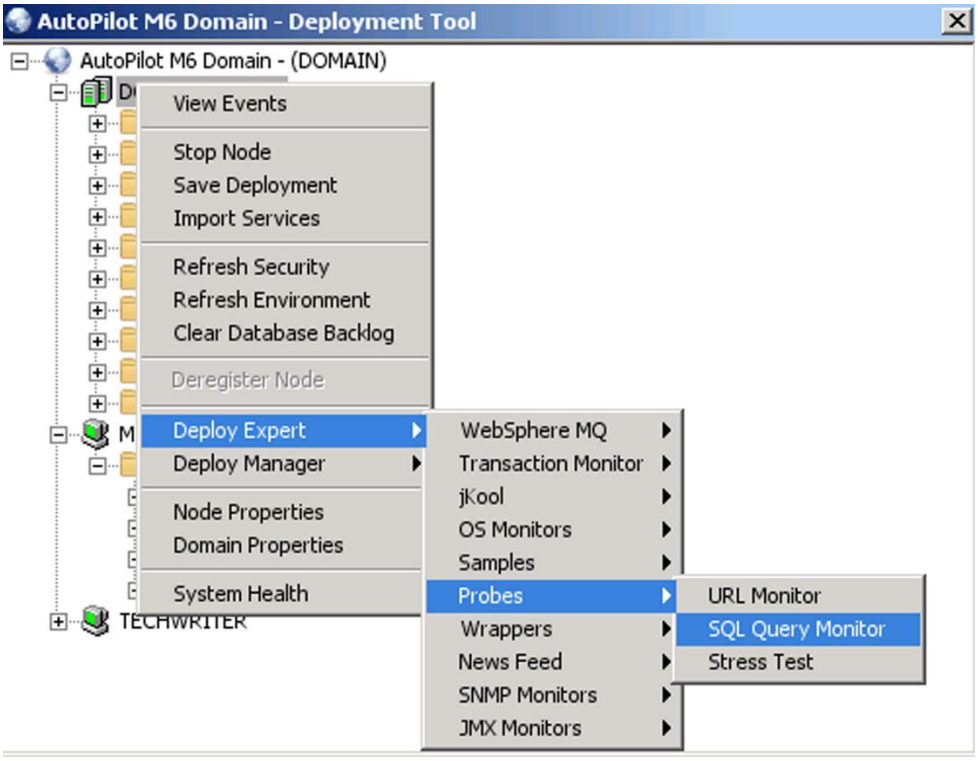


Figure 4-31. SQL Query Deployment

Configure as required for your application. An example for creating a query is shown below.

Creating an SQL Query Example:

After the SQL Query Monitor expert is deployed, do the following:

1. Give your query a name by selecting the *General* tab, if not already selected. The name must be unique within the M6 domain.

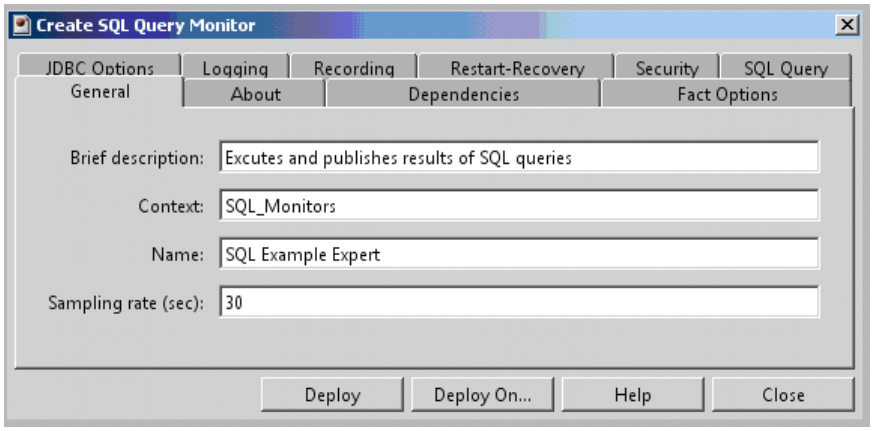


Figure 4-32. SQL Query, General Tab

2. To specify the required database, the credentials to access the database and the required driver, edit JDBC options by selecting the *JDBC Options* tab. This information will vary from database to database and from platform to platform.

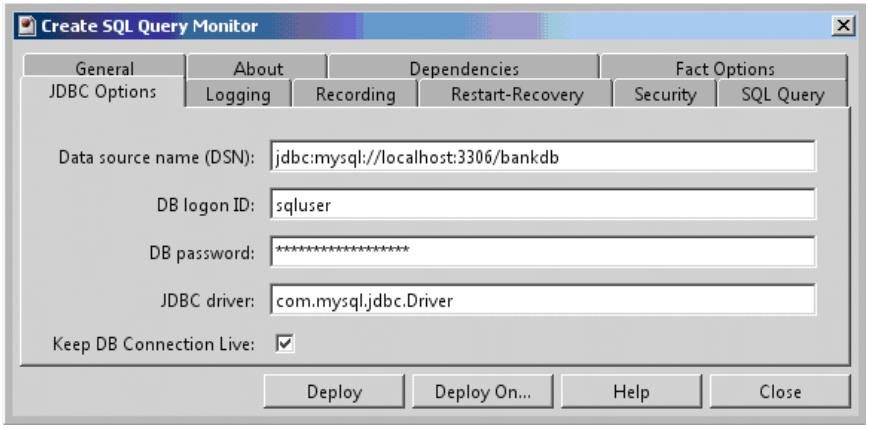


Figure 4-33. SQL Query, JDBC Options Tab

3. To specify the query and structure for the facts, select the *SQL Query* tab.

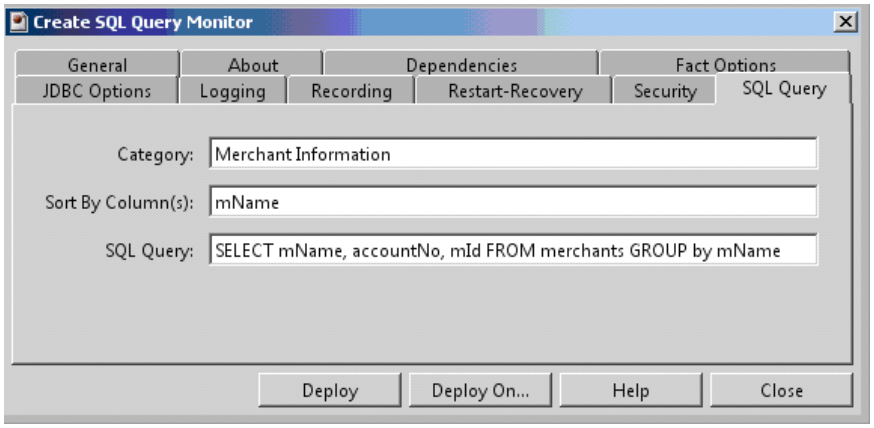


Figure 4-34. SQL Query, SQL Tab

In order to properly create facts, the query should return one row for each set of facts to be created in AutoPilot. In the figure below, the records are grouped and sorted by merchant name. AutoPilot is populated with the facts represented by the query. The facts are created under a category named *Merchant Info* which was specified on the *SQL Query* tab.

The **Sort By Column(s)** field is used to create three sets of facts for each merchant. The three facts selected were created as facts within each merchant.

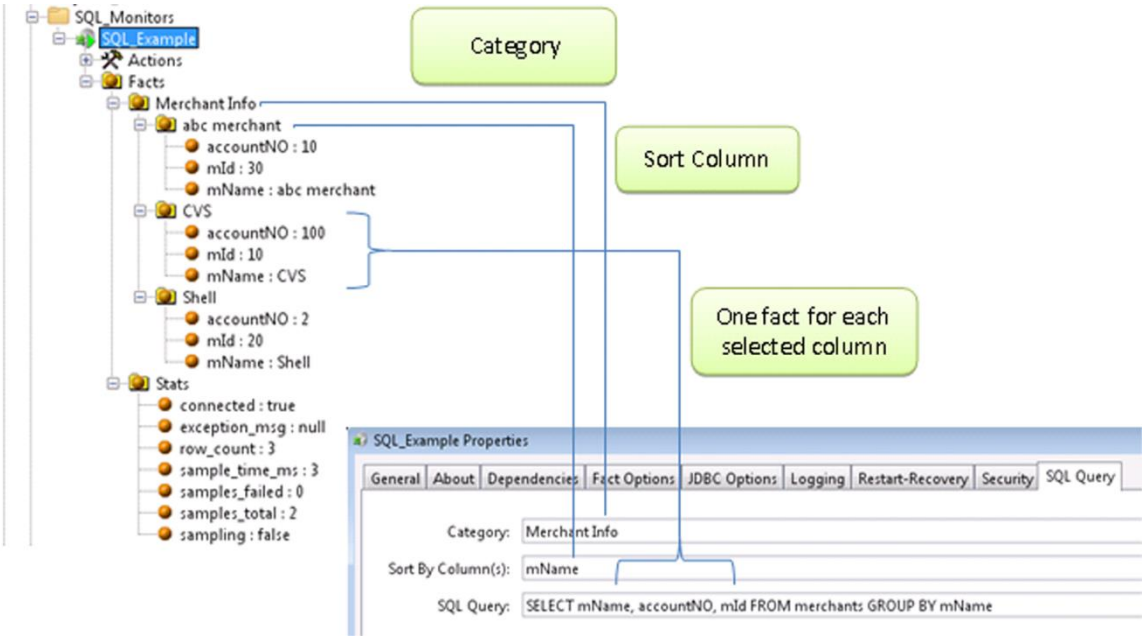


Figure 4-35. SQL Query Facts

4.5.1.3 RSS News Feed

M6 has one built-in News Feed that lets you display current news based on the RSS URLs you selected to be scanned. This option is only available when AIM plug-in is installed.

4.5.1.4 Wrappers

There are two wrapper experts built-in as application templates. The wrappers are defined in the following paragraph.

- **Process Wrapper:** Allows the user to associate a process or a script that would be started by M6 and be monitored by the process wrapper. Each wrapper may be associated with only one process or log. Process Wrapper is also used in conjunction with Fact Publishing utility **apfact**, which allows users to instrument user applications and publish metrics directly to AutoPilot M6 via a process wrapper.
- **File Monitor:** Separates log metrics from folder metrics. Lets you monitor the contents of a file in a specified directory. All folder-related facts are published under Facts\Folders. Also, lets you to monitor application error logs and publish logs to M6 as facts and events. All log-related facts are published under Facts\Logs. There is also support for multiple files and event profiles within a single file monitor instance.

4.5.2 Deploying Experts

Experts can be deployed on CEP servers (including the domain server) within the M6 Network. M6 is supplied with built-in experts that can be used immediately after installation. In addition, several plug-ins are available which offer experts that are unique to each managed application.

This section provides you with instructions for the deployment of experts within the M6 network. The actual deployment of built-in experts should take just a few seconds in most cases. The following section is a typical example for deploying experts.

4.5.3 Deploying Process Wrapper

To deploy and configure an instance of a process wrapper:

1. Right click the desired CEP server.
2. Select **Deploy Expert > Wrappers > Process Wrapper**
3. Configure Process Wrapper as required for your application.

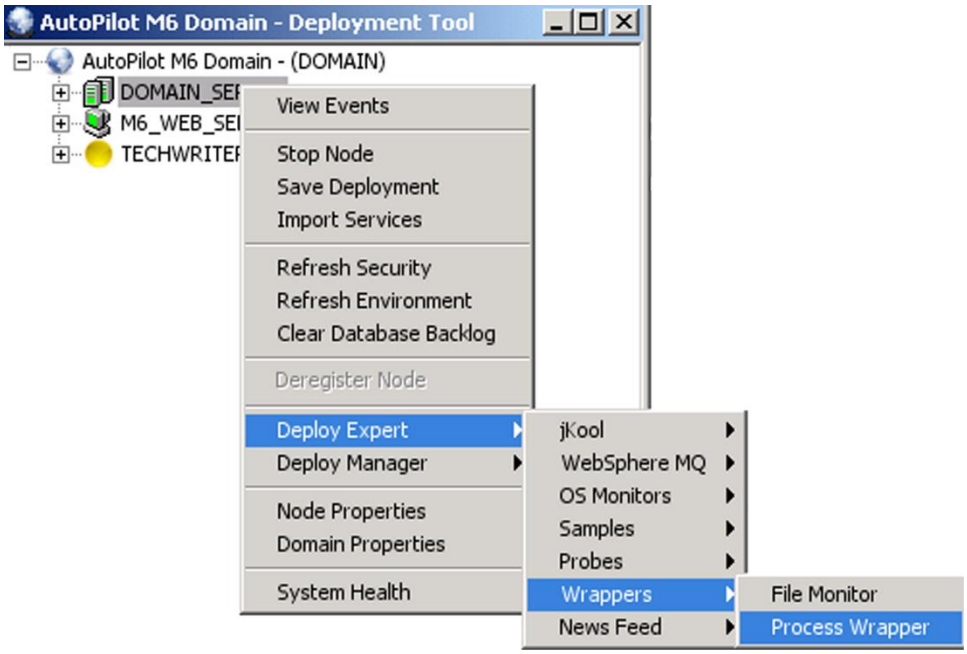


Figure 4-36. Deploying Process Wrapper Experts

> **NOTE** The system assigned unique name is *Service* and timestamp in milliseconds. The name can be user defined but has to be unique within the M6 domain.

4. Click **Deploy** button to deploy the expert on the CEP server. The *Create Process Wrapper* screen is displayed. The name of the expert being deployed is repeated along with the node where it will be deployed.

   OR

   Click **Deploy On** to deploy on multiple CEP servers. The *Deploy Across Network* screen is displayed. Select the nodes to receive the expert. You can use **Ctrl/click** to select multiple nodes in the domain. Click Deploy to deploy the expert on the selected nodes. When you have deployed, a check mark indicates which nodes have the expert deployed.
5. Right-click on CEP server(s) that had the expert deployed on and click **Save Deployment**.

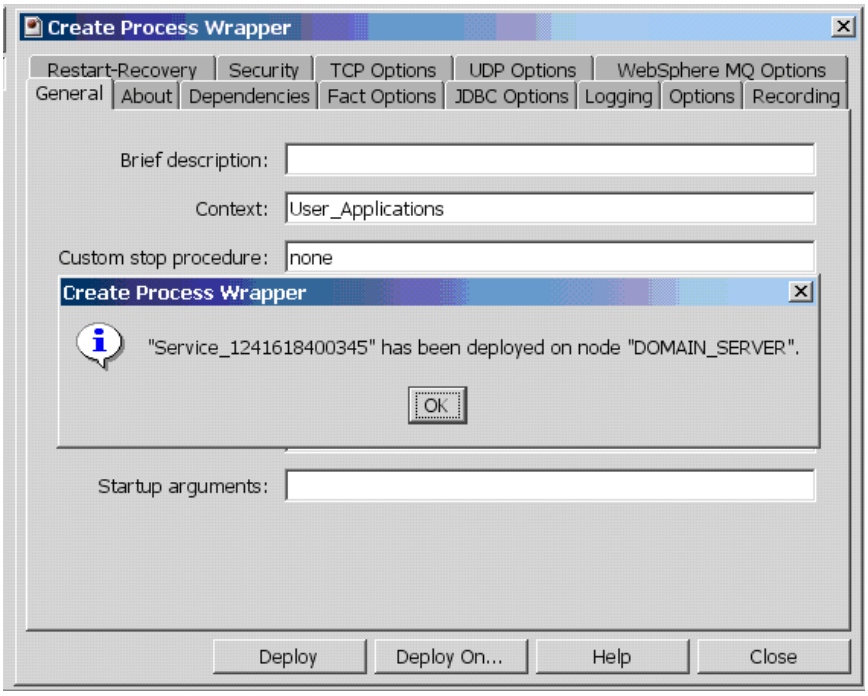


Figure 4-37. Deploying Wrappers

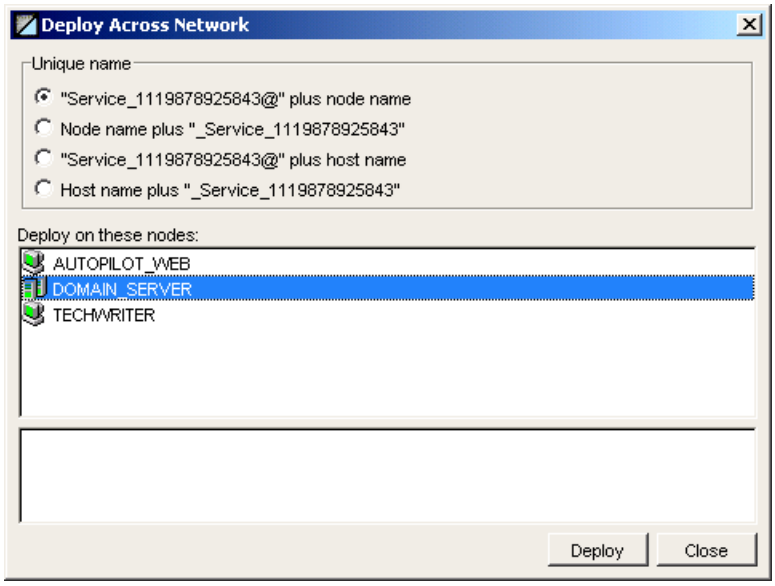


Figure 4-38. Deploy Wrapper on Multiple Nodes

4.5.4 Configuring Process Wrapper

A Process Wrapper can be used to initiate applications, accept facts from *apfact* and WebSphere MQ queues and record facts into a JDBC database.

Configure UDP Options section if process wrapper is used in conjunction with *apfact* (see C.4 APFACT – Fact Publisher).

A Process wrapper can also accept facts from a WebSphere MQ queue. Configure WebSphere MQ Options if facts are published into a WebSphere MQ queue. Applications that record facts into a WebSphere MQ queue must put messages with MQSTR format and message body should contain facts as described in C.4 APFACT – Fact Publisher.

NOTE

Process Wrapper uses WebSphere MQ Java client to communicate with the queue manager. Make sure that your queue manager is configured to accept WebSphere MQ clients – queue manager must have a channel listener as well as a server connection channel defined. Please refer to WebSphere MQ Administration Guide for more details.

Configure the process wrapper to support your needs using the table below. Mandatory or minimum entries are in **red**. There are 15 configuration screens for the process wrapper, not all are required, but all are defined in the table.

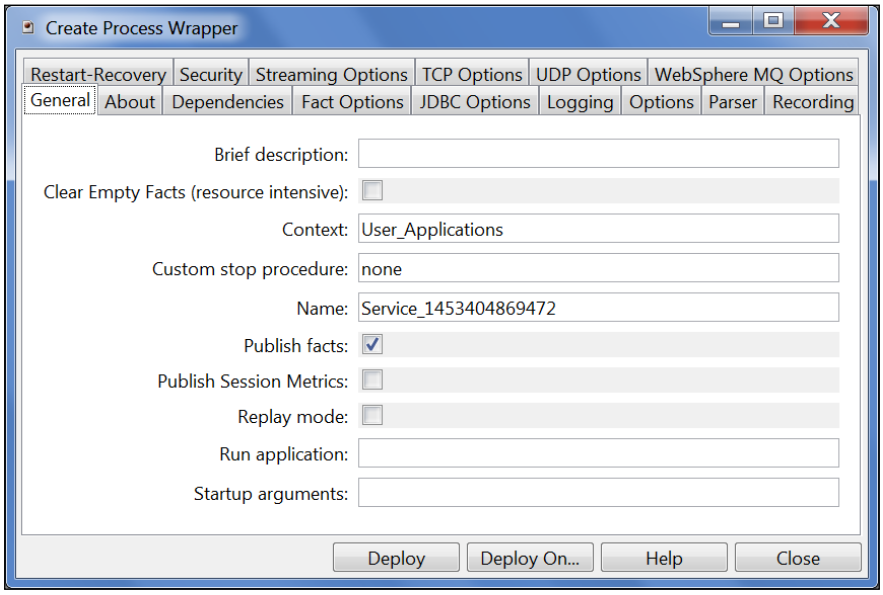


Figure 4-39. Process Wrapper General Properties

Table 4-16. Process Wrapper Expert Configuration	
Property	**Description**
General	
Brief Description	A short, user defined, description of the service
Clear Empty Facts	When enabled, clears facts with empty or null values.
Context	A user defined category that will be registered with the domain server. The default is: *User_Applications.* Context can be changed as needed. M6 will define new category.
Custom stop procedure	Application or script that gracefully shuts down started applications. The default is: *none*. The application will be terminated using OS specific stop procedure.
Name	Name that uniquely identifies the service in the domain. The default name system assigned with the word service and 12 random digits (example: service_123456789012)
Publish facts	Enable/Disable ☑/☐ to publish received facts locally.
Publish Session Metrics	When enabled, publishes session performance metrics.
Replay Mode	Enable/Disable ☑/☐ to emulate fact source and replay all received facts.
Run Application	Fully qualified name of application/script/process to run (example: /usr/bin/tar)
Startup Arguments	Start–up parameters that are passed to the application (example: cvf mytar/opt/nastel)
About	
Package title	Implementation title of the source package.
Package vendor	Name of implementation vendor.

Table 4-16. Process Wrapper Expert Configuration	
Property	**Description**
Package version	Package version as assigned by the vendor.
Dependencies	
Platform Dependencies	Comma separated list of operating system platforms this expert is dependent on.
Service Dependencies	Comma separated list of services this expert is dependent on.
Fact Options	
Exclude Expire Filter (regexp)	Do not expire facts that match specified regular expression.
Exclude Fact Filters	Comma separated list of fact paths to exclude during publishing.
Expire facts (ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size *	Automatically maintains specified number of samples for each published fact in memory
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Fact service alias	Override fact service prefix for all published facts. Facts will be appearing under specified service name.
Include Expire Filter (regexp)	Facts that match the specified regular expression are expired
Include Fact Filters	Comma separated list of fact paths to include during publishing.
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to **Fact History Time** or **Fact History Size** which ever limit is reached first. If disabled newer history samples replace older on a rolling basis.
JDBC Options	
Data source name (DSN)	Logical data source name that points to the physical database.
DB logon ID	Login ID to access database if required.
DB password	Password to access database if required.
DB table	Name of the physical database table (example: CPU_TABLE)
Enable DB logging	Enable/Disable ☑/☐ database logging.
JDBC driver	Class name of the JDBC driver.
User table columns	(LogTime) VALUES (?)
Logging	
Audit	Enable/Disable ☑/☐ service audit trace.
Log name	Log name associated with the service.
Log service activity	Enable/Disable ☑/☐ service activity trace. (Should only be enabled for troubleshooting purposes to eliminate significant performance degradation.)
Log size (bytes)	Log size in bytes. Real log size is the maximum value of server.log.size and logsize.
Options	
Auto-restart delay (ms)	Delay before restarting stopped applications.
Auto-restart	Enable/Disable ☑/☐ to specify whether application should be restarted when stopped.
Environmental variables	Comma separated list of environment variables required for the application (example: PATH=/opt.nastel,TEMP=/temp)

Table 4-16. Process Wrapper Expert Configuration	
Property	**Description**
Is daemon	Enable/Disable ☑/☐ to specify whether application is long running (daemon) or terminates after execution.
Redirect output	Enable/Disable ☑/☐ to redirect program output to private event log.
Parser	
Fact name group	Obtain fact name part from the specified group in the fact pattern expression.
Fact pattern	Fact pattern expression used for parsing incoming facts.
Fact value group	Obtain value part from the specified group in the fact pattern expression.
Quote character	Quote character used to parse incoming facts; default is " (double quotation mark)
Space characters	Space characters used to parse incoming tokens (default is , (comma) and ; (semicolon)
Recording	
Anomaly Deviation Limit	Number of standard deviations above or below the mean.
Exclude Filter (regexp)	A regular expression filter to exclude certain facts from being written to the database. Facts have the format `expert\class\instance\leaf=value` such as in the example `Servers\Linux\Serv7\processes=40`.
Fact Anomaly Frequency	Frequency at which anomalies are checked and recorded.
Fact State Frequency	If **Record Fact State** is enabled, the value entered here specifies how often the Fact State is updated.
Fact Summary Frequency	If **Record Fact Summary** is enabled, used to write an intermediate summary record every Xth update to the fact during the Summary Interval. This is done to avoid waiting the full Summary Interval for a summary record to appear in the summary table.
Include Filter (regexp)	A regular expression filter to include certain facts being written to the database. Same format as described for the exclude filter.
Record Fact Anomalies	Enable/disable fact anomaly recording for this service.
Record Fact History	If enabled, records every fact change into the History database. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.
Record Fact State	If enabled, records the last value published (current state) into the state database and restores that value when the CEP Server is stopped and restarted. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.
Record Fact Summary	If enabled, records summary record at the interval designated in the **Summary Interval (ms)** field into the Summary database. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.
Storage for Anomalies	SQL table where anomalies are recorded.
Storage for History	Database table where the Fact History data is stored.
Storage for State	Database table where the Fact State data is stored.
Storage for Summary	Database table where the Fact Summary data is stored.
Summary Interval (ms)	If **Record Fact Summary** is enabled, designates in milliseconds, how often the Fact Summary data is written.

Table 4-16. Process Wrapper Expert Configuration		
Property	**Description**	
Restart-Recovery		
Automatic start	Enable/Disable ☑/☐ automatic restart.	
Save in registry	Enable/Disable ☑/☐ to save persistent service in registry.	
Synchronous Control	Enable/Disable ☑/☐ to initiate synchronous service.	
Security		
Inherit permissions from owner	Enable/Disable ☑/☐ inheriting of permission from owner's permission masks.	
Owner	User that owns the object.	
Permissions	Permissions for users of the same group and others. Disable/Enable as required.	
	Group:	**Other Users:**
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, and disable.	Others may execute control actions such as start, stop, and disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.
Streaming Options (Refer to Section 4.16.)		
Application name	Sets application name	
Data center name	Sets data center name	
Exclude filter (regexp)	Ignore facts that match specified regular expression	
Include filter (regexp)	Log facts that match specified regular expression	
Location	Sets server location	
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework)	
Streaming configuration	Streaming configuration block name	
TCP Options		
Accept TCP facts	Enable/Disable ☑/☐ to accept facts published to the specified TCP port.	
Option SO_TIMEOUT	TCP option – read timeout	
TCP port	Unique port on which this service will listen on for incoming facts	
UDP Options		
Accept UDP facts	Enable/Disable ☑/☐ to accept facts published to the specified UDP port.	
UDP port	Unique port on which this service will listen on for incoming facts.	
WebSphere MQ Options		
Accept truncated msg	Accept and truncate incoming message if exceeds maximum message size.	
Facts from WebSphere MQ	Enable/Disable ☑/☐ reading facts from WebSphere MQ.	

Table 4-16. Process Wrapper Expert Configuration	
Property	**Description**
Log message context	Enable/Disable ☑/☐ to log WebSphere MQ descriptor such as type and persistence.
Password	Password to be used when connecting to the queue manager.
Queue manager host	Name of the host where the queue manager is defined.
Queue manager name	Name of the local queue manager.
Queue manager port	Channel listener TCP port for the queue manager.
Queue name	Name of the queue where the facts are being published.
Server connection channel	Name of the SVR CONN channel on the queue manager.
SSL certificate store	SSL certificate store location
SSL cipher suite	Cipher suite that matches the CipherSpec of the channel
SSL enable	Enable/disable ☑/☐ communication for channel communication.
User ID	User ID to be used when connecting to the queue manager.

*** Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact history is only maintained up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to section 4.8.3.4 for more detailed information on fact volatility.)

4.5.4.1 Recording Facts

To enable fact recording for this service, click the *Recording* tab (Figure 4-40). You can filter the facts you want to record by using the exclude/include filters.

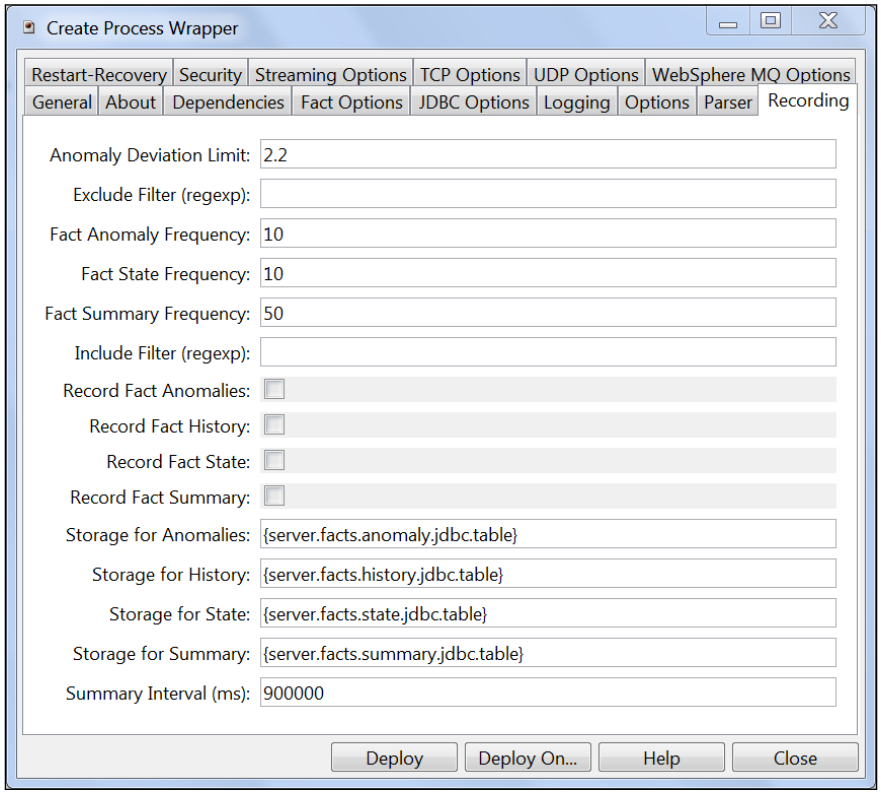


Figure 4-40. Recording Characteristics

Refer to Table 4-16 for an explanation of each *Recording* property. In addition, to enable fact recording for state, history, or summary, you must complete the following steps to define the database tables and set the AutoPilot options. The following steps are for MySQL.

1. Create a database schema, for example `history_logging`.
2. Use the sample SQL in *[AUTOPILOT_HOME]*`/sql-scripts/system/core-mysql.sql` to create the tables required.
3. Copy the sample properties from *[AUTOPILOT_HOME]*`/sql-scripts/system/core-sql.properties` into the require properties file, for example *[AUTOPILOT_HOME]*`/localhost/node.properties`. (See Section 5.3, Server Runtime – PROPERTY FILES for more information.)
   a. Change the following properties to match your installation:
      - property `server.service.facts.logging=true` – must be added to activate fact recording (for testing can be specified as record when starting the server on the command line)
      - property `server.facts.jdbc.driver=com.mysql.jdbc.Driver` – specifies the name of the JDBC driver
      - property `server.facts.url=jdbc:mysql://localhost:3306/history_logging` – specifies the url (host and port) and schema name for the MySQL Database
      - property `server.facts.jdbc.user=user` – specifies the user to use to connect
      - property `server.facts.jdbc.password=pswd` – specifies the password to use to connect
      - property `server.facts.summary.jdbc.table=ft_summary` – default value for the summary table (can be overridden as shown above)
      - property `server.facts.state.jdbc.table=ft_state` – default value for the state table (can be overridden as shown above)
      - property `server.facts.history.jdbc.table=ft_history` – default value for the history table (can be overridden as shown above)
   b. Review the following and change only as needed:
      - property `server.facts.summary.update.frequency=50` – frequency which summary records are updated (a lower value increases the potential I/O activity, a higher value increases the time before recent data is written to summary table)
      - property `server.facts.state.update.frequency=10` – default frequency at which state records are synced to the database. A value of 1 writes the state information to the database every time the fact is published and should be used for low volume facts.
      - property `server.facts.jdbc.batch.size=20` – JDBC batch size used for writing fact history
      - property `server.facts.jdbc.retry=30000` – JDBC connection retry in ms. in case of a SQL failure

4.5.4.2 Process Wrapper JDBC Configuration

This section describes the procedure for enabling database fact logging with an M6 Process Wrapper. There are several important steps that must be taken to ensure proper database insertion. These steps

include determining the source of the data, configuring the insertion format, and configuring the database table correctly.

> **NOTE** The functionality provided by using JDBC options can be done much simply by using the *Recording* tab and selecting the facts to record.

Determining Data Source

There are several ways that an M6 Process Wrapper can receive data and publish as facts. The first method is using TCP protocol, the second UDP and the last is from an MQ queue.

TCP

TCP is a reliable data connection to ensure facts will be published, but there is a slight performance hit due to the extra networking overhead required.

1. Click *TCP Options* tab, to enable an M6 Process Wrapper to receive TCP data.
2. Check *Accept TCP Facts* checkbox ☑ , and then enter a port. The port will also have to be specified in the application sending the fact data.

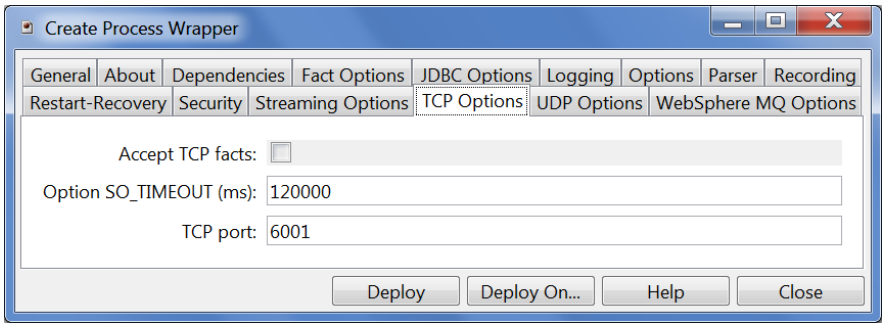


Figure 4-41. TCP Options for JDBC Configuration

UDP

UDP is an unreliable data connection used mainly for speed and application-to-application decoupling. While this is the fastest protocol with the least amount of network overhead, there is potential that data could be lost since there is no acknowledgment/hand shaking between the sender and receiver.

1. Click *UDP Options* tab to enable an M6 process wrapper to receive UDP data.
2. Check *Accept UDP Facts* ☑ checkbox and then enter a port. The port will also have to be specified in the application sending the fact data.

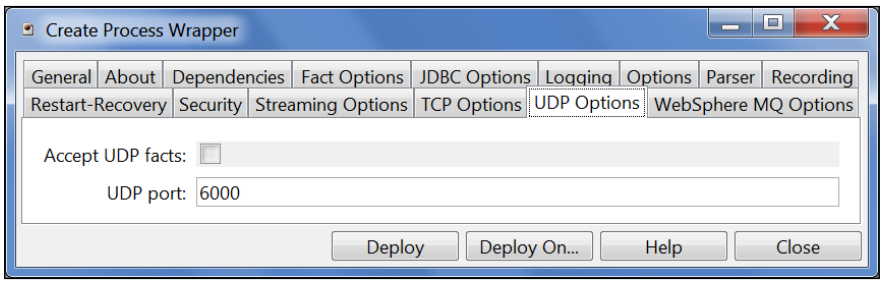


Figure 4-42. UDP Options for JBDC Configuration

WebSphere MQ

MQ is an asynchronous assured delivery messaging platform. This protocol is used when messages cannot be lost and must be decoupled from publishing applications.

1. Click *WebSphere MQ Options* tab to enable an M6 Process Wrapper to receive MQ message data.
2. Check *Facts from WebSphere MQ* checkbox.
3. Configure connection to the Queue Manager that will receive the message data from the source.
4. Enter Queue Manager Hostname.
5. Enter Queue Manager name (case sensitive).
6. Enter listener port and the Queue name where the messages will be taken.
7. Enter Server connection channel name.
8. Enter user ID.
9. Enter password if specified in the channel definition.

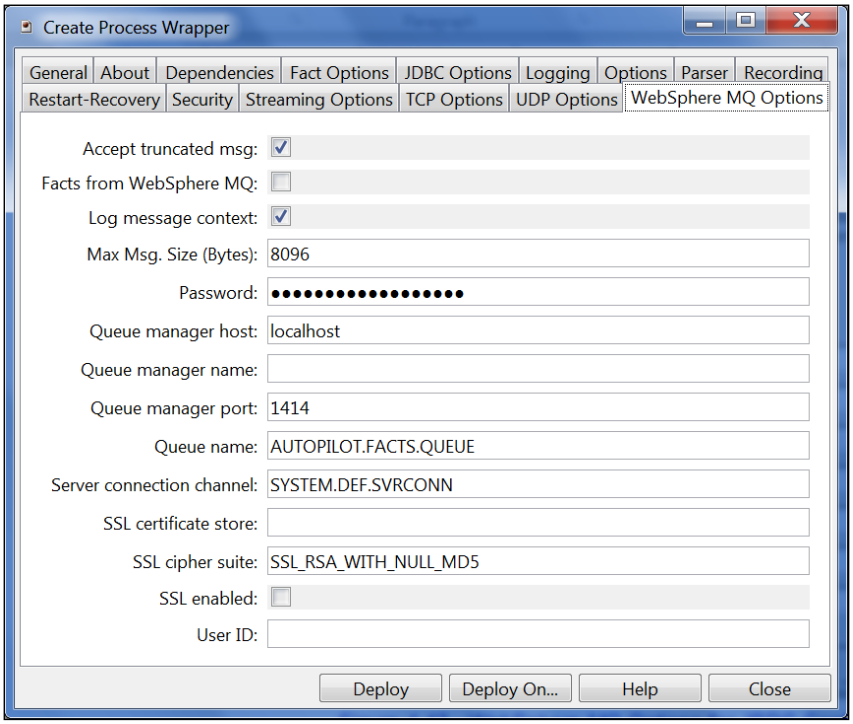


Figure 4-43. WebSphere MQ Options for JBDC Configuration

The *Log Message Context* will log additional information about each message that is received from the MQ Queue. This data captured from each message is:

messageType, persistence, priority, characterSet, encoding, putApplicationType, replyToQueueManagerName, replyToQueueName, applicationIdData, applicationOriginData, TotalMessageLength, ptime, messageId, correlationId, accountingToken

In the data field of the MQ messages being placed on the MQ Queue by the sending application, data must be paired by **fact_name=value** and comma separated. For example:

```
FLOW=NASHUB_FLOW, EVENT=FLOWTIME, TOTALTIME_TIMEMETRIC = 1308,
NASTIME_TIMEMETRIC = 1128, TOTALNASTIME_TIMEMETRIC = 1128, NAST_UID = 39196
WWS0330300007001NN1245448920050303, MSGID =
414d5120574249514130312020202020421e8db0201836b8, REINPUTS = 0, PUTTIME =
17:45:48.350, NAS_TRANID = CISI50
```


The facts will be overwritten every time a new message is written to the queue and retrieved by the Process Wrapper. These facts can be used to define business rules with thresholds. They could also be correlated together or with other metrics published within other parts of M6.

Configuring JDBC Options:

Once the data is obtained and written into M6, this same data can be written to a database. The reasons for doing this may be to retain critical corporate information for historical purposes, capacity planning, or compliance with industry business standards such as Sarbanes-Oxley.

1. Click *JDBC Options* tab to enable an M6 process wrapper to write fact data to a database.
2. The Data Source Name (DSN) and the JDBC Driver will be the following format:

Table 4-17. Process Wrapper JDBC Options Configuration

Database	JDBC Driver	Data Source Name (DSN) format
ODBC Compliant	`sun.jdbc.odbc.JdbcOdbcDriver`	`jdbc:odbc:dsn_name`
MS SQL Server (lib/jtds.jar)	`net.sourceforge.jtds.jdbc.Driver`	`jdbc:jtds:sqlserver://host:port/dbname`
Sybase (lib/jtds.jar)	`net.sourceforge.jtds.jdbc.Driver`	`jdbc:jtds:sybase://host:port/dbname`
Oracle (lib/ojdbc14.jar)	`oracle.jdbc.driver.OracleDriver`	`jdbc:oracle:thin:@hostname:port:sid`
DB2 UDB (lib/db2java.zip, lib/db2jcc.jar)	`COM.ibm.db2.jdbc.net.DB2Driver`	`jdbc:db2://hostname:port/dbname`
Hypersonic SQL	`org.hsql.jdbcDriver`	`jdbc:hypersonicSQL:hsql://host:port/dbname`
Informix	`com.Informix.jdbc.lfxDriver`	`jdbc:informix-sqli://host:port/dbname`
MYSQL	`com.mysql.jdbc.Driver`	`jdbc:mysql://host:port/dbname`

3. Enter user ID and password.
4. Enter name of the database table.
5. Check *Enable DB Logging* ☑ checkbox.
6. Configure the connection to the database that will receive the data.
7. Set up the *User table columns* properties: This is the most critical part to ensure proper database table insertions.
   - You must know the ***order*** that the fact data is being published into the M6 process wrapper. This is not initially intuitive because once the facts are received; they are viewed through the M6 User Console in alphabetical order. To determine the order facts are received, an examination of the sending source must be done.
   - In the case of UDP or TCP, the script or application will have to be viewed to make the determination of the order of publishing. It should be a simple line by line or loop.

- In the case of MQ, the MQ application can be examined or a simpler approach would be to look at one of the messages using M6 for WMQ Message Explorer. Once the order is determined, the *User table columns* can be defined correctly.
- If the order is not accounted for, data could be inserted into the wrong columns in the database.

The format for this field follows this example:

```
(LogTime,FLOW,EVENT,TOTALTIME_TIMEMETRIC,NASTIME_TIMEMETRIC,
TOTALNASTIME_TIMEMETRIC, NAST_UID,MSGID,REINPUTS,PUTTIME,NAS_TRANID)
VALUES(?,?,?,?,?,?,?,?,?,?,?)
```

Note that:

- LogTime is a pre-requisite for any configuration and is **always** the first field in the user table columns string. This field is defined by default when a Process Wrapper is created.
- After LogTime, insert all the columns in the correct order as they are published. Make sure the syntax is correct. Characters are case sensitive. The entire list is comma delimited and ended with a parenthesis.
- Next, VALUES are defined. This is simply used by the application to know how many columns it has to publish. The total number of question marks should match the total number of columns to be published. Simply count each one, including LogTime, and place the correct number of question marks in the VALUE field.

The following figure is an example of a configured Process Wrapper to publish fact data to a database:

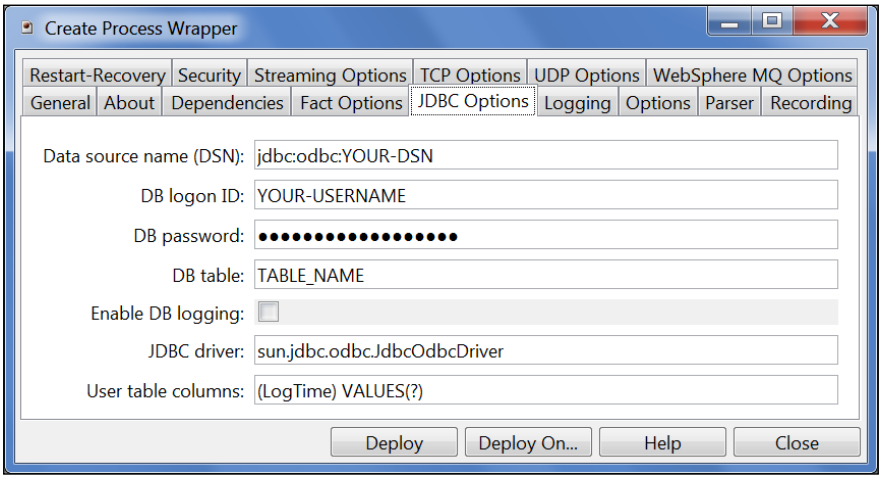


Figure 4-44. User table columns of JBDC Process Wrapper

Configuring the Database:

The final step to ensure proper database logging is to correctly define the database objects. In this case, this means the database table and columns.

- The database can be given a meaningful name, but it is user defined.
- Table column names should be defined the same way as they are in the *User table columns* in the JDBC Options configuration.
- To determine what data type each column should be, the M6 Database Tool can be used. To use Database Tool, make sure your M6 User Console is open and click the Tools menu and choose Database Tool. Once opened, it will look like this example:

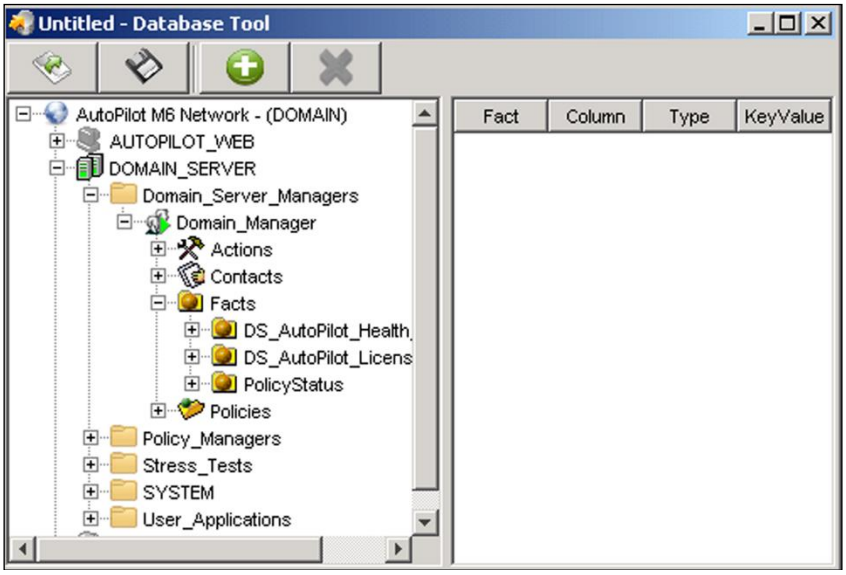


Figure 4-45. AutoPilot M6 User Console Database Tool

1. Navigate to the facts being published by the M6 process wrapper:
2. Add each of the facts to the scheme template by selecting the fact and clicking the  icon in the toolbar. When the fact is added the data type of each fact will be displayed in the *Type* column.

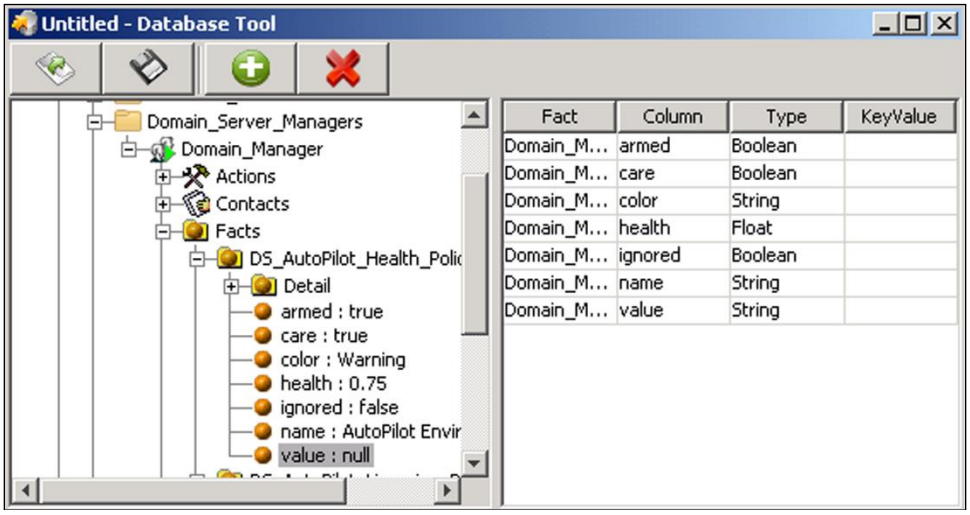


Figure 4-46. Adding Published Facts to Database Scheme

3. From this view, define your database data types for each column. Ensure the database is available and the user ID has the correct rights to insert to the database.

Verification:

Once the configuration is complete from the publishing side, M6 side and database side, start the process wrapper and observe if facts are being published. Depending on the application sending the facts, they may only come at specified intervals. Once facts are populated, we know that data is being received correctly. In turn, logging to database should happen as soon as the first batch of facts is received. To verify the data is going into the database, the user can use a DB admin tool or SQL query to view/edit table contents. If there are no entries in the database, then the user should check the M6 log files. Mainly, right click the Process Wrapper and select *View Events*. If there are connection or insertion problems, they will be reported in this log file. Typically, connection problems are caused by syntax errors setting up the JDBC *User table columns* and/or the database column name/data type definitions.

4.5.5 Deploying File Monitors

To deploy and configure an instance of a file monitor:

1. Right click desired CEP server.
2. Select **Deploy Expert > Wrappers > File Monitor**
3. Configure file monitor as required for your application.

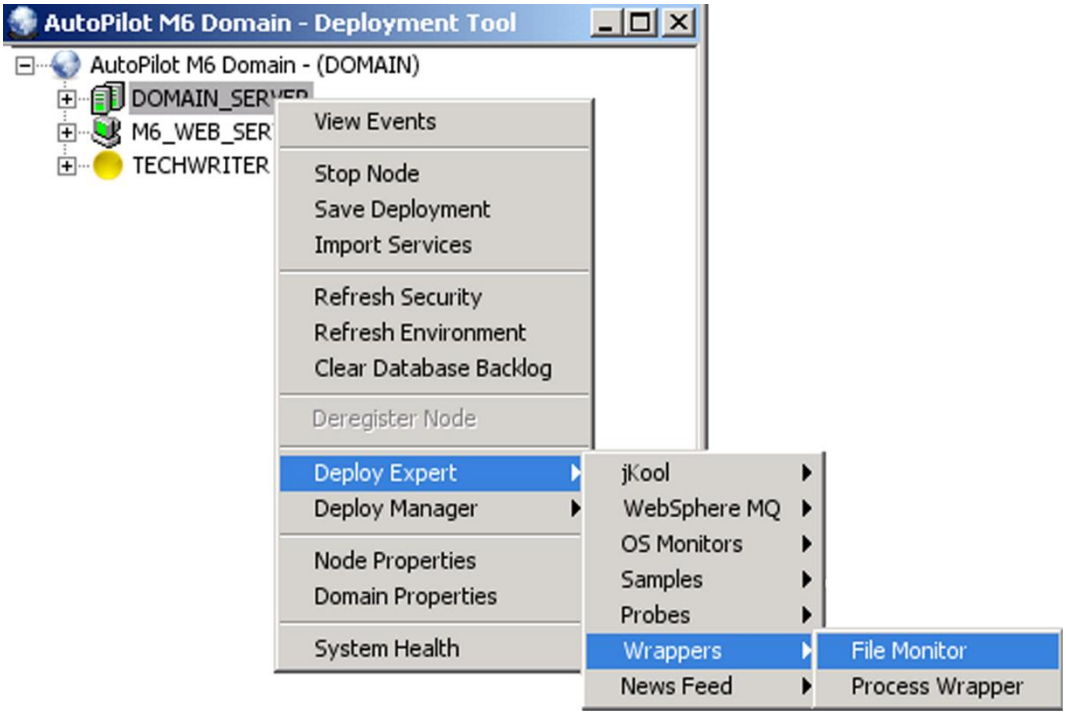


Figure 4-47. Deploying File Monitors

> **NOTE** The system assigned unique name is *Service* and timestamp in milliseconds. The name can be user defined but has to be unique within the M6 domain.

4. Click **Deploy** to deploy the expert on the CEP server. The *Create File Monitor* screen is displayed. The name of the expert being deployed is repeated along with the node where it will be deployed.

   OR

   Click **Deploy On** to deploy on multiple CEP servers. The *Deploy Across Network* screen is displayed. Select the nodes to receive the expert. You can use **Ctrl/click** to select multiple nodes in the domain. Click **Deploy** to deploy the expert on the selected nodes. When deployed a check mark indicates which nodes have the expert deployed.

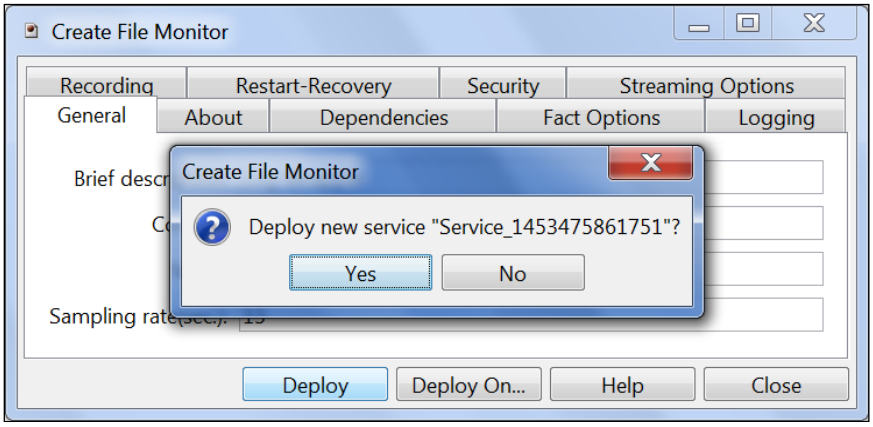


Figure 4-48. Deploying File Monitor on Individual Node

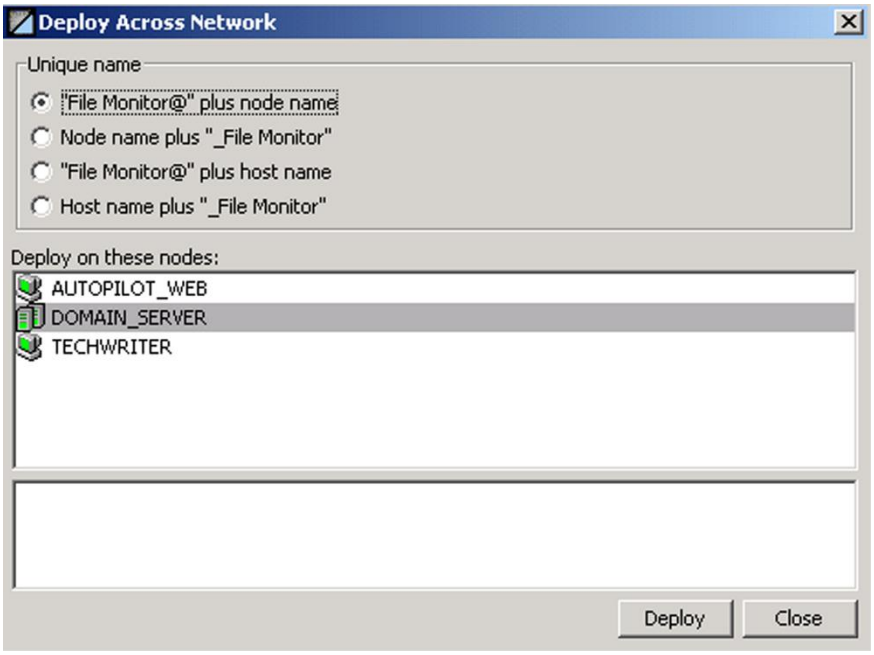


Figure 4-49. Deploying File Monitor on Multiple Nodes

4.5.5.1 Configuring File Monitors

Configure the File Monitor to support your needs using the table below. Mandatory or minimum entries are in red. There are eight configuration screens for the File Monitor, not all are required, but all are defined in the table.

Table 4-18. File Monitor Expert Configuration

Property	Description
General	
Brief Description	A short, user defined, description of the service.
Context	A user defined category that will be registered with the domain server. The default is: *User_Applications.* The context can be changed as needed. AutoPilot M6 will define the new category.
Name	Name that uniquely identifies the service in the domain. The default name system assigned with the word service and 13 random digits (example: Service_1234567890123)
Sampling Rate (sec)	How many times per second the fact is sampled.

Table 4-18. File Monitor Expert Configuration	
Property	**Description**
About	
Package Title	Implementation title of the source package.
Package vendor	Name of implementation vendor.
Package version	Package version as assigned by the vendor.
Dependencies	
Platform Dependencies	Comma separated list of operating system platforms this expert is dependent on.
Service Dependencies	Comma separated list of services this expert is dependent on.
Fact Options	
Exclude Expire Filter (regexp)	Do not expire facts that match the regexp
Exclude Fact Filters	Comma separated list of fact paths to exclude during publishing.
Expire facts (ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size*	Automatically maintains the specified number of samples for each published fact in memory.
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Fact service alias	Override fact service prefix for all published facts. Facts will appear under specified service name.
Include Expire Filter (regexp)	Expire facts that match the regexp.
Include Fact Filters	Comma separated list of fact paths to include during publishing.
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to **Fact History Time** or **Fact History Size** which ever limit is reached first. If disabled newer history samples replace older on a rolling basis.
Logging	
Audit	Enable/Disable ☑/☐ service audit trace.
Log name	Log name associated with the service.
Log service activity	Enable/Disable ☑/☐ service activity trace.
Log size (bytes)	Log size in bytes. Real log size is maximum value of server.log.size and logsize.
Recording	
Anomaly Deviation Limit	Number of standard deviations above or below the mean.
Exclude Filter (regexp)	A regular expression filter to exclude certain facts from being written to the database. Facts have the format `expert\class\instance\leaf=value` such as in the example `Servers\Linux\Serv7\processes=40`.
Fact Anomaly Frequency	Frequency at which anomalies are checked and recorded.
Fact State Frequency	If Record Fact State is enabled, the value entered here specifies how often the Fact State is updated.
Fact Summary Frequency	If **Record Fact Summary** is enabled, used to write an intermediate summary record every Xth update to the fact during the Summary Interval. This is done to avoid waiting the full Summary Interval for a summary record to appear in the summary table.

Table 4-18. File Monitor Expert Configuration		
Property	**Description**	
Include Filter (regexp)	A regular expression filter to include certain facts being written to the database. Same format as described for the exclude filter.	
Record Fact Anomalies	Enable/disable fact anomaly recording for this service.	
Record Fact History	If enabled, records every fact change into the History database. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.	
Record Fact State	If enabled, records the last value published (current state) into the state database and restores that value when the CEP Server is stopped and restarted. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.	
Record Fact Summary	If enabled, records summary record at the interval designated in the **Summary Interval (ms)** field into the Summary database. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.	
Storage for Anomalies	SQL table where all anomalies are stored.	
Storage for History	Database table where the Fact History data is stored.	
Storage for State	Database table where the Fact State data is stored.	
Storage for Summary	Database table where the Fact Summary data is stored.	
Summary Interval (ms)	If **Record Fact Summary** is enabled, designates in milliseconds, how often the Fact Summary data is written.	
Restart/Recovery		
Automatic start	Enable/Disable ☑/☐ automatic restart.	
Save in registry	Enable/Disable ☑/☐ to save persistent service in registry.	
Synchronous Control	Enable/Disable ☑/☐ to initiate synchronous service.	
Security		
Inherit Permission from Owner	Enable/Disable ☑/☐ inheriting of permission from owners' permission masks.	
Owner	User that owns the object.	
Permissions:	Permissions for users of the same group and others. Disable/Enable as required.	
	Group:	**Other Users:**
Read	Group members may read/view attributes of an object.	Others may read/view attributes of an object.
Change	Group members may change the attributes of an object.	Others may change the attributes of an object.
Delete	Group members may delete the object.	Others may delete the object.
Control	Group members may execute control actions such as start, stop, and disable.	Others may execute control actions such as start, stop, and disable.
Execute	Group members may execute operational commands on the object.	Others may execute operational commands on the object.

Table 4-18. File Monitor Expert Configuration	
Property	**Description**
Streaming Options (Refer to Section 4.16.)	
Application name	Sets application name
Data center name	Sets data center name
Exclude filter (regexp)	Ignore facts that match specified regular expression
Include filter (regexp)	Log facts that match specified regular expression
Location	Sets server location
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework)
Streaming configuration	Streaming configuration block name

*** Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact history is only maintained up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to section 4.8.3.4 for more detailed information on fact volatility.)

4.5.5.2 Multiple Log File Support

A File Monitor can provide support for multiple files and events profiles within a single File Monitor instance. The File Monitor will publish the following additional facts:

- EventOccurence
- EventTime
- EventText
- History/EventOccurence
- History/EventText
- History/EventTime

This is accomplished only after the File Monitor has been deployed.

Add Log File and Event Profile:

1. Expand the deployed File Monitor.
2. Right-click the *Logs* folder and click **Add Log File**.
3. Enter Log File properties and click **OK**.

   Log file – Path where log file is located.

   Log file alias – User-defined name for log file.

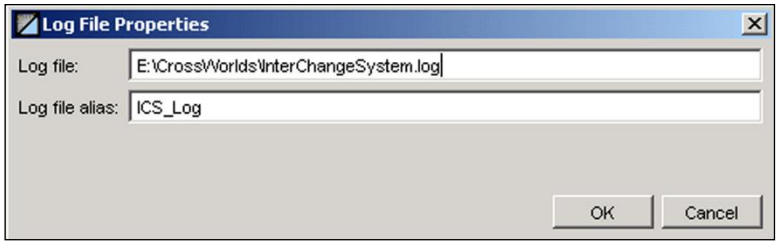


Figure 4-50. Adding Log Monitors

4. Expand the folder for the Logs file.

5. Right-click *Logs* folder and click **Add Event Profile**.
6. Configure the event profile as described below.

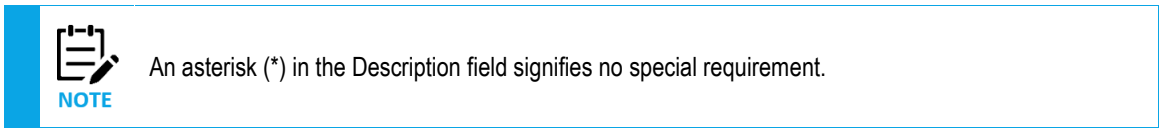


Table 4-19. Event Profile Properties

Property	Description
General	
Event profile name	User-defined name for event.
Event ID	ID of the event to look for.
Event type	Specifies the severity of the event. Field values are from 0 through 8.
History size	Number of event occurrences to keep in history. (1 = only the most recent occurrence is stored.)
Filters	
Event begins as	Specifies text the event must begin with. Typically specified as Error*
Event ends as	Specifies text the event must end with. Typically specified as *end
Event filter	Specifies text that must be contained within the event. Typically specified as *text* (*may occur more than once.)
Number of lines per event	Displays the event with the number of lines specified.

Example of History size: To detect an event occurrence of 10, type 10 in the History size text field on the Event Profile Properties screen. The Log Monitor will hold 10 entries in memory and publish the oldest entry under History. EventOccurence will count up from 1 and the Update-Count can be checked for the number of event occurrences since last reset (Reset-Fact() action). Reset_Fact actions should be used to reset the Update-Count on the EventOccurence.

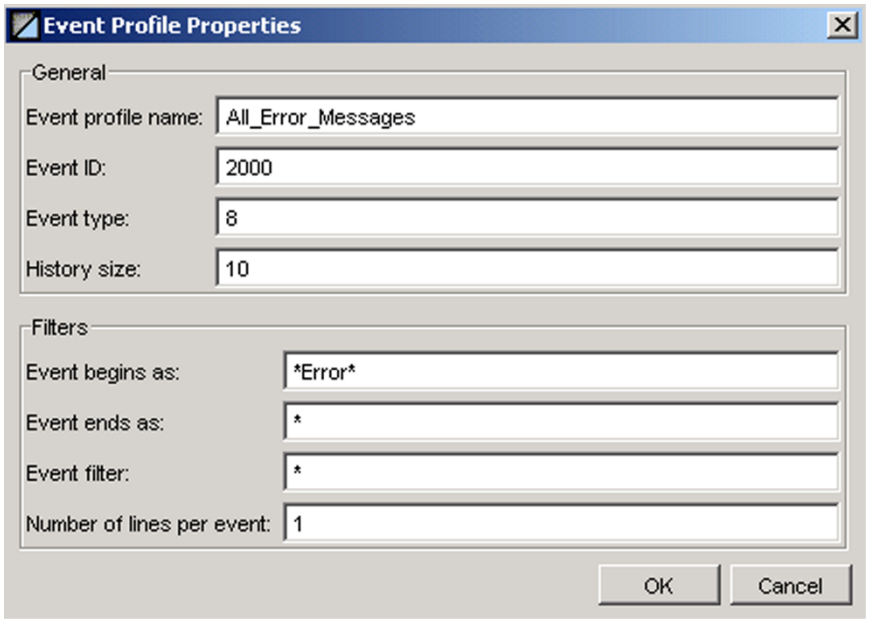


Figure 4-51. Event Profile

Filters Example:

The line

error this is a text message that I want that contains red and ends with blue

can be specified in two ways:

1. *Event begins as*: error*
   Event ends as: *blue
   Event filter: *red*

OR

2. *Event begins as*: *
   Event ends as: *
   Event filter: *red*blue

If 3 lines were specified as the *Number of lines per event*, then the result would be:

error
red
blue

Add New File Monitor Subfolder:

1. Expand the folders for the deployed File Monitor.
2. Right-click the *Folders* subfolder and click **New Folder Monitor**.
3. Enter Folder Monitor properties and click **OK**.

Table 4-20. Folder Monitor Properties

Property	Description
Folder alias	User-defined name for folder.
Folder name	Path where log file is located.
File filter	Directory of filter. For example: /var/x.log
Include sub-folders check box	Enable/Disable ☑/☐ to include sub-folders.

NOTE

An asterisk (*) in the Description field signifies no special requirement.

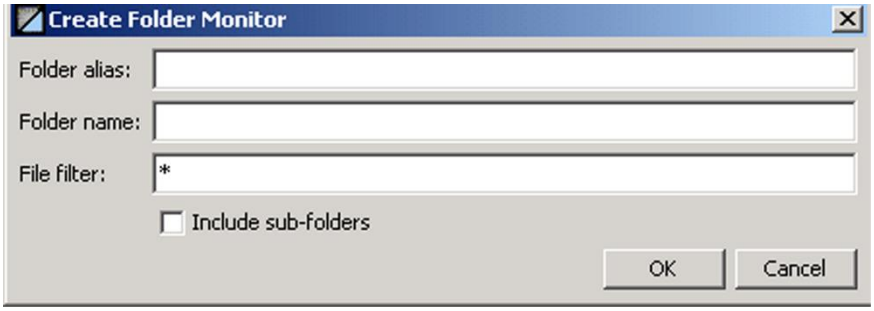


Figure 4-52. Creating New Folder Monitor

Edit File Monitor:

1. Right-click File Monitor to be edited and click **Properties**.
2. Change File Monitor properties and click **Apply**.

Remove File Monitor:

1. Right-click File Monitor to be removed and click **Remove**.
2. A Confirmation window will be displayed. Click **yes** to remove the File Monitor.

4.5.6 Deploying WebSphere MQ Experts

See *AutoPilot M6 Plug-in for WebSphere MQ* for instructions on how to configure and deploy WebSphere MQ experts.

4.6 Managers

The operational structure within M6 is similar to a human management structure. A manager is an M6 management service that has the authority, responsibility, delegation, and decision-making ability. M6 managers have the following characteristics:

- Mobile software components equipped with policies and contacts.
- Filter information before forwarding up the hierarchy – other managers.
- Managers behave similarly to M6 experts, except managers are not specific to a managed resource and can be equipped with advanced automation policies and functions.
- Managers can be made aware of, and manage, any application through the subscribing experts.
- Managers communicate with each other through policies and facts.
- Managers can be created on any CEP server from the M6 User Console. Unlike experts, managers can be moved from node to node so that they are able to function from the most suitable location.


NOTE

Service names are unique in the M6 domain. Contact your M6 Administrator for domain naming conventions.

Each M6 manager has the following:

- Subordinates called Contacts that are lists of managed experts and managers. Once an expert is included in the contact list, its facts are routed to its Manager for evaluation. Contacts can be experts on any node or other managers. M6 managers can be organized into a management hierarchy.
- Managers that subscribe to facts published by their contacts. They make decisions based on Policies and take actions on these facts. Policies are lists of policies that define manager's behavior based on facts published by the services included in the manager's contact list. (*also see*: Policies).
- Capability of publishing summaries of facts for superior managers (including human managers) and consoles.
- Managers that can be programmed to send notifications via e-mail, pager, or user-defined action.


IMPORTANT!

Managers must be deployed on the domain server and/or CEP server before policies can be deployed.

4.6.1 Built-in Managers

M6 has two built-in Managers.

- Policy Manager
- Speed Manager

4.6.1.1 Policy Manager

The Policy Manager is used to manage policies which are manually deployed within a domain. You will need to deploy at least one policy manager on each CEP server monitored. All policies under this manager must be manually deployed after Startup.

4.6.1.2 Speed Manager

The Speed Manager automatically loads policies from a user-specified speed folder (ds://folder). All business views and business policies located in the speed folder will be loaded automatically upon Manager Startup.

Using the Speed Manager gives users the following advantages over other managers:

- Simplification and ease of use.
- Faster loading time when more than one policy needs to be loaded.
- Easier deployment model.
- Assists in replication time. For example: if 10 policies to be loaded, user only needs to load once. If Policy Manager were used, user would have to load 10 times (each one separately).

System Services

System Services are an instance of the Speed Manager using the default <server>_facts. All managers load system policies designed to monitor each server.

The default location, for System Services can be changed by modifying the following property:

```
property server.system.speed.folder=ds://SYSTEM/nodes
```

Each loaded policy has the same name as the file located under the speed folder with applied manager convention default: SYS_[*policy*]. The default convention can be changed by defining a property:

```
property server.system.naming.convention=SYS_%name%
```

where `%name%` designates the name of the policy.

4.6.2 Deploying Managers

1. Click **Deployment Tool** to display Directory Viewer.

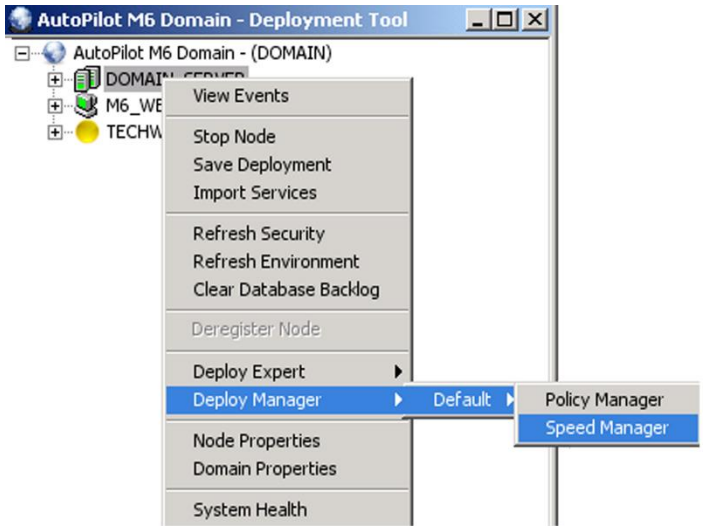


Figure 4-53. Deployment Tool and Networked Nodes

2. Click the node to receive the manager.
3. Right click Node to display Node Menu. Click **Deploy Manager**. The default menu is displayed.

4. Click the Manager. The Manager Properties screen is displayed.

> **NOTE**
>
> 1. Each Manager deployed in your Nastel M6 Network must have a unique name.
> 2. The configuration and requirements of each Manager is unique. Configuration Screens for each Manager will be different in one or more areas.

5. Review Manager Configuration. (Refer to section 4.6.3, Manager Configuration for details.)

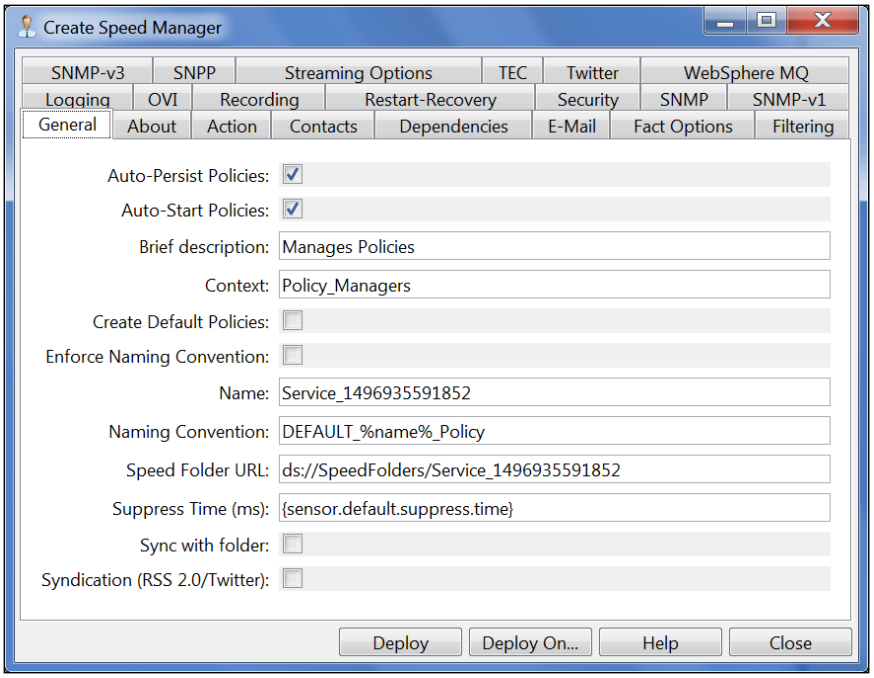


Figure 4-54. Deploying Managers

> **NOTE**
>
> The **Suppress Time** option allows you to suppress alerts for the time specified. This option is primarily used when testing so the user will not be inundated with alerts.

6. Click **Deploy**. The Manager will be deployed on the Node selected in step 2. Click **OK** to verify creation of the new service. The system will create a service name and verify the node you selected for deployment.

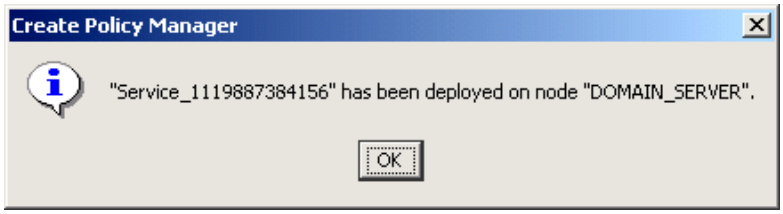


Figure 4-55. Deploy Manager

OR

7. Click **Deploy On** to deploy on multiple Nodes or Domains within Nastel M6 Network.
8. Select *Unique name* format to be used on each Node. Click the preferred naming convention.

   Click each node where the manager will be deployed. In *Deploy On* mode all nodes to receive managers must be selected, node selected in step 2 must be re-selected. Keep in mind that each

name can only be used once per CEP server/domain server. The check mark in the *Deploy on these nodes* box indicates nodes selected to receive the manager.

9. Click **Deploy**.

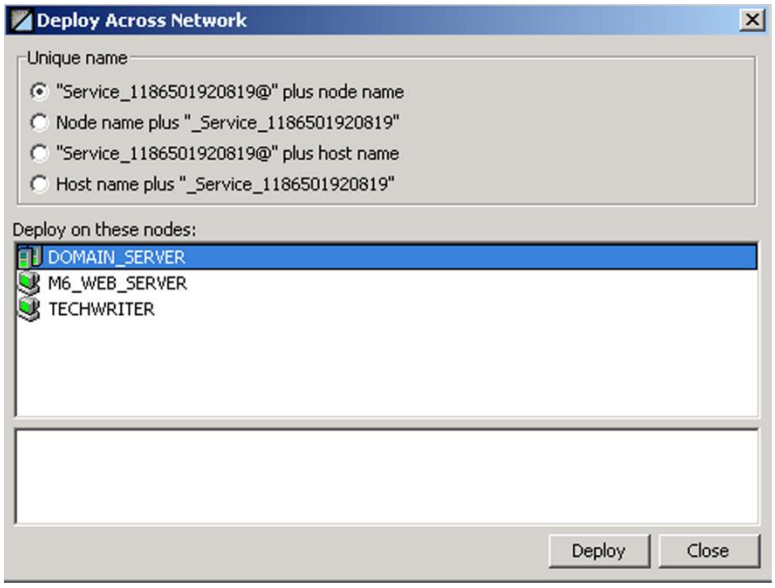


Figure 4-56. Deploy Manager On

4.6.3 Manager Configuration

It is important to fill-in "Name" when deploying managers; otherwise, system will assign a default name that consists of *Service_{timestamp}* format. Select *Naming Convention* and enable *Enforce Naming Convention* if all policies need to have same naming convention automatically applied by the manager. Users may provide e-mail and SNPP (Simple Network Paging Protocol) details, when events triggered by deployed policies need to be forwarded via e-mail or pager. Policy alerts can also be syndicated via an RSS news feed. Configure or customize other properties as required.

Table 4-21. Manager Properties

Parameter	Description
General	
Auto-Persist Policies (Speed Manager Only)	Enable/Disable ☑/☐ to enable registry persistence for all automatically loaded policies.
Auto Start Policies (Speed Manager Only)	Enable/Disable ☑/☐ to auto-start all automatically loaded policies.
Brief description	A short description of the Manager. This name will appear in brackets (e.g.: [Process Monitor]) on screen with listed Node.
Context	User defined category that will be registered in the Domain Server. Context is displayed as folder icon under each CEP server.
Create Default Policies	Enable/Disable ☑/☐ to create default policies that were pre-packaged for this manager. Default is disabled.
Enforce Naming Convention	Enable/Disable ☑/☐ to enforce naming conventions for all policies. Default is disabled.
Name	The name that uniquely identifies the manager in the domain. Enter or modify name as applicable, or in accordance with local guidelines. Variations of name are used when deploying services on multiple Nodes. No spaces or blanks are allowed in Service Names.

Table 4-21. Manager Properties	
Parameter	**Description**
Naming Convention	Naming convention that applies to all policies. The manager automatically applies Naming Convention to all subsequent Policies. %name% stands for the name of the policy as supplied by the user.
Speed Folder URL (Speed Manager Only)	Location of all policies loaded automatically upon Manager's Start.
Sync with folder	Synchronize policy deployment with folder contents.
Syndication (RSS 2.0)	Enable/disable policy syndication, via RSS news feed, of policy alerts. By default all RSS feeds go into `<aphome>/rssfeeds`. The folder must be created before enabling syndication. You can override the default RSS location by using "`property server.rss.folder=<rss_location>/`" (must end with a file separator). The feed name will be `<manager_name>.xml` file.
About	
Package Title	Implementation title of the source package.
Package vendor	Name of implementation vendor.
Package version	Version of implementation vendor.
Action	
Enable Action	Enable/Disable ☑/☐ to automate actions for all alerts.
User Action	User-defined script or executable triggered for every alert.
Contacts	
Contact List	A space separated list of services managed by this Manager.
Dependencies	
Platform dependencies	Comma separated list of Operating Systems platforms dependencies.
Service dependencies	Comma separated list of service dependencies.
E-Mail	
CC Recipients	Comma separated list of CC recipients (e.g., Bob@hotmail.com)
E-mail Notification	Enable/Disable ☑/☐ E-mail notification.
E-mail User Name	User name as defined in the email server (e.g., John)
Enable STARTTLS	Enable/Disable ☑/☐ STARTTLS.
Outgoing Mail Server (SMTP)	Name of SMTP server for outgoing mail.
Primary Recipients	Comma separated list of primary recipients (e.g.: John.J@hotmail.com)
SMTP Authentication	Select if SMTP server requires authentication.
SMTP Port	SMTP port
SMTP User Name	SMTP user name
SMTP User Password	SMTP user password
Fact Options	
Exclude Expire Filter (regex)	Do not expire facts that match the regexp.
Exclude Fact filters	Comma separated list of fact paths to exclude during publishing.

Table 4-21. Manager Properties	
Parameter	**Description**
Expire Facts(ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size*	Automatically maintains the specified number of samples for each published fact in memory.
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Fact service alias	Override fact service prefix for all published facts. Facts appear under specified service name.
Include Expire Filter (regexp)	Expire facts that match the regexp.
Include Fact Filters	Comma separated list of fact paths to include during publishing.
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to **Fact History Time** or **Fact History Size** which ever limit is reached first. If disabled newer history samples replace older on a rolling basis.
Filtering	
Exclude Facts	Filter for excluding facts. For example: MyExpert1*
Include Facts	Filter for including facts. For example: MyExpert2*
Logging	
Audit	Enable/Disable ☑/☐ service audit trace.
Log name	Log name associated with the service.
Log service activity	Enable/Disable ☑/☐ service activity trace.
Log size (bytes)	Size of Log in bytes. Real log size is the maximum value of server.log.size and logsize.
OVI**	
Application name	Application name reporting the event.
Enable OVI	Enable/Disable ☑/☐ for notifications using Open View Interface (OVI).
Message group	OVI message group for all reported applications.
Object name	Object name associated with reported event.
OVI URL	URL of the HP OVI Event Create Pluglet.
Service name	OVI service name associated with reported event.
Recording	
Anomaly Deviation Limit	Number of standard deviations above or below the mean.
Exclude Filter (regexp)	A regular expression filter to exclude certain facts from being written to the database. Facts have the format `expert\class\instance\leaf=value` such as in the example `Servers\Linux\Serv7\processes=40`.
Fact Anomaly Frequency	Frequency at which anomalies are checked and recorded.
Fact State Frequency	If **Record Fact State** is enabled, the value entered here specifies how often the Fact State is updated.
Fact Summary Frequency	If **Record Fact Summary** is enabled, used to write an intermediate summary record every Xth update to the fact during the Summary Interval. This is done to avoid waiting the full Summary Interval for a summary record to appear in the summary table.
Include Filter (regexp)	A regular expression filter to include certain facts being written to the database. Same format as described for the exclude filter.
Record Fact Anomalies	Enable/disable fact anomaly recording for this service.

Table 4-21. Manager Properties	
Parameter	**Description**
Record Fact History	If enabled, records every fact change into the History database. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.
Record Fact State	If enabled, records the last value published (current state) into the state database and restores that value when the CEP Server is stopped and restarted. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.
Record Fact Summary	If enabled, records summary record at the interval designated in the **Summary Interval (ms)** field into the Summary database. The exclude/include filters are respected. To define database tables and set AutoPilot options, refer to section 4.5.4.1.
Storage for Anomalies	SQL table where anomalies are recorded.
Storage for History	Database table where the Fact History data is stored.
Storage for State	Database table where the Fact State data is stored.
Storage for Summary	Database table where the Fact Summary data is stored.
Summary Interval (ms)	If **Record Fact Summary** is enabled, designates in milliseconds, how often the Fact Summary data is written.
Restart-Recovery	
Automatic Start	Enable/Disable ☑/☐ automatic start.
Save in Registry	Enable/Disable ☑/☐ persistent services saved in Registry.
Synchronous Control	Enable/Disable ☑/☐ synchronous service initiation.
Security	
Inherit permissions from owner	Enable/Disable ☑/☐ to allow inheriting of permissions from the owner's permission mask.
Owner	User that owns this object.
Permissions	Permission for users in the same user group and others, see Account Permission Masks in section 4.3 for details.
SNMP	
Debug level (0-15)	SNMP debug level; 0 is the lowest level.
Enable SNMP-v1	Enable/Disable ☑/☐ notifications using SNMP-v1.
Enable SNMP-v2	Enable/Disable ☑/☐ notifications using SNMP-v2.
Enable SNMP-v3	Enable/Disable ☑/☐ notifications using SNMP-v1.
SNMP v2/v3 trap oid	Trap Object Identifier (OID): 1.3.6.1.6.3.1.1.4.1.0 which identifies the notification being sent as a trap.
Trap hostname	Name of the host where traps are sent.
Trap port	Port number of the trap listener.
User variable OID	Object identifier of a user-defined variable to be added to every trap, perhaps to identify the generator of the trap. Consult the SNMP system administrator for a valid OID that can be used. It could be a pre-existing SNMP variable, such as sysName, whose OID is 1.3.6.1.2.1.1.1.
User variable value	Text string value of the user variable. For example: alpha.nastel.com (if user variable sysName was used) or Nastel.AutoPilot (if some other user variable was used).
SNMP-v1	

Table 4-21. Manager Properties	
Parameter	**Description**
Enterprise OID	SNMP enterprise OID. The dotted numeric path always starting with 1.3.6.1.4.1 (iso.org.dod.internet.private.enterprises) and ending with the object ID assigned by Internet Assigned Numbers Authority (IANA) to the vendor of the device being monitored. For example: If using Nastel's object ID of 1796, then the SNMP Enterprise OID would be 1.3.6.1.4.1.1796, which is used in the context of Nastel's WMQ MIB. You must know the enterprise ID for the vendor of the device or software being monitored. For example: Cisco, Juniper Networks, or Nastel.
Generic trap	Generic trap indicator: 0 – coldStart 1 – warmStart 2 – linkDown 3 – linkUp 4 – authenticationFailure 5 – egpNeighborLoss 6 – enterpriseSpecific If 'Generic Trap 6' is specified, it is always followed by 'Specific Trap n'.
Specific trap	A number identifying the specific trap number.
Trap community	Name of community (group) receiving traps. This is a security feature in SNMP-1 to prevent unauthorized monitoring of your devices or generating spoofed traps.
SNMP-v3	
Authentication Protocol	Encryption protocol used during password authentication.
Authentication	Enable/Disable ☑/☐ authentication.
Context engine id	SNMP context engine Identification.
Context name	SNMP context name.
Privacy password	Privacy password.
Privacy	Enable/Disable ☑/☐ privacy password.
User name	Authentication user name.
User password	Authentication user password.
SNPP	
Enable Paging	Enable/Disable ☑/☐ SNPP paging.
Outgoing Paging Server (SNPP)	Host name where SNPP is running.
Pager PIN	Pager's Personal Identification Number.
Paging Server Port	Port where SNPP server is listening. (See System Administrator for additional assistance).
Streaming Options (Refer to Section 4.16.)	
Application name	Sets application name
Data center name	Sets data center name
Exclude filter (regexp)	Ignore facts that match specified regular expression
Include filter (regexp)	Log facts that match specified regular expression

Table 4-21. Manager Properties	
Parameter	**Description**
Location	Sets server location
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework)
Streaming configuration	Streaming configuration block name
TEC* **	
Enable Tivoli TEC	Enable/Disable ☑/☐ notifications using Tivoli Enterprise Console (TEC).
Event Attributes	Assigns a value to any valid attribute. The attribute should be one defined for the event class. Separate multiple attribute=value expressions with spaces.
Event class	Specifies class of the event. It must match a class that is configured at the server.
Event message	Text of the event in double quotation marks.
Event source	Specifies the source of the event. If any blank spaces are in the source name, enclose the source name in double quotation marks.
TEC command	Path to the command that sends events to a TEC event server.
TEC config file	Specifies the name of TEC/wpostzmsg configuration file.
TEC server	Specifies the name of event server in name registry format. Enter @EventServer to have events submitted to locally named event server.
Twitter	
Twitter	Enable/Disable ☑/☐ Twitter posting of policy events.
Twitter Access Key	Twitter OAuth access key.
Twitter Access Secret	Twitter OAuth access secret.
Twitter Password (deprecated)	Twitter user account password.
Twitter User	Twitter user account name.
WebSphere MQ	
Facts to WebSphere MQ	Enable/disable writing facts to WebSphere MQ
Password	WebSphere MQ user password
Queue manager host	Name of the host where queue manager is defined
Queue manager name	Name of local queue manager
Queue manager port	Channel listener TCP port for the queue manager
Queue name	Name of queue where facts will be sent
Server connection channel	Name of the SRV CONN channel on the queue manager
SSL certificate store	SSL certificate store location
SSL cipher suite	Cipher suite that matches the CipherSpec of the channel
SSL enable	Enable SSL communication for channel communication
User ID	WebSphere MQ user ID

*** Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact history is only maintained

up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to section 4.8.3.4 for more detailed information on fact volatility.)

** If **OVI** selected, users must install and configure OVI and make oviEventCreatePluglet available. M6 health severities are mapped to OVO severities as follows:

Table 4-22. M6 and OVO Health Severities

M6 Severity	OVO Severity
UNKNOWN	UNKNOWN
EMERGENCY	CRITICAL
CRITICAL	CRITICAL
FAILURE	MAJOR
ERROR	MINOR
WARNING	WARNING
SUCCESS	NORMAL
DEBUG	UNCHANGED
INFO	NORMAL
NOTICE	NORMAL

*** If **TEC** selected, M6 health severities are mapped to TEC as follows:

Table 4-23. M6 and TEC Health Severities

M6 Severity	TEC Severity
UNKNOWN	UNKNOWN
EMERGENCY	FATAL
CRITICAL	CRITICAL
FAILURE	CRITICAL
ERROR	MINOR
WARNING	WARNING
SUCCESS	HARMLESS

4.7 Policies

Policies reside within one or more M6 managers. They are triggered upon changes in facts or upon any other built-in conditions supplied by users using business views wizard. These policies can vary in complexity depending on the automation tasks they are assigned to accomplish. A policy that must take into account a number of different conditions before the manager can act, is more complex. Complex policies perform data correlation and analysis.

NOTE

1. The configuration of each policy is unique and depends on the user-defined rules, thresholds, and criteria. Policies can only be deployed within managers. Deploy managers first if required.

2. The names of policies must be unique within a single manager.

This section covers the deployment and configuration of policies. The policies outlined in this section are four-level templates. They define non-application specific Policy development. Policies designed for specific applications are addressed in the supporting documentation for that software or option. The following policy template configurations are discussed in this section:

- **New View:** Loads a specified user defined business views into a manager. Business views are created using M6 User Console. Once loaded, business views run in the background and act on behalf of the owner.
- **New Filter:** Contains all the attributes of the view policy and is used by managers to filter information from Manager's contacts. The information is then published on behalf of the Manager that hosts the Filter Policy.
- **New Schedule:** Contains all the attributes of the filter policy and is a schedule policy that can schedule start-up/shutdown of any M6 service/experts/managers including policies. Services can be initiated at any time for a specified duration (in minutes) and any day of the week.
- **New Persistence:** Contains all the attributes of the schedule policy, plus maps M6 facts to any database table. It allows users visually map facts to DB table and save the mapping schema to a schema file. These schema files are saved with `.TBL` extension. The schema can then be deployed using persistence policy within any M6 manager.

TIP

Right-click the target manager and select "Deploy Policy" menu option, to deploy policies. Business views should be deployed using Business View Tool or Business View Explorer.

4.7.1 Deploying Policies

IMPORTANT!

Managers must be deployed on the domain server and/or CEP server before policies can be deployed.

1. Click **Deployment Tool** to display Directory Viewer.

NOTE

Policy icons are assigned by the system and are random. There is no functional deference implied.

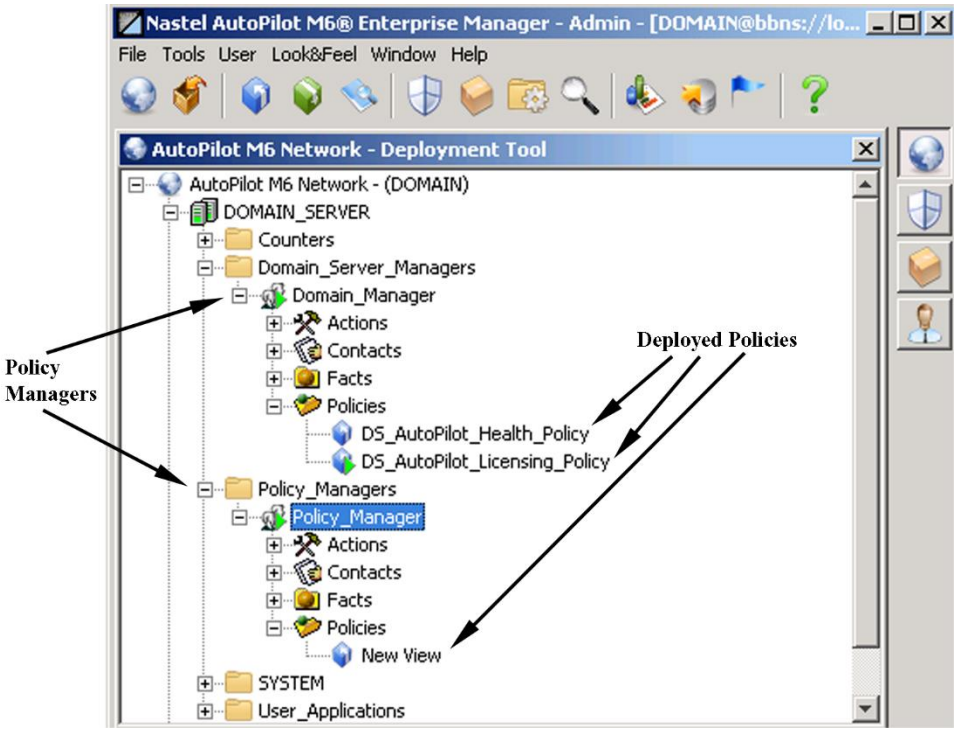


Figure 4-57. Deployed Policies

2. Right click the manager to receive the policy to display Menu. Click **Deploy Policy**. The Policy sub-menu is displayed.
3. Click desired Policy type to be deployed from the groups displayed.

> **NOTE** Built-in Policies may vary based on installation options exercised and local Nastel M6 deployment requirements. The default Policy templates are provided to enable users to develop custom Policies for local use.

4. Select the Policy to be deployed from the sub-menus. The configuration screen for the selected Policy will be displayed.

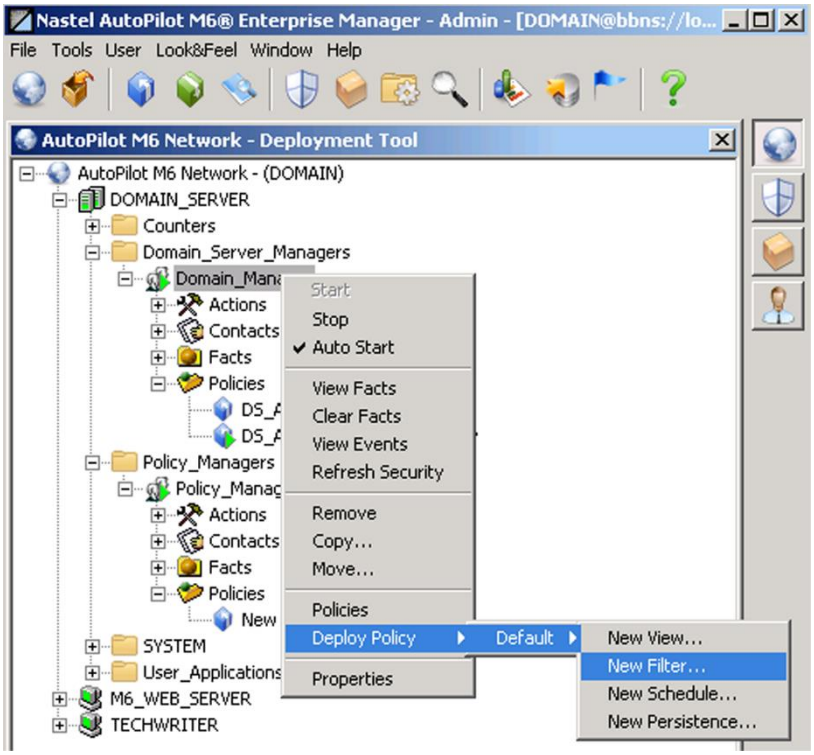


Figure 4-58. Policy Deployment

NOTE

Configuration and requirements of each Policy is unique. Configuration Screens for each Policy will be different in one or more areas.

5. Review Policy Configuration. Refer to section 4.7.2, Configuring Policies for details of policy parameter configuration.
6. Click **Deploy**. The Policy will be deployed with Manager selected in step 2.

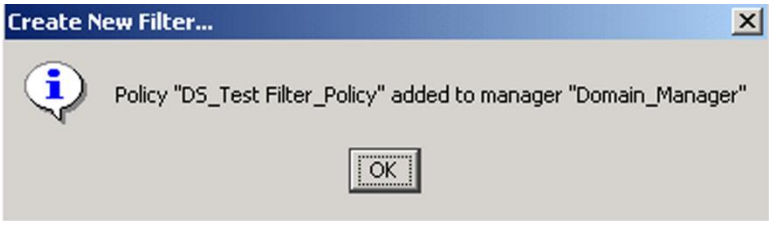


Figure 4-59. Policy Created

OR

7. Click **Deploy On** to deploy on multiple managers within domain or node. Click the manager to receive the policy and click **Deploy**. Deploy the policy within as may managers as needed; each time you deploy the policy an acknowledgement statement is displayed. The green check mark indicates successful deployment.

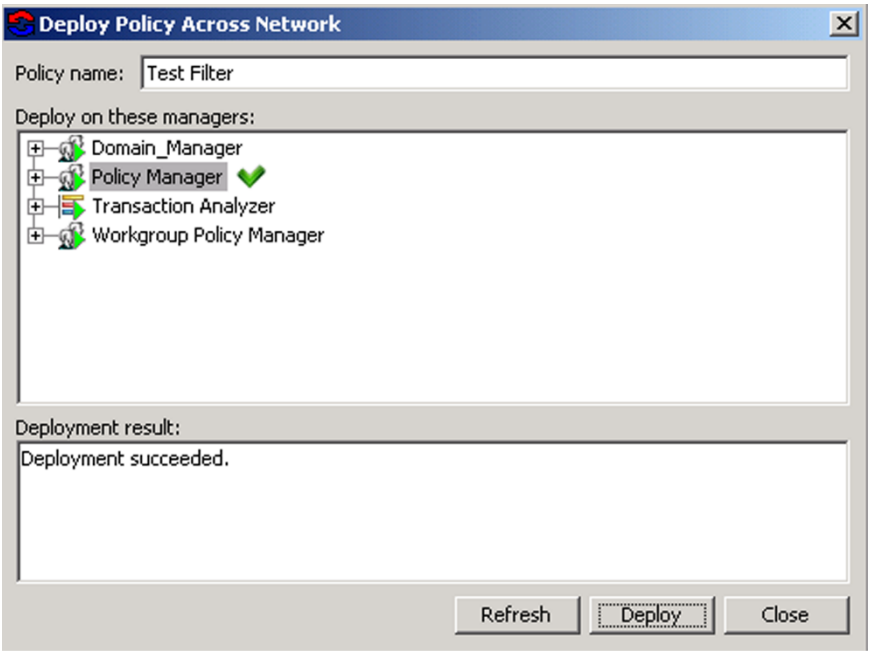


Figure 4-60. Policy Selection and Deployment

8. Click **Close** once you have deployed the policy where required.

Deployment View

To display all deployed policies, click **Tools > Deployment View** from the toolbar. The deployment panel displays server name along with manager name (manager@server label).

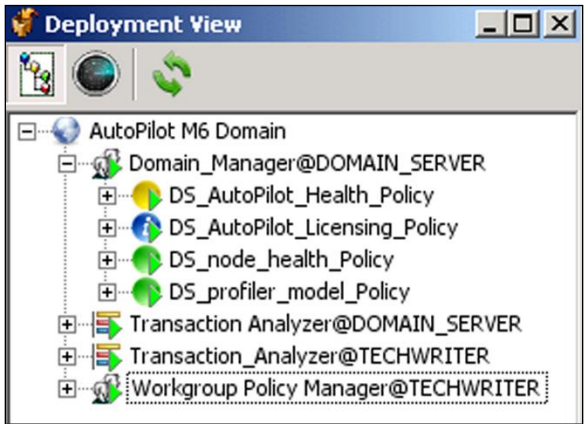


Figure 4-61. Deployment View

Sensor Search

There is an option to search within policies to find a specific sensor without having to expand all nodes.

1. Right-click a policy and select **Open** from the popup menu.

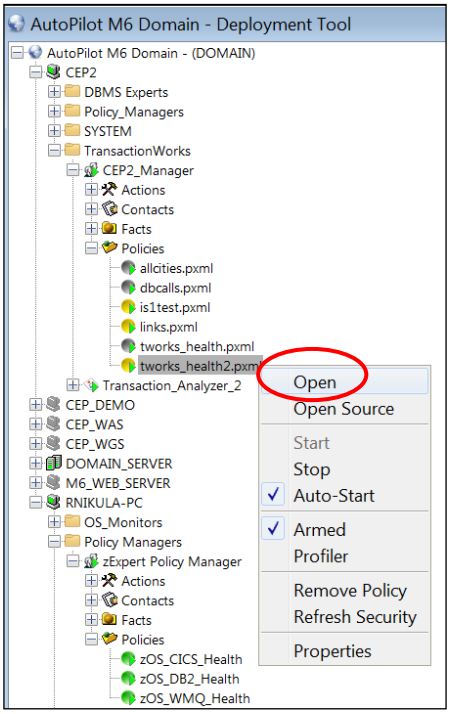


Figure 4-61A. Policy Menu

2. Click the **Filter Sensors** icon  to display the sensor list which allows you to search.

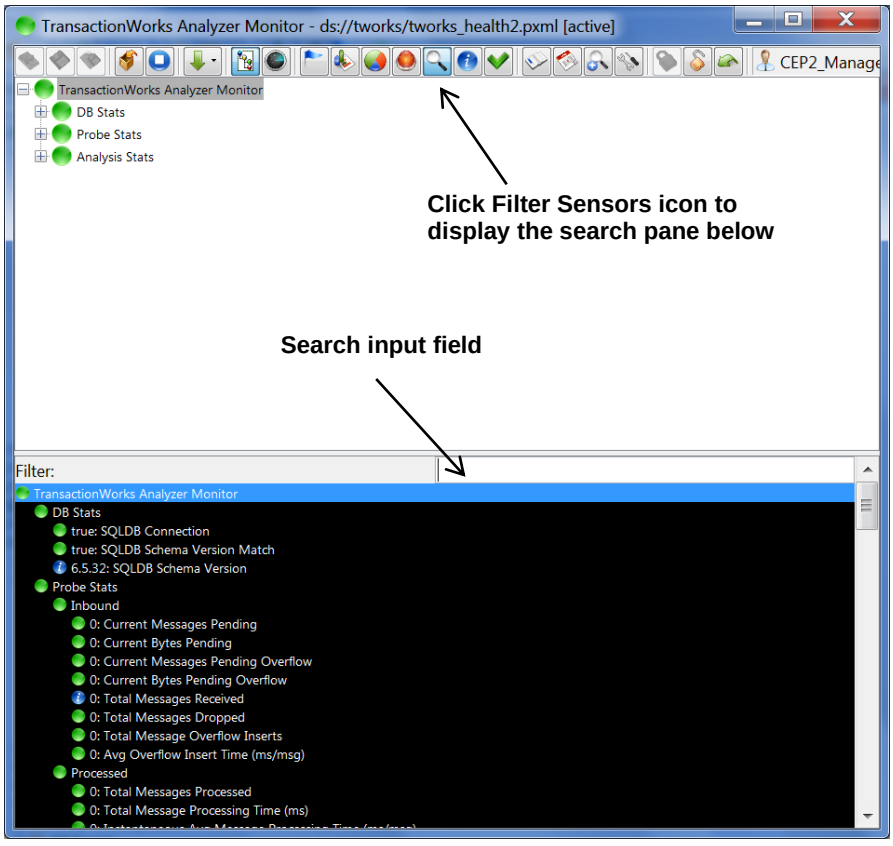


Figure 4-61B. Sensor Search List

3. In this example, we searched for a sensor with **lag** in its name. The result was the **Analysis Time Lag (ms)** sensor. Click **Analysis Time Lag (ms)** to expand the tree in the top pane to see where sensor is located (Figure 4-61D).

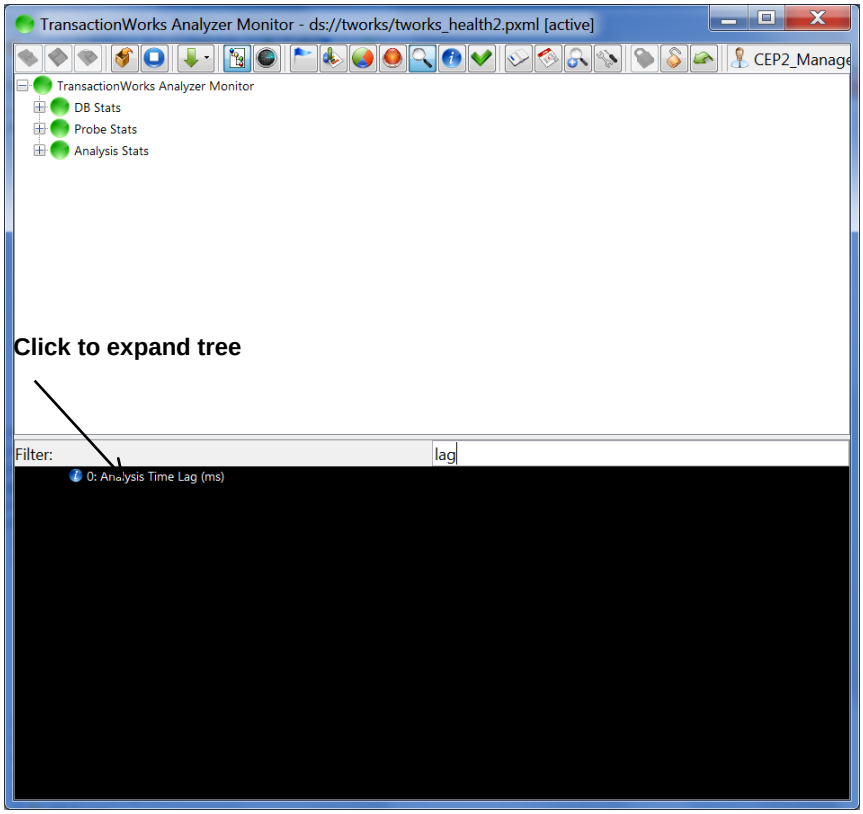


Figure 4-61C. Filtered Sensor List

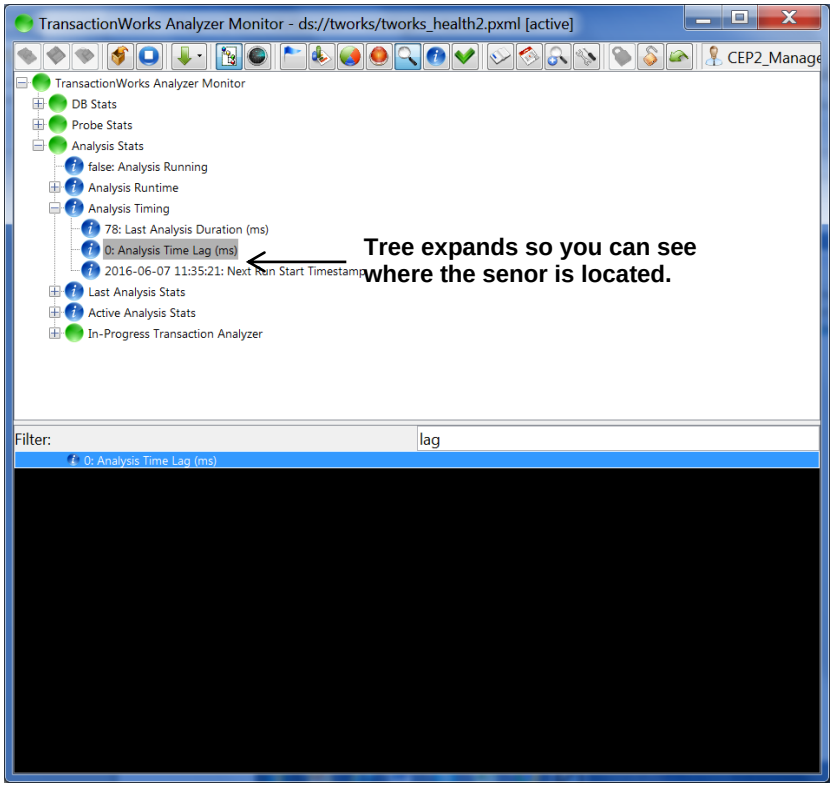


Figure 4-61D. Expanded Tree

4.7.2 Configuring Policies

The built-in policy properties are identified below. Properties that are unique to a policy are annotated accordingly. All other properties are common to all policies. Configure or customize the policy as needed for your applications.

NOTE When making changes to a policy, the changes are logged to a file in the Domain Server. These audit records are viewed by right-clicking the Domain Server and selecting View Events from the popup menu.

Table 4-24. Policy Properties

Parameter	Description
General	
Brief Description	A short description of the policy. This name will appear in brackets (for example: [Process Monitor]) on screen with listed Node.
Name	Default name is system assigned. Enter or modify name as applicable or in accordance with local guidelines. Variations of defined name are used when deploying Policies on multiple Managers/Nodes.
Data Base Scheme File (Persistence Policy only)	Data base scheme file determined by the database tool (example: `/opt/nastel/mytable.tbl`)
Schedule Services (Schedule Policy only)	Comma separated list of services to schedule (for example: Expert1, Manager, Policy@manager1)
Armed (New View Policy only)	Disarms/Arms the policy. Action notifications are enabled when Policy is armed.
Business view file (New View Policy only)	File name created by Business View Tool (for example: `ds://AutoPilot_EE/myfile.bsv`)
Publish stats (New View Policy only)	Disables/enables publishing of Policy statistics.
Publish view (New View Policy only)	Disables/enables publishing of Policy view from web client.
About	
Package Title	Implementation title of the source package.
Package Vendor	Name of implementation vendor.
Package Version	Package version as assigned by the vendor.
Dependencies	
Platform dependencies	Comma separated list of Operating Systems platforms dependencies.
Service dependencies	Comma separated list of service dependencies.
Fact Options	
Exclude Fact Filters	Comma separated list of fact paths to exclude during publishing.
Expire facts(ms)	Automatically expires facts that have not been updated in the specified time (ms).
Fact History Size*	Automatically maintains the specified number of samples for each published fact in memory.
Fact History Time (ms)*	Automatically maintains fact history not exceeding specified time in (ms).
Include Fact Filters	Comma separated list of fact paths to include during publishing.

Table 4-24. Policy Properties	
Parameter	**Description**
Lock Fact History	Enables/disables history collection after accumulating the first history batch up to **Fact History Time** or **Fact History Size** which ever limit is reached first. If disabled newer history samples replace older on a rolling basis.
Filtering	
Filter Facts	Comma separated list of fact filters.
Logging	
Audit	Enable/Disable ☑/☐ service activity trace.
Debug Trace	Enable/Disable ☑/☐ business view debug trace mode.
Log Name	Log name associated with the service.
Log Service Activity	Enable/Disable ☑/☐ service activity trace.
Log size (bytes)	Log size in bytes. Real logsize is the maximum value of server.log.size and logsize.
Performance	
Index Relations	Enable/Disable ☑/☐ to control the indexing of sensors. This allows inter-policy cross referencing. It is not recommended to use this option when speed and CPU usage is an issue.
Suppress Duplicate Value Propagation	Enable/Disable ☑/☐ to suppress propagation of duplicate sensor values to parent sensors. *Recommended configuration.* Disabling will cause memory and CPU degradation.
Suppress Real-time Health Aggregation	Enable/Disable ☑/☐ to suppress real-time sensor health aggregation. *Recommended configuration*. Disabling will cause memory and CPU degradation.
Profiler	
Profile sensor performance	Enable/Disable ☑/☐ profile sensor performance for each sensor. Default is disabled.
Profile state counts	Enable/Disable ☑/☐ profile state counts for each sensor. Default disabled.
Profile state timing	Enable/Disable ☑/☐ profile state timing for each sensor. Default disabled.
Profile thread	Enable/Disable ☑/☐ profile sensor worker thread performance. Default enabled.
Profile sampling rate(ms)	Profiler sampling and publishing rate in milliseconds. Default 30000.
Sensor profiler	Enable/Disable ☑/☐ sensor profiling. Default is disabled.
Restart and Recovery	
Automatic Start	Enable/Disable ☑/☐ automatic start.
Save in Registry	Enable/Disable ☑/☐ persistent services saved in registry.
Synchronous Control	Enable/Disable ☑/☐ synchronous service initiation.
Security	
Inherit permissions from owner	Enable/Disable ☑/☐ to allow inheriting of permissions from the owner's permission mask.
Owner	User that owns this object.
Permissions	Permission for users in the same user group and others, see Account Permission Masks in section 4.3 for details.
Database (New Persistence Policy Only)	
Data source name (DSN)	Logical data source name that points to the physical table.
Database Table	Name of the physical Database table (Example: `CPU_Table`).
DB logon ID	Logon ID required to access database.

Table 4-24. Policy Properties	
Parameter	**Description**
DB Password	Password to access the database.
JDBC driver	Class name of the JDBC Driver provider.
Schedule (New Schedule Policy Only)	
Begin Date	Date to trigger service (example: 09/20/2001).
Begin Time	Time to trigger service (example: 12:00 PM EST).
Duration (mins)	Maximum time allocated for each service (in minutes).
Expiration Date	Date to end services (example: 06/30/2003).
R-R Service Timer (ms)	Time in (ms) given for each service to complete (example: 60000 ms = one minute).
Round Robin Delay (ms)	Delay (ms) before next service is initiated by the schedule (R-R only).
Round Robin Schedule	Round Robin schedules service one after another.
Schedule Every	Comma separated list of days scheduled. (example: Day (everyday), Monday, Tuesday, etc.)

*** Fact History Size** and **Fact History Time** work in conjunction with each other. Facts can contain their value history by size, time, or both. If Fact History Size is not specified, then fact history is only maintained up to the specified time. If Fact History Time is not specified, then fact history is only maintained up to the specified size. When History is turned on (value specified), fact volatility can be measured. (Refer to section 4.8.3.4 for more detailed information on fact volatility.)

To control policy execution and performance, the following properties are used:

- **server.sensor.delivery.flowpct=0**. Setting the value between 0 and 100 temporarily suspends the flow to business views when the usage reaches 100%. The flow will remain suspended until the usage drops below the value specified by the variable. Setting the value to 0 or below, disables this property and the flow will always stay enabled. Enabling the flow control puts a maximum limit on the number of outstanding events for each business view and limits the memory growth associated with unlimited growth of queued events.
- **server.sensor.delivery.batch=2000**. The processing of sensor events is done in batches. It is not recommended to set this value below 100. It may create too much thread contention and decrease performance of the rule engine.

4.7.3 Using Managers for Displaying Policy Information

Information about all policies under a manager can be displayed by right-clicking on the manager and selecting Policies. This information includes:

- Name of policy
- Severity of policy in icon format
- Status – active, unknown, stopped, etc.
- Health percentage – 0% to 100%
- Type – Business View (bsv), policy, business process (bsp), etc.
- Value – if previously defined.

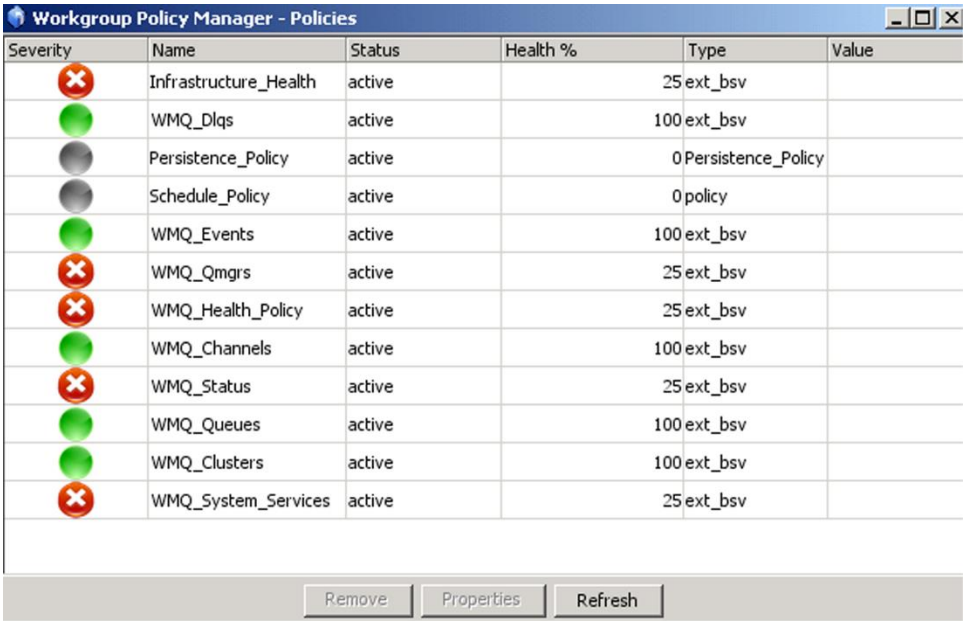


Figure 4-62. Policy Information

Selecting a policy enables **Remove** and **Properties** buttons. The policy list can be refreshed at any time by clicking **Refresh**.

Double clicking a bsv opens it.

Double clicking a policy displays its property screens.

NOTE This information can also be displayed by right-clicking **Policies > Manage**.

4.7.4 Policy Profiler

The performance of each business view can be tracked by enabling its profiler. Once enabled, the profiler metrics are published under individual managers in the *Facts\Profiler* subfolder. To enable the policy profiler, right click the policy name and select *Profiler (*see Figure below) or enable *Sensor Profiler* under the *Profiler* tab in Properties. (Refer to Table 4-24.) Checkmark indicates *Enabled*.

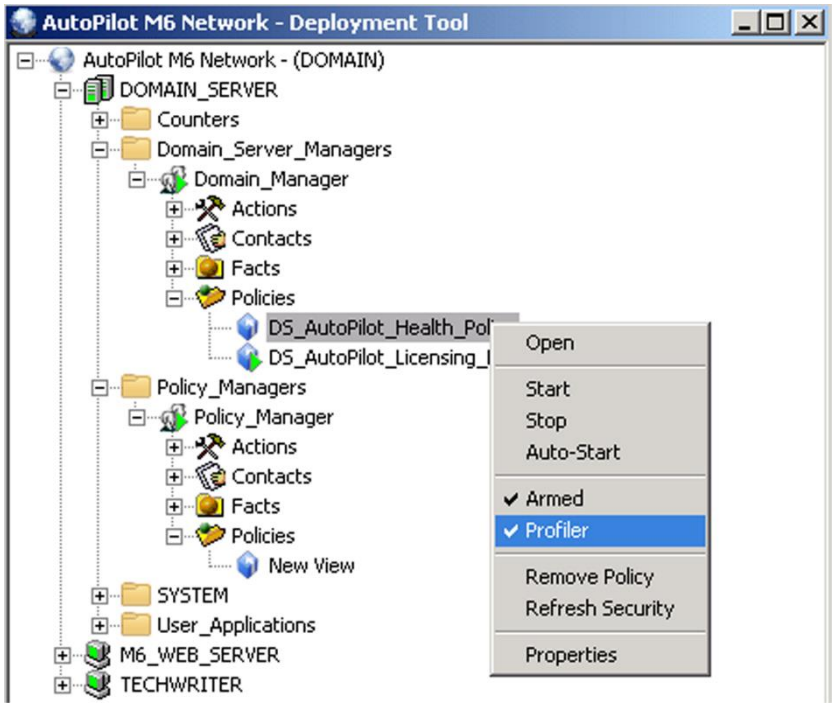


Figure 4-63. Enabling Policy Profiler

M6 profiler metrics are shown below.

Table 4-25. Profiler Metrics Per Policy	
Root Sensor Only	Thread\sensor_delivery_backlog
	Thread\sensor_rule_backlog
	Thread\sensor_delivery_rate_per_sec
	Thread\sensor_arrival_rate_per_sec
	Thread\sensor_worker_thread_alive
	Thread\sensor_total_delivered_events
	Thread\sensor_total_arrived_events
	Thread\sensor_total_dropped_events
	Thread\sensor_delivery_limit
	Thread\sensor_worker_thread_full
	Thread\sensor_worker_thread_util
	Thread\sensor_worker_thread_flow

Table 4-26. Profiler Metrics Per Sensor	
Action Related Metrics	Performance\Actions\sensor_last_action_rc
	Performance\Actions\sensor_last_action_exec_time_ms
	Performance\Actions\sensor_action_failures
	Performance\Actions\sensor_action_execs
	Performance\Actions\sensor_action_timeouts
Rule Performance Related Metrics	Performance\Rules\sensor_rule_exec_count
	Performance\Rules\sensor_last_exec_time_ms
	Performance\Rules\sensor_peak_exec_time_ms
	Performance\Rules\sensor_min_exec_time_ms
	Performance\Rules\sensor_rule_instr_count
	Performance\Rules\sensor_rule_rate_per_sec
	Performance\Rules\sensor_rate_per_sec
	Performance\Rules\sensor_idle_pct
	Performance\Rules\sensor_waiting_pct
General Run-Time	Performance\General\sensor_current_state
	Performance\General\sensor_runtime_ms
	Performance\General\sensor_time_waiting_ms
	Performance\General\sensor_child_count
	Performance\General\sensor_uptime_sec
Obtain State Counts – Number of times sensor spent in each state	StateCounts\sensor_idle_count
	StateCounts\sensor_running_count
	StateCounts\sensor_stopped_count
	StateCounts\sensor_starting_count
	StateCounts\sensor_stopping_count
	StateCounts\sensor_action_count
	StateCounts\sensor_subscribing_count
	StateCounts\sensor_error_count
	StateCounts\sensor_requesting_count
	StateCounts\sensor_waiting_count
Time Spent in Each State	StateTiming\sensor_idle_ms
	StateTiming\sensor_running_ms
	StateTiming\sensor_stopped_ms
	StateTiming\sensor_starting_ms
	StateTiming\sensor_stopping_ms
	StateTiming\sensor_action_ms

	StateTiming\sensor_subscribing_ms
	StateTiming\sensor_error_ms
	StateTiming\sensor_requesting_ms
	StateTiming\sensor_waiting_ms

4.7.5 Health and Load Balancing Policy

The SYS_node_health.bsv monitors a hosting CEP server's health and automatically adjusts its load. It retains 1.0MB worth of historical data about node key performance indicators in server_health.log under \naming and \localhost for each running instance. The data is helpful during problem determination. This user-defined policy is automatically installed on every node by the [*node*] _Facts system service. Users can modify it to meet their specific needs. By default, the policy source is located in `ds://SYSTEM//nodes` folder.

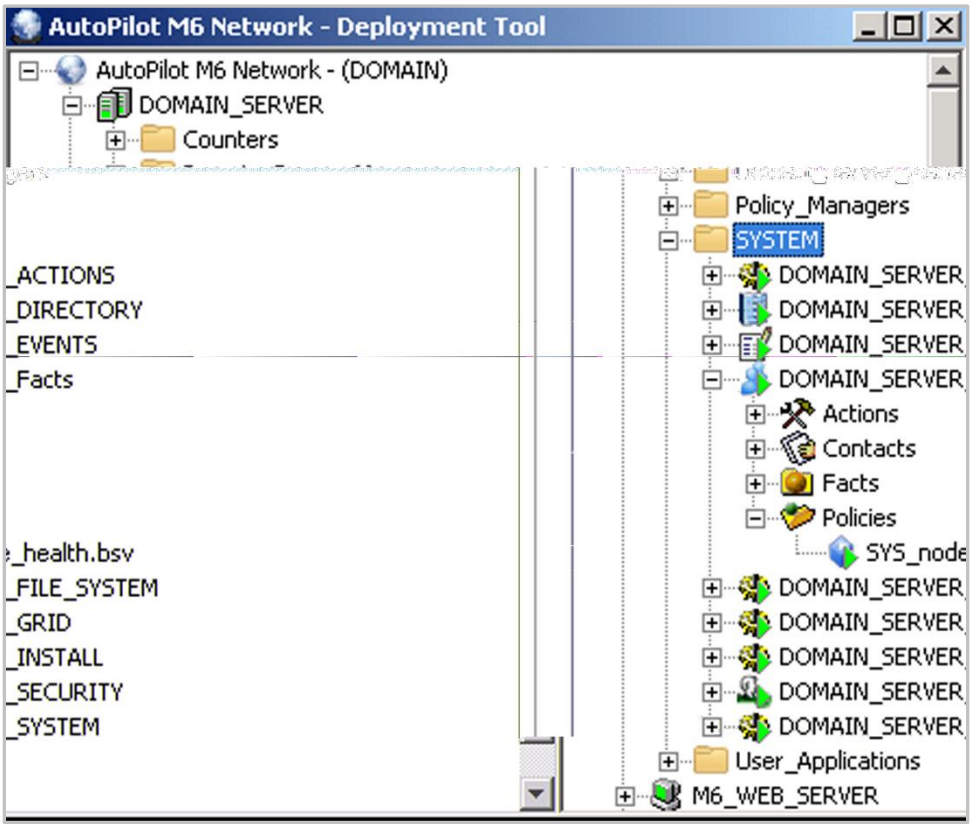


Figure 4-64. System Service

The Response Time Delay drains an activity if the response time is over a user-defined threshold and automatically stops all user-defined services when a threshold is reached. The policy response time rate is in milliseconds.

The Service Status and Recovery monitors the status of all local services registered in the domain and starts them automatically, if Auto-Start is set, when it is no longer overloaded.

Both sensors work together to load balance the hosting CEP server.

When defining properties, the *User Action* field under the *Alert* tab is automatically populated for Service Restart and Recovery and Response Time Delay. The properties *Sensor\sensor_turn_around_time_ms* and *Topic\fact_delivery_turn_around_ms* are used to measure how much time it takes for facts to be processed and displayed by the policies. Alerts can be set up to monitor these metrics and alert the user

prior to a load unbalance. The sum of both metrics is the total delay incurred by the system due to load. The longer the time, the less responsive the CEP server becomes. The user is alerted when the sum exceeds a user-defined limit which is usually 30000 ms.

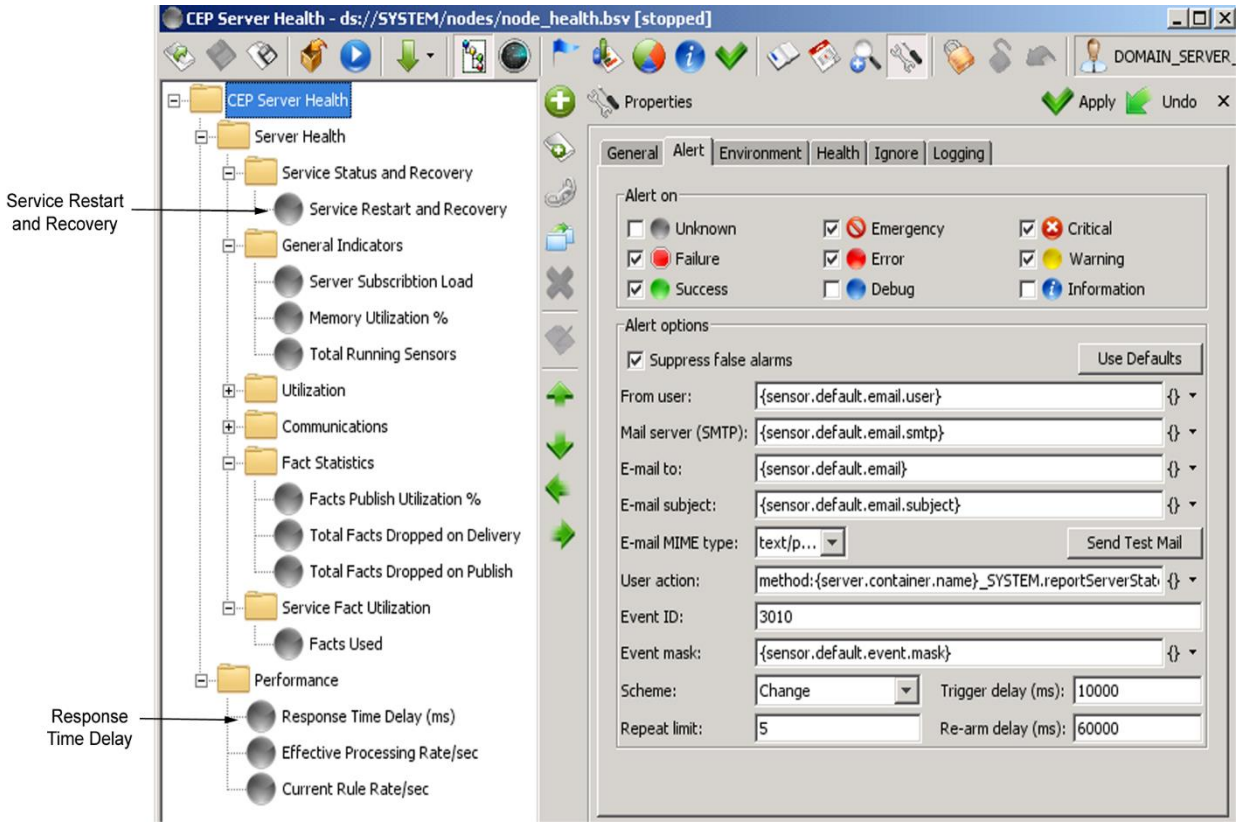


Figure 4-65. Default Alert Tab Properties Screen

The *User action* field can be cleared by highlighting the entire field and deleting from your keyboard.

The Auto Stop/Start action can be enabled by clicking on the down arrow to the right of the *Scheme* field and selecting *Change.*

The environment variables used in the CEP server Health.bsv include the following:

MID_LIMIT=8000

HIGH_LIMIT=25000

If the top-level CEP Server Health sensor folder is selected and **Properties** > **Environment** tab is selected, the two environment variables from above are visible:

MID_LIMIT=8000

HIGH_LIMIT=25000

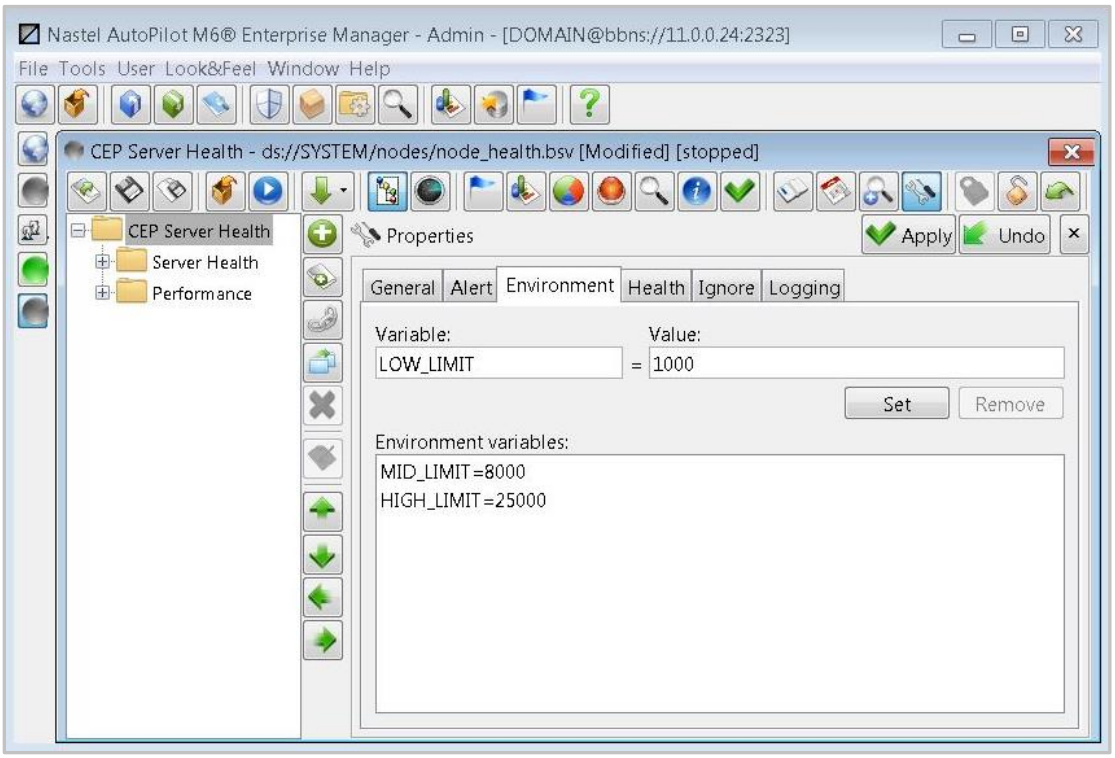


Figure 4-65-A. Adding a User Defined Environment Variable to a Policy

These can be used in any sensor of the policy. The figure above shows the two environment variables in the Environment tab of the policy properties.

Detail steps:

Select the policy SYS_node_health.bsv under DOMAIN_SERVER > SYSTEM > DOMAIN_SEFVER_FACTS > Policies.

Right-click and select Open.

Right-click on the top-level folder with policy name CEP Server Health, select Properties.

Select the Environment tab.

An additional environment variable LOW_LIMIT can be added, as shown in the figure, by entering he Variable Name and Value, clicking Set.

Then click the **Save** icon on the tool bar (2nd from left). When the save yes/no/cancel dialog to save the changes appears, click **yes**.

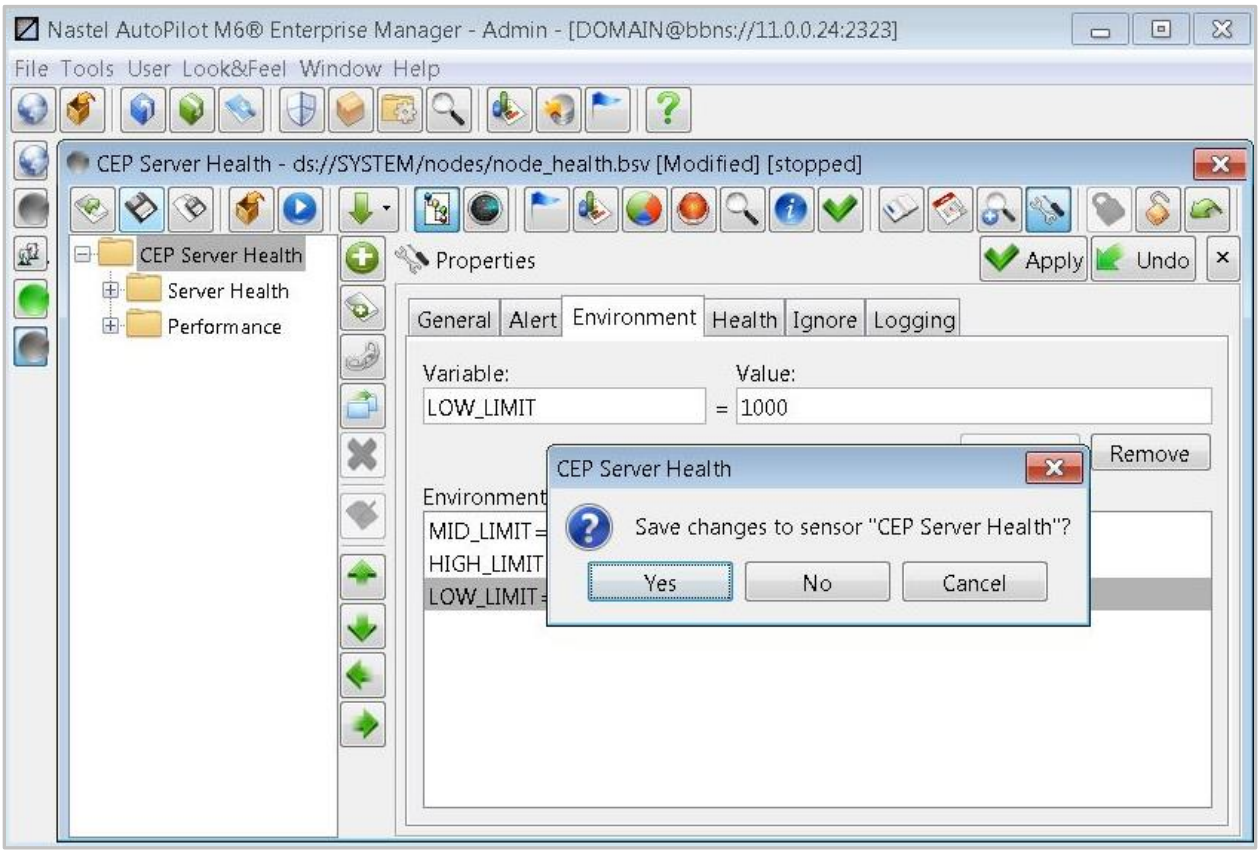


Figure 4-65-B. Adding a User Defined Environment Variable – Save Changes

The three environment variables can all be used in any sensor of the CEP Server Health policy.

The related business view (BSV) file is updated with the new env var inserted with the other two:

naming\policies\SYSTEM\nodes\node_health.bsv:

<Value>MID_LIMIT=8000,HIGH_LIMIT=25000,LOW_LIMIT=1000</Value>

If you want to define an environment variable for use by any one sensor, select the sensor and add the environment variable similarly as done above for the entire policy.

To use the env var in a sensor, right click a sensor, such as Service Restart and Recovery, select Wizard, and click through to the Dynamic Sensor Options, where the If-Then-Else rules are located.

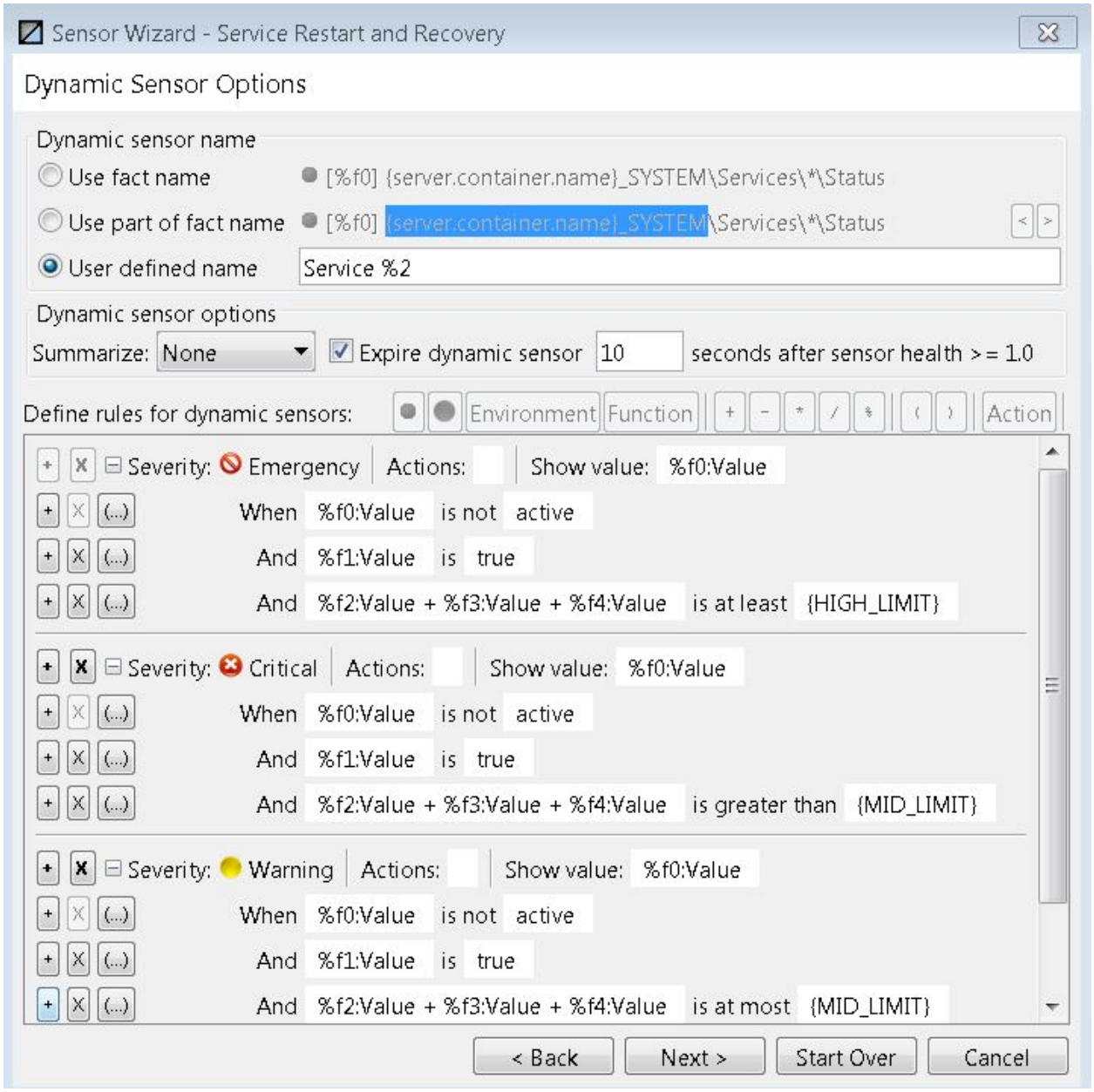


Figure 4-65-C. Policy CEP Server Health Sensor Wizard Service Restart and Recovery

Note how HIGH_LIMIT and MED_LIMIT are used.

If you select one of those fields, the env var choices will appear and the **Environment** button on the **Define rules for dynamic sensors** line is activated, to indicate you can make an env var choice.

4.8 Business Views

Business view is a collection of rules that define the desired-state of the *e*-Business environment. Using a graphical user interface (GUI) to identify and correlate facts, the user can graphically represent the state of *e*-Business, its applications and middleware and the impact on each in case of any degradation or failure.

Business views are customized to present the information in the way best suited for the user. Business view offers notifications by *e*-mail and/or pager. Actions can be associated to any business view to create powerful automation for any application.

Any business view can be deployed and is available on AutoPilot M6 Web Console. Business views can be accessed or shared via M6 User Console or M6 Web Console simultaneously. Refer to Appendix D for M6 Best Practices.

NOTE

In AutoPilot M6, new business views are saved with the extension .pxml to support standard java XML serialization. Support for .bsv extension still exists.

4.8.1 Business View Deployment Cycle

Business views cannot be started, unless deployed. Please review the deployment cycle below. All business views must be deployed before they can be activated.

Business View deployment cycle goes through the following stages:

- **Develop** – user or a team creates a business view, which is usually stored on the local file system.
- **Check-in** – business views must be saved or uploaded into central domain repository (using Business View Explorer or Business View Tool).
- **Deploy** – business view is assigned to run on one or more managers. At this stage all user defined rules, alerts and automations are being delegated to the selected manager(s). All the resources such as user actions and scripts, database definitions, file definitions must exist on the server where the assignment occurs.

NOTE

When starting the business views, Business View Tool will automatically trigger the deployment cycle. The deployment cycle must complete successfully before a business view can be started. Make sure all business views are saved onto a Domain Server.

4.8.2 Exploring and Managing Business Views

Business View Explorer is a tool supplied with AutoPilot Console and can be launched from console's *Tool* menu. Use Business View Explorer to:

- Explore business view domain repository
- Save/Export sensor information including facts, logic, conditions, alerts, etc. to an HTML, PDF, RTF, or XML file.
- Deploy business views from the repository onto one or more managers
- View where business views are deployed
- Set permissions for ownership and access by users and groups
- Lock and check out business views/check in and unlock business views.
- Upload locally created business views to domain repository
- Download business views from domain repository to your local system

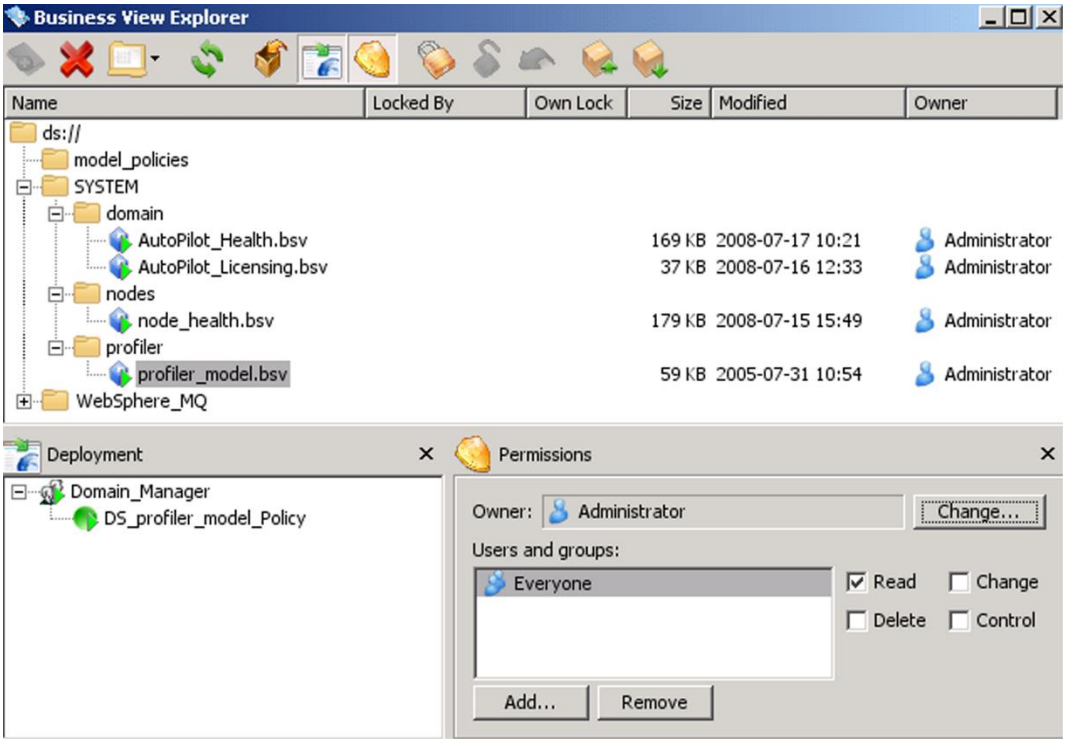


Figure 4-66. Business View Explorer

4.8.2.1 Setting Business View Permissions and Access

Permissions are set to specify ownership and access control to prevent unauthorized manipulation of a business view. Each business view has a defined owner. Only the owner can change ownership of a business view and specify which users and groups can read, change, delete and/or control a business view. Permissions are created and modified in the Business View Explorer.

Ownership

The creator of the business view is the default owner.

To change ownership, do the following:

1. Open the Business View Explorer by either clicking on its icon or by using the menu toolbar (*Tools/Business View Explorer*). See Figure above.
2. Click **Show Permissions** icon.
3. Click on the business view you want to change.
4. Click **Change** button next to *Owner* field. The following screen is displayed.

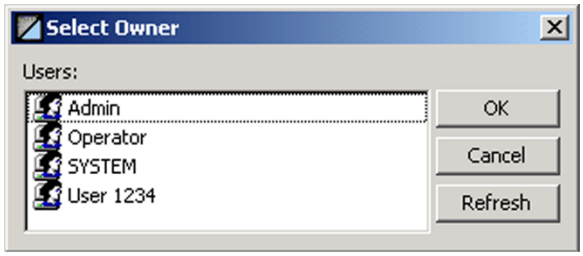


Figure 4-67. Selecting Ownership of Business View

5. Select an *Owner* from *Users* column and click **OK**.

Add/Remove Users and Groups

To add a User and/or Group, do the following:

1. Click **Add** button next to Users and Groups field. The following screen is displayed.

Figure 4-68. Selecting Users or Groups Permission

2. Select user or group and click **OK**.

To remove a User and/or Group, highlight user or group to be removed and click **Remove** button.

Modify Access

To modify access by a user or group, select *Read, Change, Delete* and/or *Control.*

- *Read* – user may only read/view the business view.
- *Change* – user may change any of the attributes of the business view.
- *Delete* – user may delete the business view.
- *Control* – user may execute control actions on the business view such as start, stop, or disable.

4.8.3 Business View Sensors

Sensors are the building blocks of every business views. They are organized into a hierarchical structure and feed one another with status and real-time events.
Sensors are used to monitor one or more facts. Sensors have the ability to enumerate facts at run-time and determine which facts need to be evaluated. It works very much like a FOR or WHILE loop on a variable list of facts. (Example: A difficulty in monitoring peak response time in all servlets within WebSphere is that the number of servlets could be large and may change at run-time.)

Sensors act like probes that evaluate facts or other sensors. Each sensor cycles through the following run-time stages:

- Receives inputs from facts or other child sensors
- Evaluates inputs by applying user defined logical rules
- Assigns severities to (source) fact/sensor input based on user-defined Sensor parameters
- Triggers an alert or an action as defined by the user
- May log its information to a log file or relational database

4.8.3.1 Sensor Categorization

Sensors can be categorized as follows and are configured in *sensor properties* under the *General* tab:

- **Service category** – classification of the monitored service. (Refer to Table 4-28.)
- **Service type** – name or identity of the monitored service.
- **Object type** – specific name of the resource.

4.8.3.2 Sensor Tables

The sensor database table layout below, lists all the sensors along with the total number of allocated characters for each (if applicable).

Table 4-27. Sensor Database Layout

Sensor	Total Characters/Value
ManagerName	varchar(128)
PolicyName	varchar(128)
SensorName	varchar(255)
Severity	varchar(32)
SensorValue	varchar(255)
NumericValue	float
Health	float
LogTime	timestamp
EventID	integer
ServerName	varchar(255)
OSName	varchar(128)
OSVersion	varchar(12)
OSArch	varchar(12)
OSUser	varchar(48)
APUser	varchar(48)
ServiceCategory	integer
ServiceType	varchar(48)
ObjectType	varchar(48)

The Service Category Table is defined below.

Table 4-28. Service Category

Service Category Number	Service Category Name
0	Hardware
1	Network
2	Server
3	Operating System
4	Middleware
5	Database
6	Application Server
7	Web Server
8	Web Service
9	Client
10	Application
11	IT Service
12	Business Service
13	Transaction
14	Policy
15	Miscellaneous
16	Other

4.8.3.3 Sensor Scripts

Scripts are used in the Sensors to define specified user actions and schemes. The Sensor Wizard is used to graphically select and define rules. The wizard generates the appropriate script automatically, which could be edited manually in advanced mode. Using advanced mode is recommended only for users familiar with sensor scripting language.

4.8.3.4 Fact Volatility

Volatility refers to the standard deviation of the change in value over time of a certain variable. Standard deviation is a measure of the dispersion of a set of values. It is defined as the root-mean-square (RMS) deviation of the values from their mean, or as the square root of the variance. If many data points are close to the mean, the standard deviation is small; if many data points are far from the mean, the standard deviation is large. If all data values are equal, the standard deviation is zero.

Fact volatility can be measured based on the standard deviation of the % change of the numerical value of the fact. Three new fact parameters have been added to M6:

- History-%Mean – mean based on % change
- History-% Variance – variance based on % change
- History-% Deviation – deviation based on % change.

For these new facts to be displayed, history must be turned on by specifying Fact History Size and Fact History Time from the properties screen of the selected sensor. (Refer to section 4.7.2, Configuring Policies, Fact Options tab, for information on specifying these parameters.)

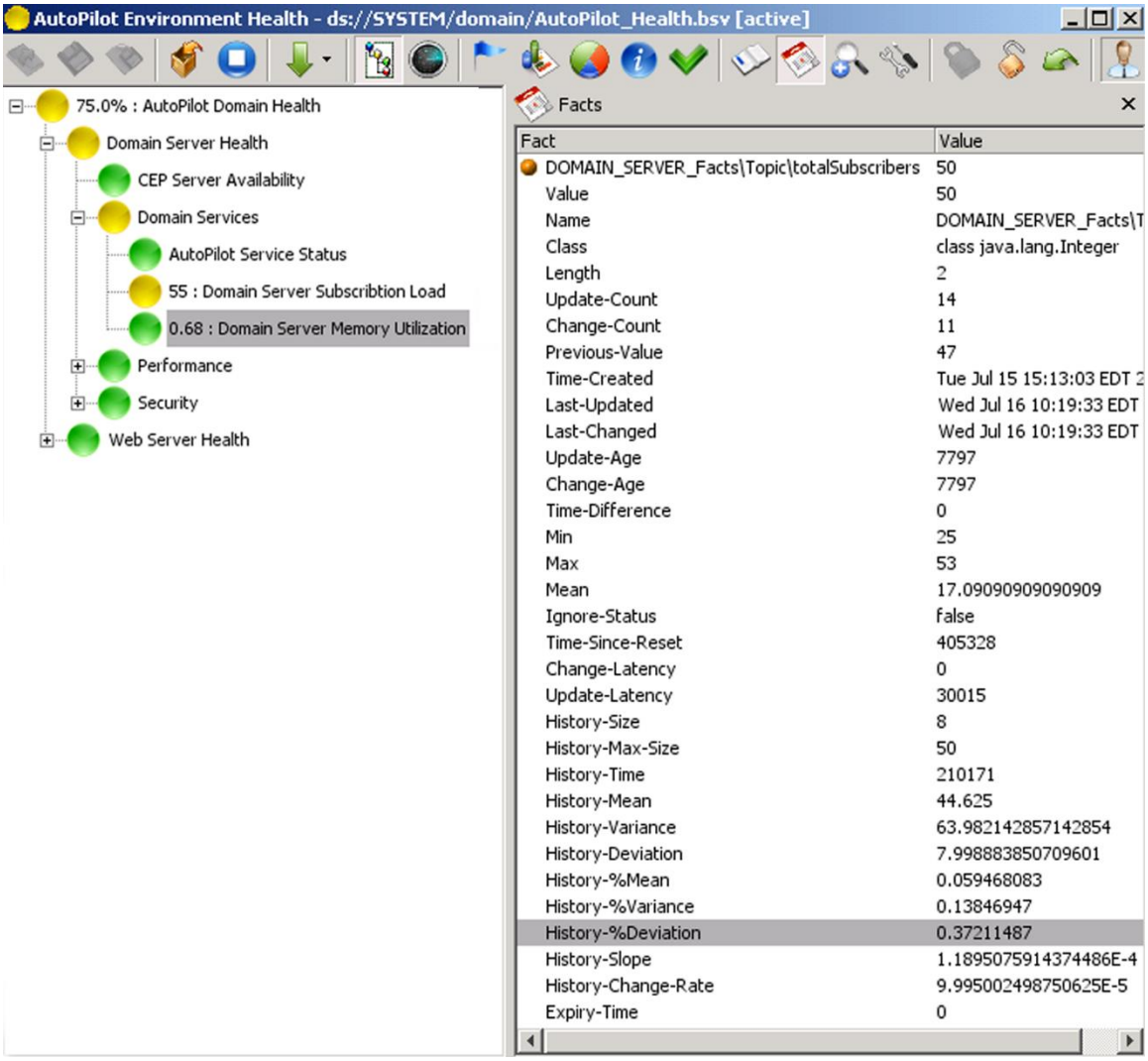


Figure 4-69. Fact Volatility

4.9 Alerts, Notifications and Rules

This section requires the use of *Business View Tool*. The tool can be accessed through M6 User Console.

> **TIP**
>
> When using Sensor Wizard, it recommended that you make use of environment variables (specified in "Environment" options for each sensor). Environment variables can be used within the wizard in the form {var}. For example, instead of *DOMAIN_SERVER_Facts\Java\free_memory*, one can *specify {SERVER}_Facts\Java\free_memory*. This makes business views easier to manage and maintain and removes hard coded variables. This is especially useful when creating business views for deployment in multiple environments such as TEST, QA, and Production. Environment variables can be defined either within business views or node.properties. (Refer to section 5.3 Server Runtime).

4.9.1 Configuring Alerts

Alerts are normally set while you are building business views. The alert determines the severity level that triggers the alert and the method of notification. Alerts can be set while you are building business views or can be configured during a sensor update or change. The severities are defined by the sensor health they represent. When selecting alert severities ensure the health percentages and weight reflects your needs. Configure alerts as follows:

1. Open the effected business view.
2. Stop the business view if it is running by clicking on the **Stop** button.

3. Right-click the sensor to be changed, the sub-menu will be displayed.

> **NOTE** If you want to make the same changes to two or more sensors, you can do so. Press **Shift** to select multiple sensors. Then make your changes on the appropriate **Properties** tab as described below.

4. Click **Properties**. The sensor properties for the selected sensor will be displayed on the right.
5. Click the **Alert** tab to access the alert properties.

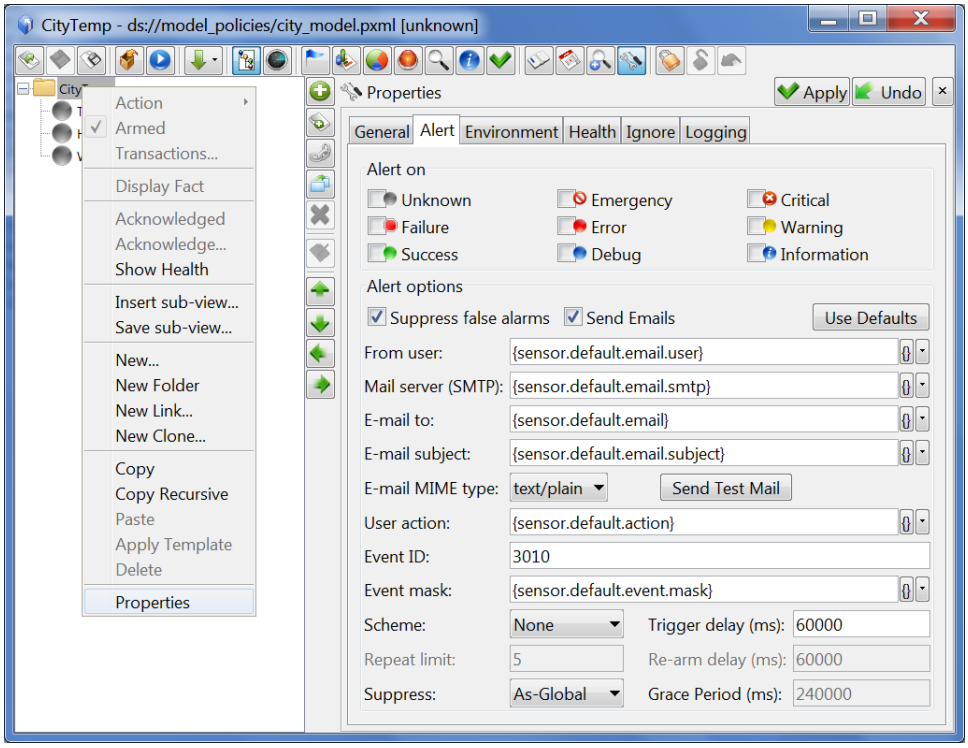


Figure 4-70. Changing Sensor Properties

6. Set the properties for your sensor alert using the options outlined in the table and figure below:

Table 4-29. Alert Properties

Parameter	Description
Severities: (Alert On)	Select the severity levels that will trigger notifications: Unknown: 0.0 (0%), default alert setting Emergency: 0.1 (10%) Critical: 0.25 (25%) Failure: 0.45 (45%) Error: 0.65 (65%) Warning: 0.75 (75%) Success: 1.0 (100%) Debug: 1.0 (100%) Information: 1.0 (100 %)
Suppress false alarms	Enable/Disable ☑/☐ false alarm. Intelligent False Alarm suppression logic. Disables duplicate actions/notifications when sensors are acknowledged/armed.
Use Defaults	Click button to use default properties that were defined in domain.properties on each domain server installation. Click **Yes** in confirmation box.
From User	Enter your mail server user id. Usually is the same as your email address.
Mail Server (SMTP)	Enter the name of the local mail server (SMTP server).
E-mail to	Enter the email address of the party to be notified (e.g., %f0:user@nastel.com).

E-mail subject	Enter user-defined e-mail subject line.
E-mail MIME type	Select text/plain or text/html e-mail.
Send Test Mail	Click button to verify e-mail reached the intended recipient.
User Action:	Use to specify an external command, shell script or executable: (e.g. *runaction.bat*). You may also pass business view context information that could be used within a script or action. Refer to section, Automation with User Actions for a complete list of supported action variables.
Event ID	User assigned event identification number. You can customize the number as needed, but only using numbers, no spaces (example: default = 3010. Custom = 4089). This ID is used to record events into M6 event log.
Event mask	Event format mask used to record events triggered by the sensor. Users may customize the event format. Refer to section, Automation with User Actions for a complete list of supported variables that can be used within the event mask.
Scheme	Open the Scheme menu and select the specific occurrence that will trigger notifications. **None** – Notification will not be generated. **Once** – Notification generated once per severity status change. All severities selected in Alert On section. **Change** – Notification generated with each change in status. **Repeat** – Sensor repeats actions/alerts until the senor goes into a state, which is not checked in the Alert On section. Sensor re-arms until the sensor goes back to an unchecked state. Notification frequency is based on Re-arm Delay intervals.
Trigger Delay	User defined delay times expressed in milliseconds (1/1000th of a second). Default is 60000 or 60 seconds. The sensor will delay notification for the prescribed time.
Repeat Limit	Maximum action/notification repetitions.
Re-Arm Delay	User defined re-arming delay of sensor notifications. Sensor will resample and resend alert notification continuously (per delay time) until condition is corrected. Default is 60000 or 60 seconds.
Suppress	From the drop-down menu select one of the following: **Off** – all facts are considered regardless of how old they are **As-Global** – defined in global properties **As-Parent** – defined in parentage **On-Change** – considers change add as set in grace period **On-Update** – considers update age as set in grace period
Grace Period	Age of facts, in milliseconds, which will not be considered for evaluation.

7. Click **Apply** to complete sensor change.
8. Click **Start** to restart the business view.
9. When sensors are functional, alert options are in effect. Alerts/E-mail notifications will be transmitted as specified in your business view.

4.9.2 Defining Sensor Rules

Sensor is a run-time user-defined object that encapsulates data, rules, behavior, notifications, logging into a single component. Each sensor may evaluate one or more facts at once. All facts must be available (must have values) in order to complete rule evaluation. Rule evaluation halts when one or more facts are not bound (have no values). Facts without a value may occur in several situations: fact does not exist, or fact value is equal to null. Evaluation will resume on fact change or when fact value becomes available.

Sensors define the purpose and scope of one or more rules. Selected facts are the basis of what sensors monitor and under what conditions. Thresholds can be fixed values, other selected facts (variables) or environment variables defined in the business view, CEP server or java environment.

Sensor rules are invoked on changes in the fact value and status. Each fact change is evaluated by the sensor rule and the result is mapped into any one of the supported severities such as SUCCESS, WARNING or CRITICAL. Automated actions, notification or logging can be triggered as a result of the rule evaluation.

Therefore, if the normal condition would indicate a status of "*Success*" the status would remain unchanged as long as the conditions for "*Success*" are maintained. If the condition changes to a condition outside of the "*Success*" parameters, the status is changed to reflect that condition. The change in status would change the existing severity status, thus triggering a pre-determined action to attempt to correct or report the condition. When the condition returns to normal, the status is updated accordingly. If the "normal" condition or status remains unchanged then the sensor will report (typically) an "information" or "Success" status.

Example: if the purpose of the rule was to monitor the number of facts being monitored, then we would base the rule on the *totalFacts* fact. *totalFacts* monitors the current number of facts monitored by all services within the CEP server.

If the maximum number of facts that can be monitored is 500, you may want to be alerted if it reaches 400 or 80% capacity. By monitoring the current number of facts with a condition of "more than" and the variable condition set at "400" the status will change if the total number of facts exceeds 400. The status will change to reflect the severity set in the rule. When the severity "Warning" is reached the rule triggers the action selected. In this sample, the action would be an audible beep. If the action you chose was to stop or update the fact, the associated fact in optional sensor/value would have been updated as specified.
Sensors have no pre-determined limit to the number of rules that can be defined.

TIP

The "/" character is a reserved symbol and cannot be used as part of any name in M6.

4.9.2.1 Defining Sensor Logic

When a sensor is transitioning to a checked (alert is enabled) state, the following logic is performed:

- If a sensor's previous state has not completed its notification before a new sensor state is completed, the previous alarm is suppressed.
- If a sensor's previous state has not completed its notification before a new sensor state is completed, notification is posted on the new sensor *only* if the new state is more severe. The previous alarm will be suppressed.
- If the new state is less severe, notification is *not* posted and there is a false alarm condition. The previous alarm will be suppressed.

4.9.2.2 Creating Sensors with the Wizard (Non-Dynamic)

1. Open the business view to receive the sensor.

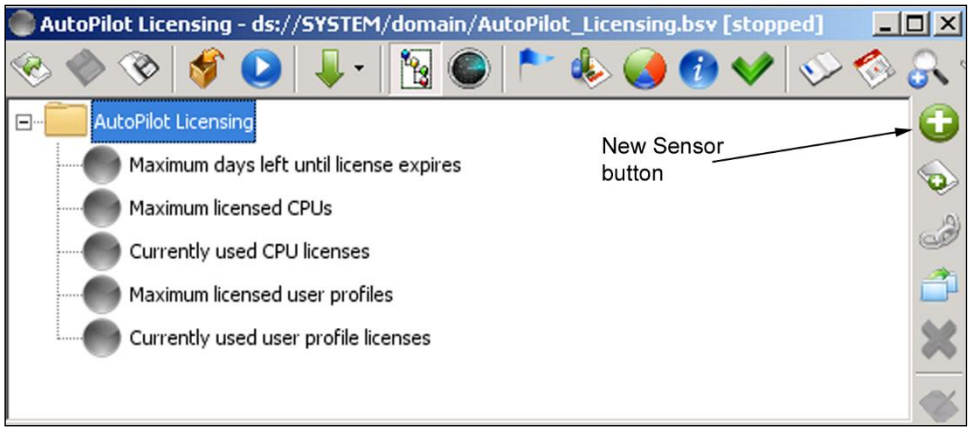


Figure 4-71. Add a New Sensor

2. Create a new sensor by clicking on the New Sensor button. The General Sensor Options screen is displayed.

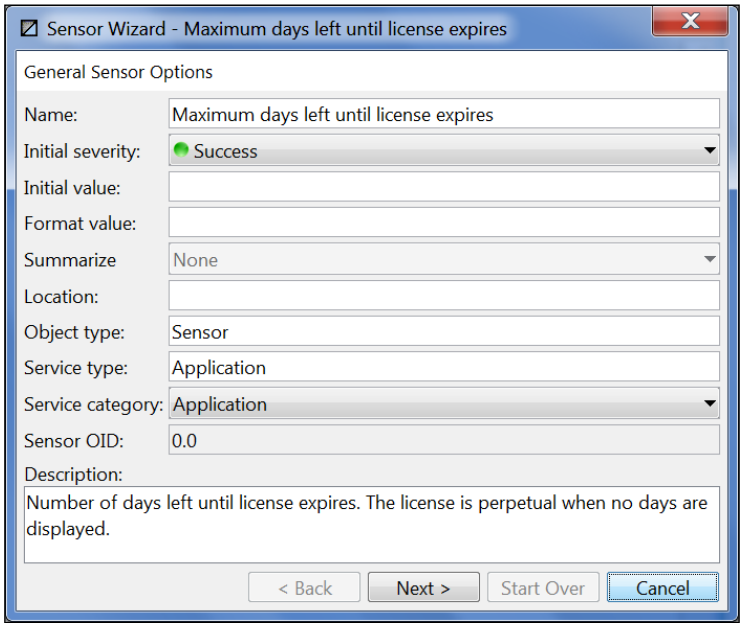


Figure 4-72. Create a New Sensor - General Screen

3. Assign a logical sensor name in the *Name* field. The default is Untitled.
4. Identify the initial severity in the *Initial severity* field. The default is Unknown.
5. Add an initial value in the *Initial value* field. An initial sensor value is assigned when the sensor is initialized or the fact is cleared. The initial value can be set for every sensor. Dynamic sensors inherit the initial value from the parent sensor. The default initial value is always set to null, unless modified by the user. Sensors will show the most recent value when the connection to the source or facts drops.

5a. Specify a numerical format, for example ###,###.##.

> **NOTE** The *Summarize* field only applies to dynamic sensors. Refer to section 4.9.3, *Dynamic Sensors*, for an explanation of this field.

6. Add the specific name of the resource in the *Object type* field, for example: Queue Manager.
7. Add name or identity of the monitored service in the *Service type* field, for example: WebSphere MQ
8. Add the classification of the monitored service in the *Service category* field, for example: Server.
9. Provide a description of the sensor functionality and purpose.

   If adding a hyperlink tag in the description field, do not use double quotes around the URL link.

   For example, **use**: <A HREF=http://www.nastel.com>Nastel</A>

   Do not use: <A HREF="http://www.nastel.com">Nastel</A>
10. Click **Next** to display the *Select Facts to Monitor* screen.
11. Select the desired fact; click **Include** to add the fact to the *Monitor these facts* list.
12. To specifically exclude a fact from monitoring, select the fact, and then click **Exclude**. The fact will be added to the *But not these facts* list.
13. If you want to convert hard coded names into names with environment variables, ensure that the *Auto parameterize* check box is selected and then click **Next**. The *Sensor Evaluation Rules* screen (Figure 4-74) will be displayed.

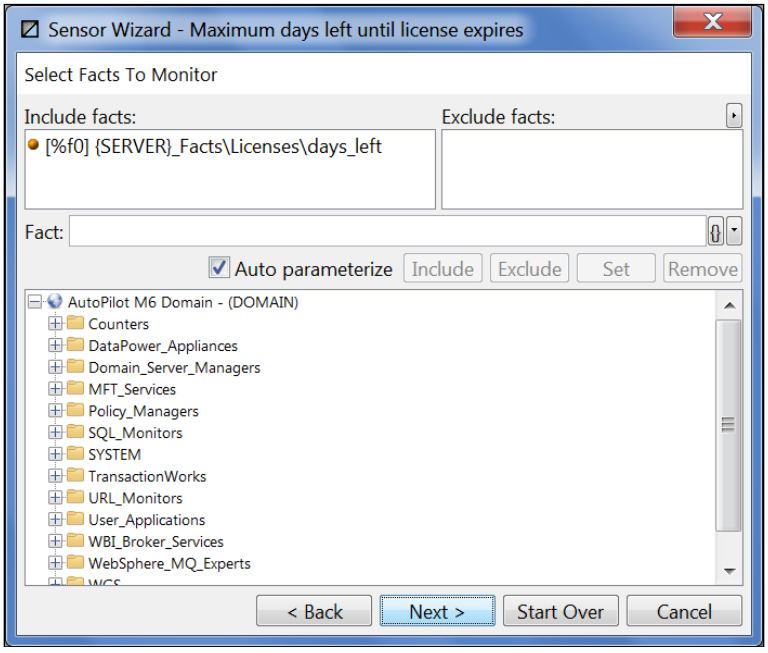


Figure 4-73. Selecting Facts to be Monitored

14. When you have added or excluded all required facts click **Next** to proceed.

To improve sensor performance, the following type qualifiers can be added to values used within the sensors.

Table 4-30. Sensor Performance

Type Qualifier	Description
$int.	integer (example - $int.100)
$long.	long (example - $long.100023)
$float.	float
$double.	double
$bool.	Boolean
$date.	date/time object
$time.	date/time object
$char.	string
$systime.	system time in milliseconds

Supported Operators: The set of operators listed below can be used while defining the states of the sensor. These can be typed in or selected by clicking on the buttons shown below.

- **+** *Adds* the two operands on either side of the operator
- **-** *Subtracts* the operand on the right from left
- ***** *Generates* a product of the two operands
- **/** *Divides* the operand on the left by the operand on the right.
- **%** Modulus operator - *divides* the operand on the left by the right and produces the remainder.

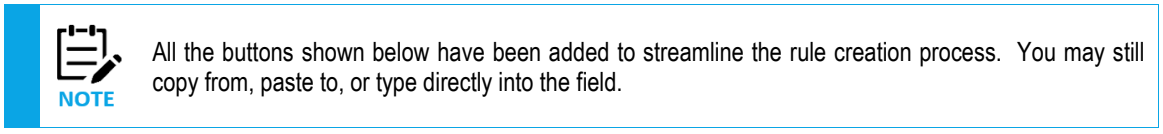
Click **Facts**  icon to insert the first fact in the list of those selected. Clicking on the newly inserted fact gives you the ability to select a different fact. Apart from the fact value you may also use other meta-data such as Last-Changed, Last-Updated, Length, Max, Mean, Min, etc. in your expression.

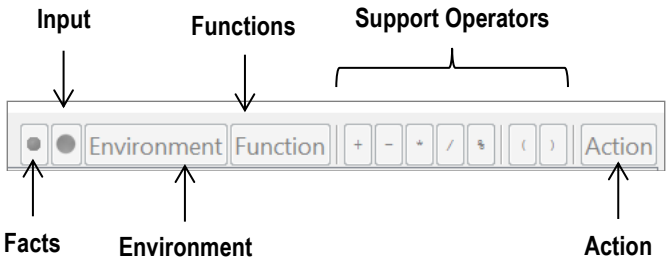
Click **Input**  icon to insert the numerical severity of the child sensor. (Useful only for static child sensors.)

Click **Environment** to insert a reference to an environment variable '`user.name`'. Clicking this variable makes a menu available for selecting other variables defined in the business view or globally in the different property files used by M6.

Click **Function** to insert a reference to a function variable 'abs( )'. Clicking on this variable displays a menu for selecting other functions as defined in the business view or globally in the different property files used by M6.

Tab to the *Action* field and click **Action** button to insert a default action '*beep*'. Clicking the newly inserted action allows you to select a fact and its first action as defined on the fact; you may now select different actions defined on the fact by clicking on the action displayed. You may also type in any action that you want executed if the sensor state gets set to the corresponding severity.





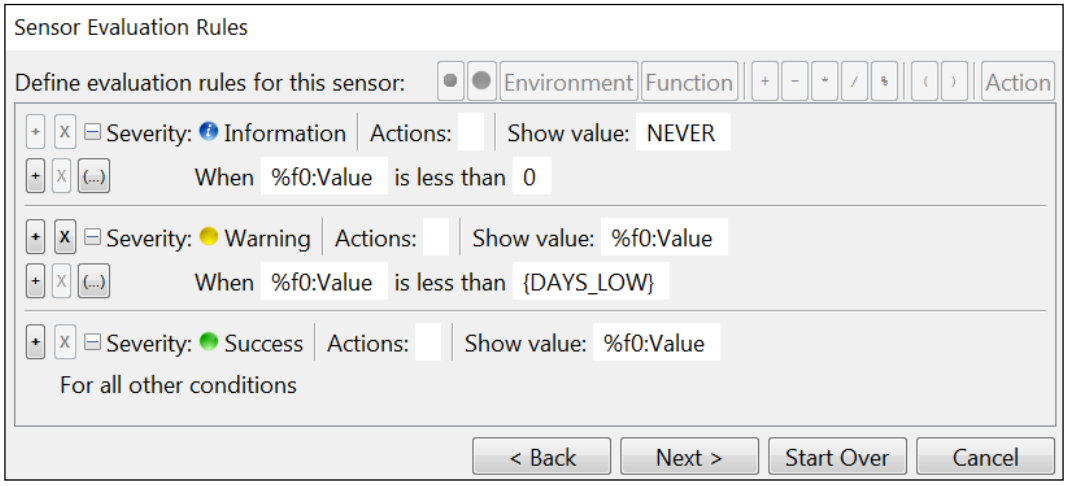


Figure 4-74. Sensor Rules

Rules

15. The + icon adds an additional evaluation rule for determining sensor state and the X icon deletes the rule. The (...) icon allows the user to create nested "and/or" evaluation rules.
16. Select the fact to be evaluated from the facts included. All facts selected for use in this sensor will be listed in the menu. Facts are numbered (prefixed) as **[%f0]**, **[%f1]** etc. Click the fact button on the upper left-hand side of the sensor screen to insert the first fact %f0. Metrics are derived from facts. There are two types of metrics: "out-of-the-box" and those based on history.

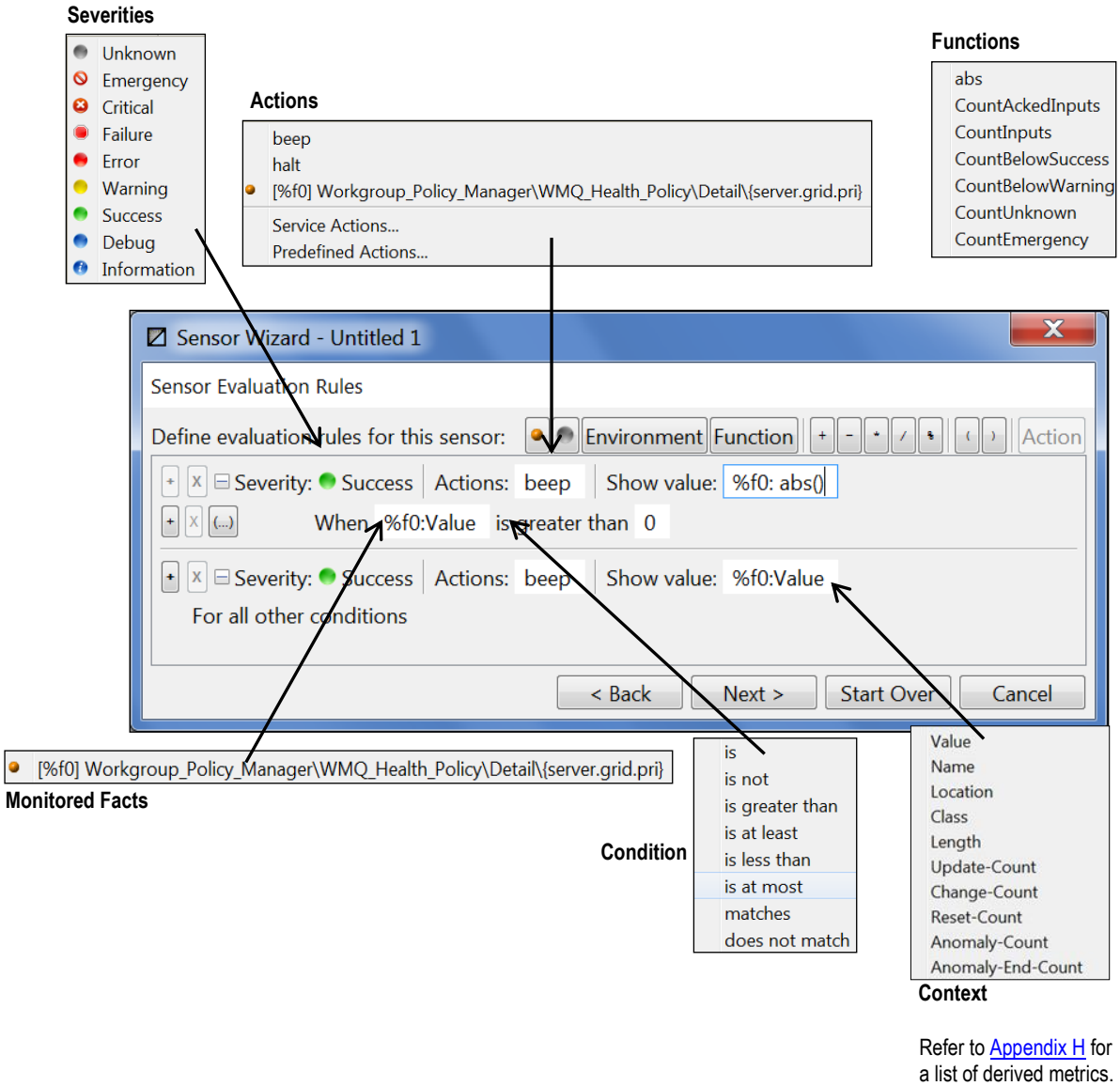


Figure 4-75. Sensor Rule Inputs

17. Select the *conditional operator* to be used. The conditional operator is applied to the operands (value evaluated in the fields) on the left and right, when the expression evaluates to true, the sensor is set to the specified severity.
    - Use "greater than" or "less than" to determine if a fact value is within normal operational parameters.
    - Use "matches" and "does not match" for comparison of string content.
      a) Use %f0:Value matches AAA, to match the value if AAA is contained anywhere within.

b) The placeholders of * and ? can be used where * matches any number of characters and ? matches any single character.

For example:

%f0:Value matches A*B?D matches anything that begins with A followed by any number of characters followed by B and then any character followed by D.

ABCD

ATTTTTTTTTTBCD

Does not match is the opposite, it triggers if the object does not contain the matching text.

18. The value in the field selected may be evaluated against any value for the selected condition.
19. Select the severity level appropriate for the conditions being evaluated. Severity levels that reflect stoppages or failures can, and usually do, send notifications alerting the relevant personal to the condition. In addition, the severity can trigger corrective actions to remedy the failure or diminished service.

> **NOTE** The default severity 'Unknown" is reserved and must be changed to an active severity level. If it remains unchanged you will not be able to proceed to the next screen.

20. Select an *action* that is appropriate for the severity reached. There can be multiple conditions and variables established for each fact monitored by using multiple rules, scripts, or user-defined commands. Hence there can be multiple severities and actions to support the needs of the sensor. Actions are triggered by the severity level reached. Some conditions will warrant no action, others will require multiple actions. The actions listed in the menu are numbered to reflect the fact number assigned when you include the fact(s) in the sensors. The pre-defined actions are repeated for each fact listed. The pre-defined actions listed are beep, halt and the list of facts included in the sensor.

    You can specify an external command, shell script, or executable (example: *runaction.bat*). You may also pass business view context information that could be used within a script or action.

    Users can manually enter actions by entering an assigned fact number (example: `%f0.Start-Service`) to initiate the specified action against the designated fact. See 4.9.8 Setting User Actions for details on creating user actions and pre-defined actions.

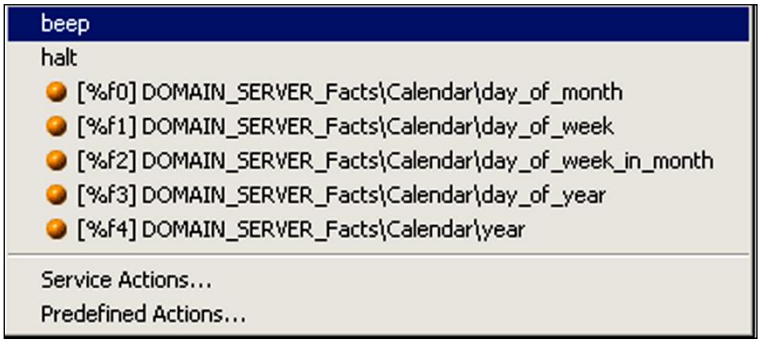


Figure 4-76. Actions

21. In addition, the *User Action* field on the **Define Alerts and Actions** tab can be used to specify an external command, shell script, or executable with AP environmental variables used as parameters of the command. (See Table 4-31, AutoPilot System Environment Variables).

    Click on the down arrow at the right end of the **User action** field to choose action choices as shown in the figure below.

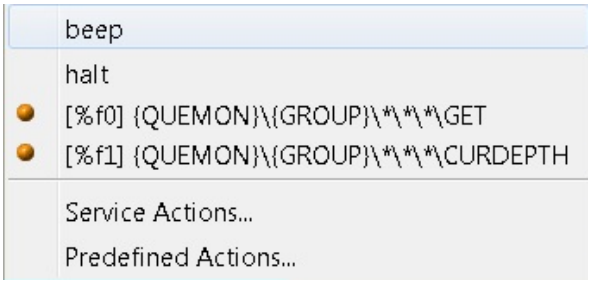


Figure 4-76-b. Sensor User Action Choices

They include:

- a simple beep
- halt
- a list of the sensor's facts: one more of the facts can be selected and inserted in the action as command parameters
- service actions: Built-in actions for each expert and policy manager, such as send an SNMP V2 trap, see figure below.

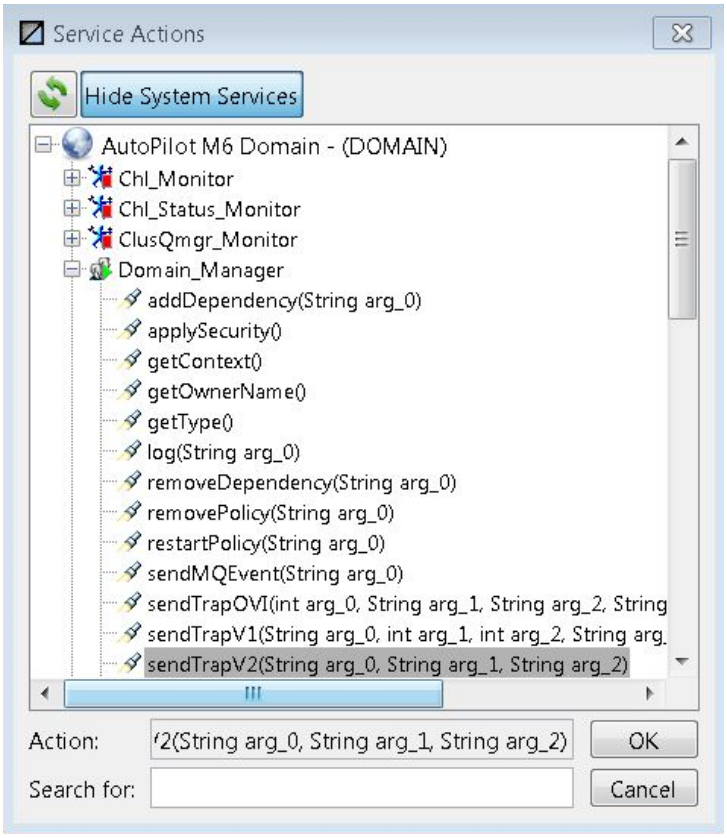


Figure 4-76-c. Service Actions

- pre-defined actions: Actions that users can define, such as a command to list the files in a directory, see figure below.

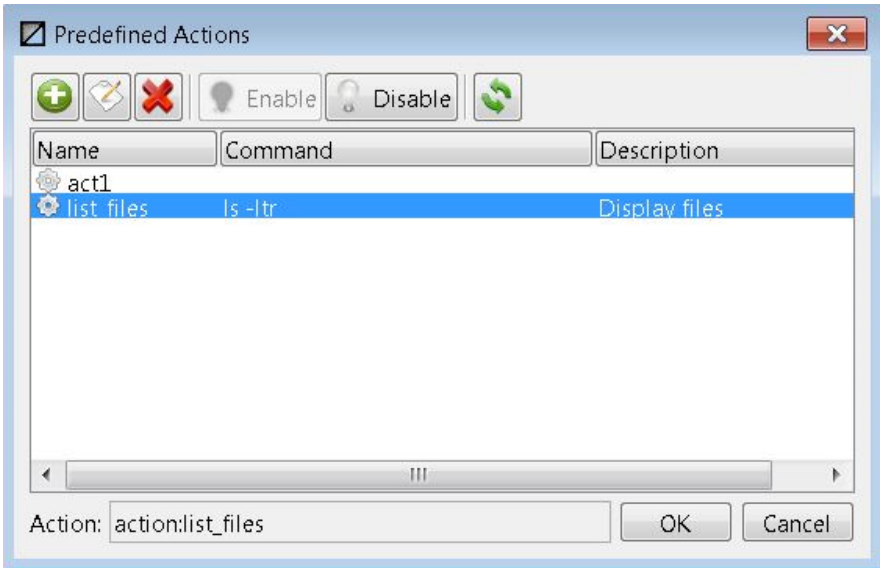


Figure 4-76-d. Predefined Actions

(Example: `myscript.bat %sev% %desc% %facts% %event%` etc.). The User Action is usually taken in response to a problem or event.

If you want to receive an e-mail notification of a specific event or condition, you can have the event, along with other status or properties of the sensor, included. Add action parameters to the *Action* to go with the script. To change the e-mail behavior, add the following properties to the `[AUTOPILOT_HOME]/global.properties` file:

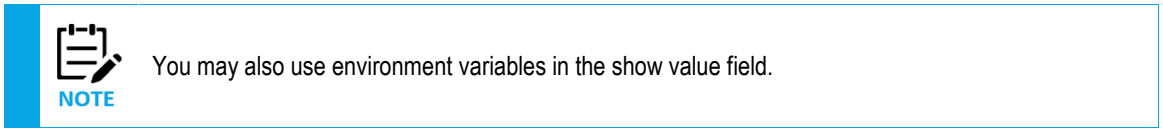
- `sensor.default.email.senderbehavior=group`
  Can be set to one of the following:
  - **group:** All email recipients are shown (all email addresses in the "To" field of the email). For example, if the email is sent to three different email addresses, each recipient will see the other two recipients.
  - **individual:** Each recipient will only see themselves instead of the entire group.
- `sensor.default.email.smtp.connectiontimeout=15000`
  This is the socket connection timeout value in milliseconds.
- `sensor.default.email.smtp.timeout=15000`
  Socket read timeout value in milliseconds.

Table 4-31. AutoPilot System Environment Table Variables

Variable	Description
%account%	Name of current user context. Used within event mask and sensor action fields.
%cause()	Is an action that gets variable data from a child sensor that triggered the parent to go into alarm. Example: Root Bob Sally Joe Chris If the parent, Bob, wants to generate the event, "My child has gone into alarm" but wants to include the child name it could be done such as "My child %cause(%from%) has gone into alarm". Alternately, there is also parent() too, that is the children want to send the event but reference their parent. "I have gone into alarm, please notify %parent(%from%)"

%date%	Date stamp of the event: **Month, Date, Year**
%desc%	Description of the firing sensor defined in the wizard or properties of the sensor.
%event%	Event message that qualifies the sensor. The message includes the same information that would be logged or viewable in an event viewer
%facts%	Facts and their values in the form: fact1=value1, fact2=value2,factN=value. The %facts% may need to be enclosed in "" quotes, since it may contain blanks or special characters.
%from%	Component that generated the event (sensor name in this case).
%f#(token)%	Where %f# refers to the fact being monitored by a sensor and # is the fact index number, token is a zero-based token of the fact delimited by "\" delimiter. Example: Given %f0=Expert\TK1\TK2\TK3\TK3\Var Token1: TK1 as %f0(1)%, expert name: %f0(0)%, Token2: %f0(2)% Example: action %sevstr% token2="%f0(2)%"
%health%	Sensor health index from 0.0 to 1.0
%id%	Message id of the sensor, as defined in the "Alert Id" field of the sensor "Alert" properties.
%ivalue%	Value of the child sensor that created the alert.
%parent%	Name of the immediate parent sensor.
%party%	List of email addresses parties as specified in *Email To* in the Alert Options.
%related%	Event message that triggered the sensor. May need to be enclosed in ""quotes, since it may contain blanks or special characters.
%root%	Name of the root sensor (top-most sensor) of the business view.
%sev%	Severity of sensor that fires the action; integer number from 0-8 representing sensor severities.
%sevstr%	Upper case string representation of %sev% integer code. Ex: WARNING, CRITICAL, SUCCESS.
%srvcaty%	Integer value of service category. Refer to Table 4-28, Service Category.
%srvtype%	String value of service type.
%objtype%	String value of object type.
%time%	Time stamp of the event: **HH:MM:SS AM/PM.**
%user%	Name of the user as specified in *From User* in the *Alert Options.*
%value%	Current sensor value as specified in the sensor wizard.
{env_var}	Where **env_var** is a java environment property. The complete list of properties varies from system to system. Please refer to http://www.javasoft.com/ and search "java properties".
%ovosev%	Maps M6 severity to HPOVO severity.
%tecsev%	Maps M6 severity to Tivoli TEC severity.

22. *Show Value* field defines the value for a sensor. This can be generated using the value of a fact directly or by evaluating an expression.



23. Click **Next** and set properties for your *sensor alert* using options outlined in table and figure below:

Table 4-32. Alert Properties	
Parameter	**Description**
Severities:	Select the severity levels that will trigger notifications: Unknown: 0.0 (0%), default alert setting Emergency: 0.1 (10%) Critical: 0.25 (25%) Failure: 0.45 (45%) Error: 0.65 (65%) Warning: 0.75 (75%) Success: 1.0 (100%) Debug: 1.0 (100%) Information: 1.0 (100 %)
Suppress false alarms	Enable/Disable ☑/☐ false alarm. Intelligent False Alarm suppression logic. Disables duplicate actions/notifications when sensors are acknowledged/armed.
Use Defaults button	Click **Use Defaults** to use the default properties that were defined in domain.properties on each domain server installation. Click **Yes** in confirmation box.
From User	Enter your mail server user id. Usually is the same as your email address.
Mail Server (SMTP)	Enter the name of the local mail server (SMTP server).
E-mail to	Enter the email address of the party to be notified (e.g., %f0:user@nastel.com).
E-mail subject	Enter user-defined e-mail subject line.
E-mail MIME type	Select text/plain or text/html e-mail
Send Test Mail button	Click button to verify e-mail reached the intended recipient.
User Action:	Use to specify an external command, shell script or executable: (e.g., *runaction.bat*). You may also pass business view context information that could be used within a script or action. Refer to section 4.9.4, Automated Actions for a complete list of supported action variables.
Event ID	User assigned event identification number. You can customize the number as needed, but only using numbers, no spaces (Example: default = 3010. Custom = 4089). This ID is used to record events into M6 event log.
Event mask	Event format mask used to record events triggered by the sensor. Users may customize the event format. Refer to section 4.9.4, Automated Actions for a complete list of supported variables that can be used within the event mask.
Scheme	Open the Scheme menu and select the specific occurrence that will trigger notifications.
	None: Notification will not be generated.
	Once: Notification generated once per severity status change all severities selected in *Alert On* section.
	Change: Notification generated with each change in status.
	Repeat: Sensor repeats actions/alerts until the senor goes into a state, which is not checked in the "Alert On" section. Sensor re-arms until the sensor goes back to an unchecked state. Notification frequency is based on Re-arm Delay intervals.
Trigger Delay (ms)	User defined delay times expressed in milliseconds (1/1000th of a second). Default is 60000, or 60 seconds. The sensor will delay notification for the prescribed time.
Repeat Limit	Maximum action/notification repetitions.
Re-Arm Delay (ms)	User defined re-arming delay of sensor notifications. Sensor will resample and resend alert notification continuously (per delay time) until condition is corrected. Default is 60000, or 60 seconds.

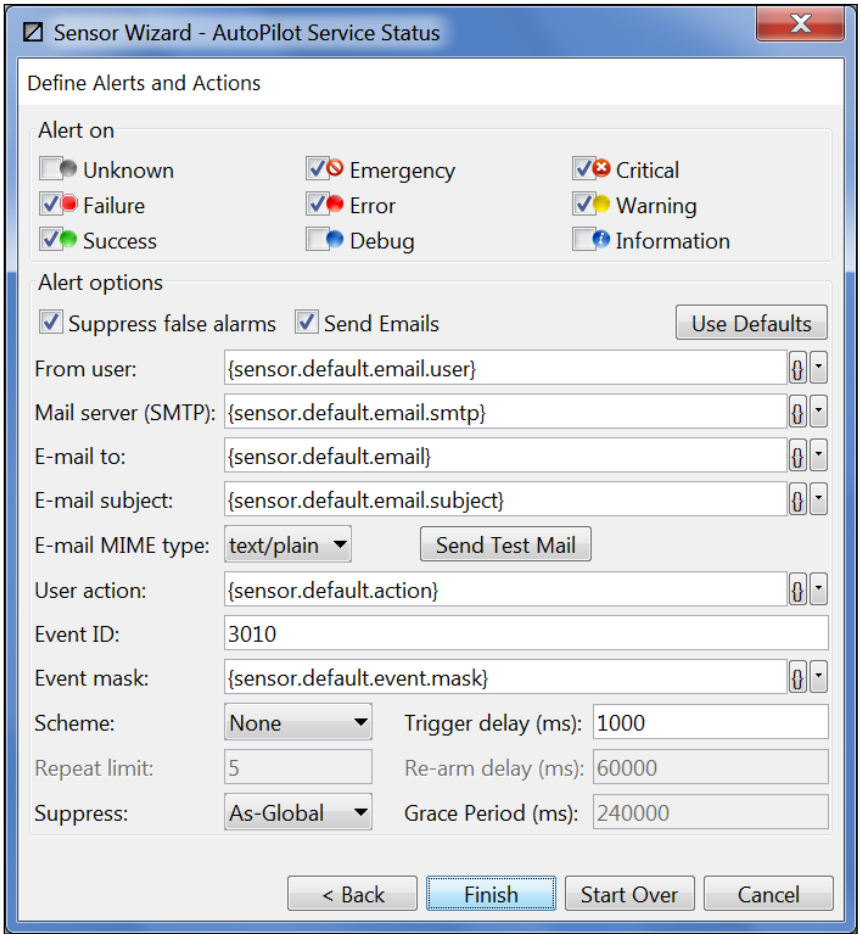


Figure 4-77. Alert Properties

24. Click **Finish** to complete the sensor.

4.9.3 Dynamic Sensors

The dynamic sensor rules are defined and performed much like other sensors. The difference is those dynamic sensors are typically used to monitor the same fact used in multiple locations within the domain. For example, they can be used to monitor the status of multiple nodes in an M6 domain. To monitor the nodes state you would have to verify that the fact reporting the state for all nodes was identical in name and function. By using a wildcard (* asterisk) in place of the node name in the address, you would include the state of every node in the domain. Once the sensor wizard recognizes the wildcard in an included or set fact, it identifies the sensor as dynamic.

For example: If the normal condition is all nodes are Active and a status of *Success* was defined for this state, then status would remain unchanged as long as all nodes were active. If one or more nodes stop, then the status would change to *Warning* or another condition defined in the rule.

The change in status would change the existing severity. The change could trigger a pre-determined action to attempt to correct or report the change. When the condition returns to normal conditions the status is updated accordingly.

Notifications defined to alert you if a condition or node changes status would only alert you that there was a change, but not which node changed.

The forward slash **"/"** character is a reserved symbol and cannot be used as part of any name in M6.

Creating a Dynamic Sensor:

1. Open the business view to receive the dynamic sensor.

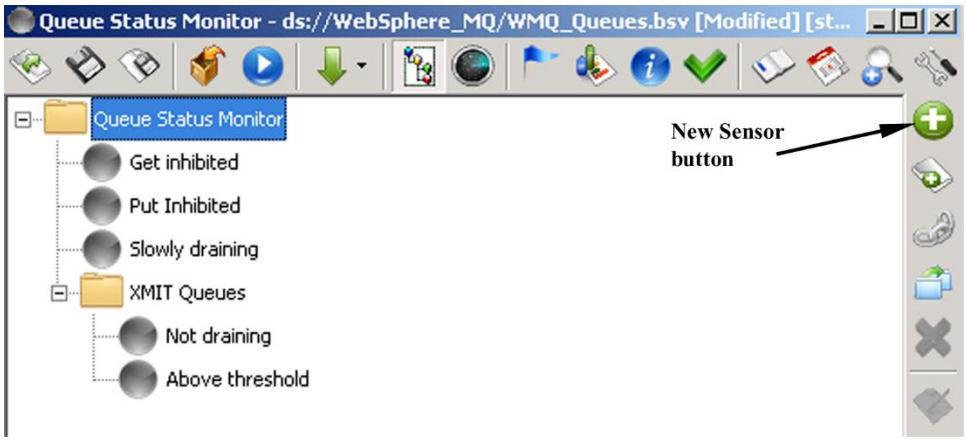


Figure 4-78. Adding New Dynamic Sensors

2. Create a new dynamic sensor by clicking on the **New Sensor** button. The *General Sensor Options* screen is displayed.

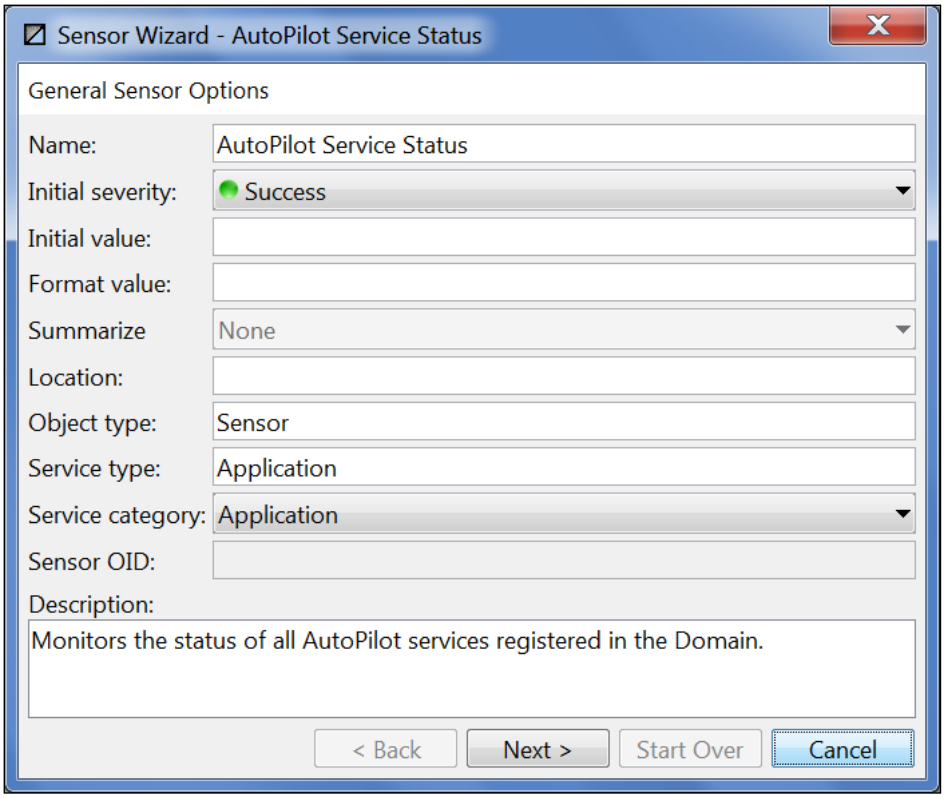


Figure 4-79. Sensor Name and Description

3. Assign a logical sensor name in the *Name* field. The default is Untitled.
4. Identify the initial severity in the *Initial severity* field. The default is Unknown.
5. Add an initial value in the *Initial value* field. An initial sensor value is assigned when the sensor is initialized or when the fact is cleared. The initial value can be set for every sensor. Dynamic sensors inherit the initial value from the parent sensor. The default initial value is always set to null, unless

modified by the user. Sensors will show the most recent value when the connection to the source or facts drops.

6. The Summarize field is always disabled and will display the default value None. If a summarization method is displayed on the *Dynamic Sensors Options* screen (Step 18 below), this field will automatically change to display the summarization selected.
7. Add the specific name of the resource in the *Object type* field, for example: Queue Manager.
8. Add name or identity of the monitored service in the *Service type* field, for example: WebSphere MQ.
9. Add the classification of the monitored service in the *Service category* field, for example: Server (refer to Table 4-28).
10. Provide a description of the sensor functionality and purpose.
11. Click **Next** to display the *Select Facts to Monitor* screen.

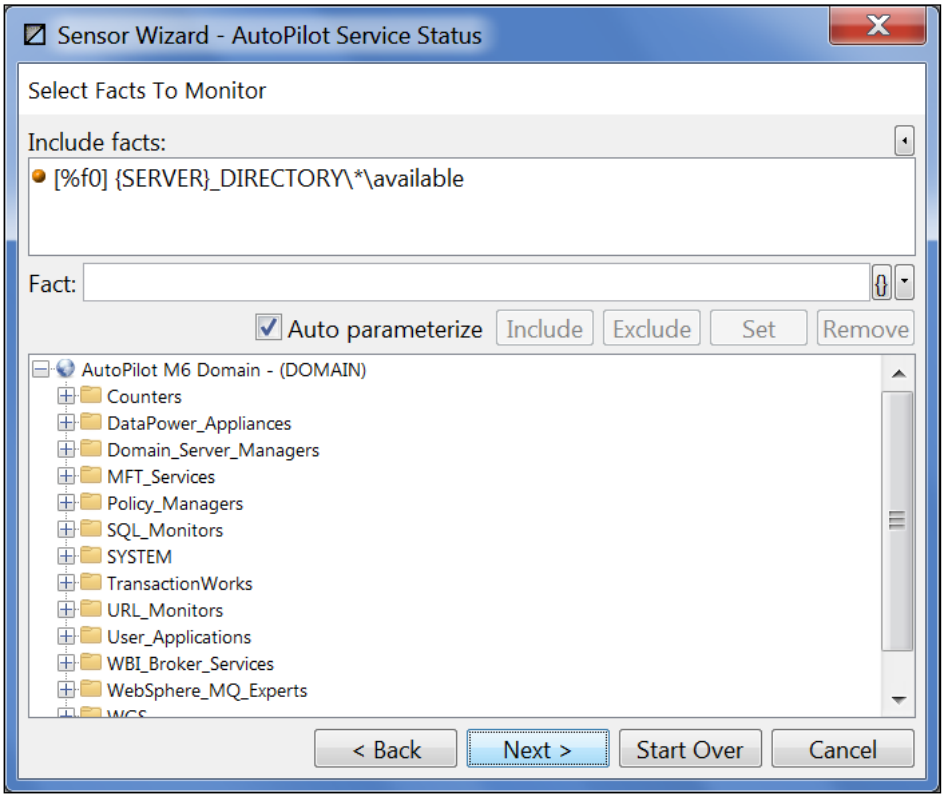


Figure 4-80. Sensor Wizard Fact Selection

12. Select the sensors that will be used to create the dynamic sensor.
13. Edit the sensor address by replacing the node name and any subsequent names (queue manager, queue, etc.) with a wild card (asterisk *). Using the wildcard will allow the sensor to be defined as a dynamic sensor and will monitor the state of all the nodes. Regexp can also be used (format %<regexp>).

    Example: the sensors selected were `Que_Monitor\Workgroup\Node_Name]\Queue_Manager_Name]\[Queue_Name]\[fact].`

    The node name, queue manager name and queue name were each replaced with an * in order to monitor all nodes, queue managers and queues under Que_Monitor for current depth and usage. If

the facts edited to include the wildcard are going to also be used for a child sensor, then you have to include these facts again to make them available.

Example for regexp:

```
OS_Monitor\%<A|B|C|>\value
```

Invalid regexp commands:

```
OS_Monitor\some text%<A|B|C|>\value
OS_Monitor\%<A|B|C|>some letters\value
```

NOTE The **Fact** field has search capability. Enter at least three characters and press **ctrl+space**. A drop-down list is displayed. Select a fact and click **Include** to include the selected fact.

14. Click **Include** to include the selected fact(s).
15. If you want to convert hard coded names into names with environment variables, ensure that the *Auto parameterize* check box is selected. Click **Next** to display the *Create Dynamic Sensors* screen.

Supported Operators: The set of operators listed below can be used when you define the states of the sensor. These can be typed in or selected by clicking on the buttons shown below

Supported Operators

- **+** *Adds* the two operands on either side of the operator
- **-** *Subtracts* the operand on the right from left
- ***** *Generates* a product of the two operands
- **/** *Divides* the operand on the left by the operand on the right.
- **%** Modulus operator *divides* the operand on the left by the right and produces the remainder.

Click the **Facts** icon to insert the first fact in the list of those selected. Clicking on the newly inserted fact gives you the ability to select a different fact. Apart from the fact value you may also use other meta-data such as Last-Changed, Last-Updated, Length, Max, Mean, Min, etc. in your expression.

Click **Input** icon to insert the numerical severity of the child sensor. (Useful only for static child sensors.)

Click **Environment** to insert a reference to an environment variable '`user.name`'. Clicking this variable makes a menu available for selecting other variables defined in the business view or globally in the different property files used by M6.

Click **Function** to insert a reference to a function variable 'abs( )'. Clicking on this variable displays a menu for selecting other functions as defined in the business view or globally in the different property files used by M6.

Tab to the *Action* field and click the **Action** button to insert a default action 'beep'. Clicking on the newly inserted action allows you to select a fact and its first action as defined on the fact. You may now select different actions defined on the fact by clicking on the action displayed. You may also type in any action that you want executed if the sensor state gets set to the corresponding severity.

Select **Dynamic folders** to dynamically create folders in business views; for example, a folder named after a Queue Manager and containing alerts for all the channels belonging to the Queue Manager. This option is only for dynamic sensors.

NOTE All the buttons shown below have been added to streamline the rule creation process. You may still copy from, paste to, or type directly into the field.

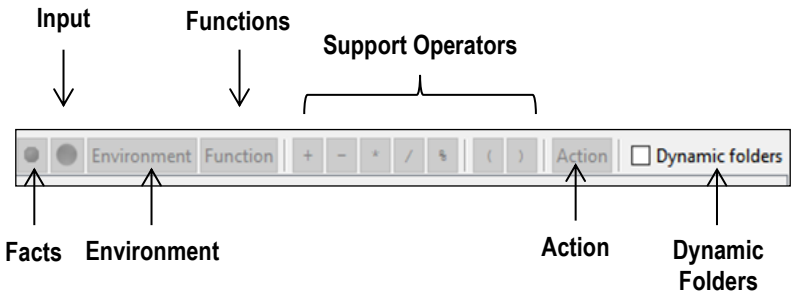


Figure 4-81. Supported Operators

NOTE

The Action support operator is not active for this screen. Actions are only applicable to the child sensors when using dynamic sensors.

16. Fact\context\conditions\variables:
    a. When: Select the dynamic fact you modified on the previous screen. The facts are numbered in the order they are included (example: `[%f6]`**)**.
    b. The context should reflect the nature of the fact and the intent of the sensor. *Value* is the default. In this sensor the default is used.
    c. Select the condition option to be used. The conditions give you a variable on which to base the sensor conditions. In this sample dynamic sensor we used *is not*.
    d. Enter the user-defined value: fact/status/value, to finish defining the conditions. This sensor is monitoring the status, *Active* or *Stopped* etc. We want the sensor to report when any of the services are not active. You can use the operator to customize the variable to suit specific needs.

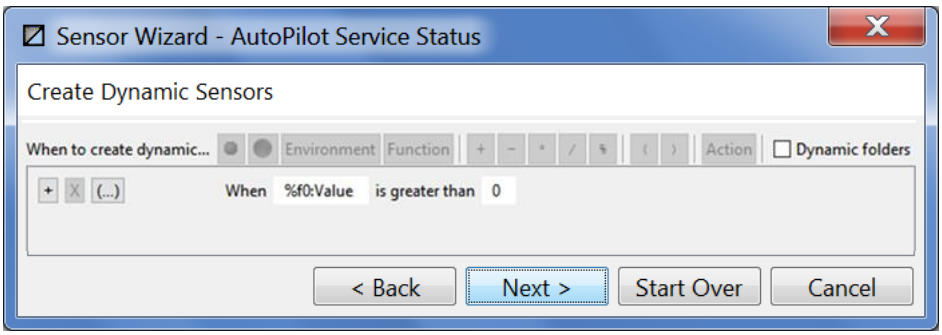


Figure 4-82. Creating Dynamic Sensors

17. Click **Next** to proceed to dynamic sensor options.

18. Define the sensor name and type as defined in the table below.

Table 4-33. Dynamic Sensor Properties

Category	Property	Description
Dynamic sensor name	Use fact name	Uses the whole fact name as the Sensor name.
	Use part of fact name	Uses the first or user-defined segment of the fact name as the sensor name. Use directional arrows < > to move name to the desired segment.
	User defined name	User defined logical sensor name. Facts name can be tokenized using %#, where # is a zero-based token number. Example: Service %1 (recommended) See example and figure. Environment variables and system variables such as %sevstr% and %date% can be included as part of the user-defined name.
Dynamic sensor options	Summarize	Select how to summarize child sensors by selecting one of the following: Count; Count Acks; Max Value; Min Value; Average; Sum or Deviation. Only available at the dynamic sensor level. Default value is None.
	Expire dynamic sensor	If checked, enables expiration of the sensor as indicated by user in the seconds field. The sensor will disappear at a user-defined time limit, after reaching 100% health. The default is after 20 seconds, but it can be set to any duration needed. This is useful when corrected conditions need to be removed from the business view (recommended).

The dynamic sensor creates child sensors that monitor the facts at all the locations assumed in the wildcard. By maintaining them as a permanent sensor, they will log to the file or database as specified in the logging properties.

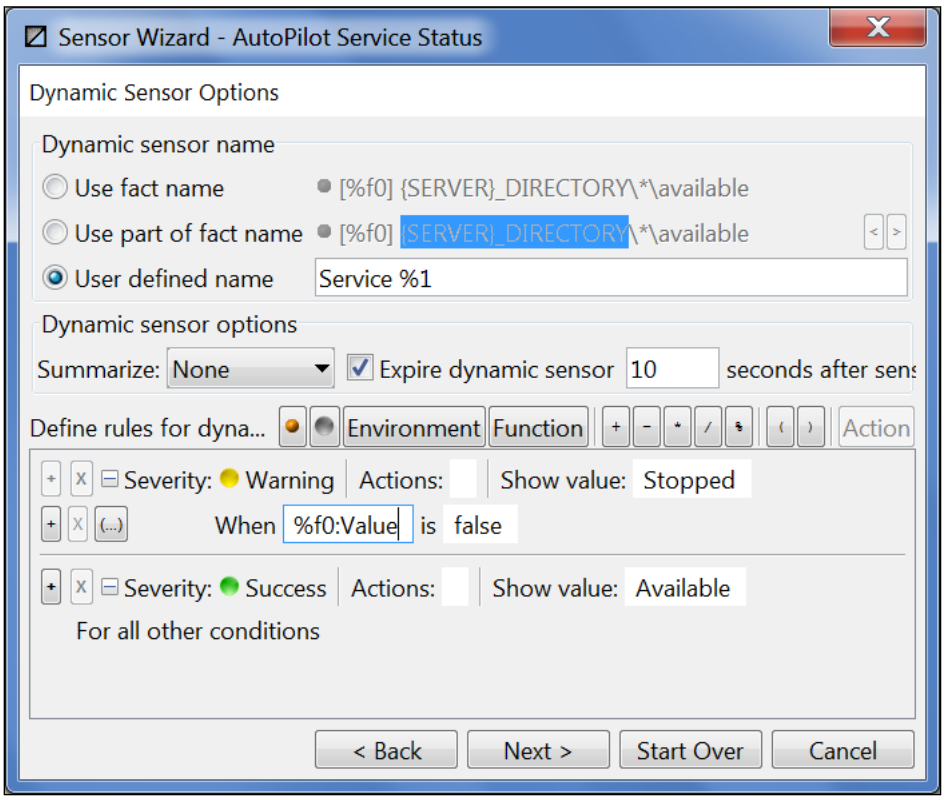


Figure 4-83. Dynamic Sensor Options

Dynamic Sensor Conditions: Define the sensor rules

19. The [+] icon adds an additional evaluation rule for determining sensor state and th [X] icon deletes the rule. The [(...)] icon allows the user to create nested "and/or" evaluation rules.
20. Select the fact to be evaluated from the facts included. All facts selected for use in this sensor will be listed in the menu. Facts are numbered (prefixed) as [%f0], [%f1] etc. Click the fact button on the upper left-hand side of the sensor screen to insert the first fact %f0. Metrics are derived from facts. There are two types of metrics: "out-of-the-box" and those based on history.

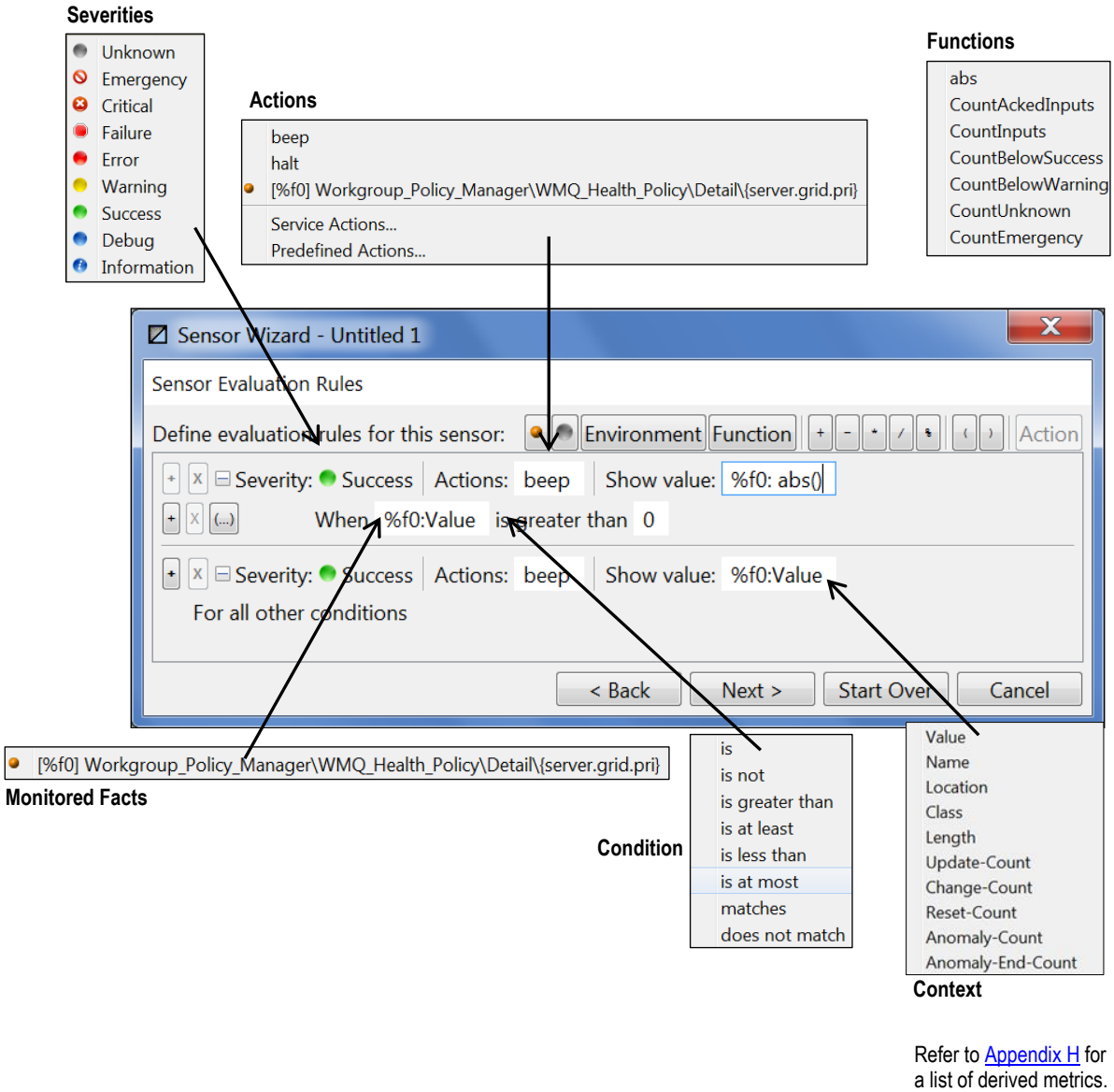


Figure 4-84. Sensor Rule Inputs

21. Select the *conditional operator* to be used. The conditional operator is applied to the operands (value evaluated in the fields) on the left and right, when the expression evaluates to true, the sensor is set to the specified severity. For example: using "greater than" or "less than" to determine if a fact value is within normal operational parameters.

    The value in the field selected may be evaluated against any value for the selected condition.
22. Select the severity level appropriate for the conditions being evaluated. Severity levels that reflect stoppages or failures can, and usually do, send notifications alerting the relevant personal to the

condition. In addition, the severity can trigger corrective actions to remedy the failure or diminished service.

> **NOTE** The default severity 'Unknown" is reserved and must be changed to an active severity level. If it remains unchanged you will not be able to proceed to the next screen.

23. Select an *action* that is appropriate for the severity reached. There can be multiple conditions and variables established for each fact monitored by using multiple rules or using scripts, user define commands. Hence there can be multiple severities and actions to support the needs of the sensor. Actions are triggered by the severity level reached. Some conditions will warrant no action, others will require multiple actions. The actions listed in the menu are numbered to reflect the fact number assigned when you include the fact(s) in the sensors. The pre-defined actions are repeated for each fact listed. The pre-defined actions listed are beep, halt and the list of facts included in the sensor, service actions and predefined actions. See similar section 4.9.2.2 Creating Sensors with the Wizard (Non-Dynamic) , step 21, for definitions and figures.
You can specify an external command, shell script or executable (e.g. *runaction.bat*). You may also pass business view context information that could be used within a script or action.

    Optionally, users can manually enter actions by entering the assigned fact number to initiate the specified action against the designated fact.

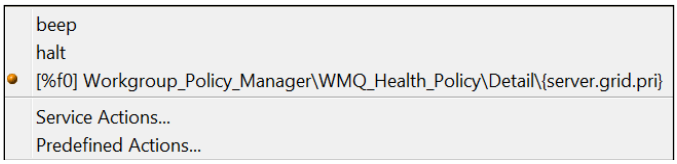


Figure 4-85. Actions

In addition, the *User action* field on the **Define Alerts and Actions** tab (see figure below) can be used to specify an external command, shell script or executable (for example: `myscript.bat %sev% %desc% %facts% %event% etc.`) (see table 4-31 AutoPilot System Environment Variables). The *User Action* is usually taken in response to a problem or event. If you want to receive e-mail notifying you of a specific event or condition, you can have the event, along with other status or properties of the sensor included by adding action parameters to the *Action* to go with the script.
See 4.9.8 Setting User Actions for details on creating user actions and pre-defined actions.

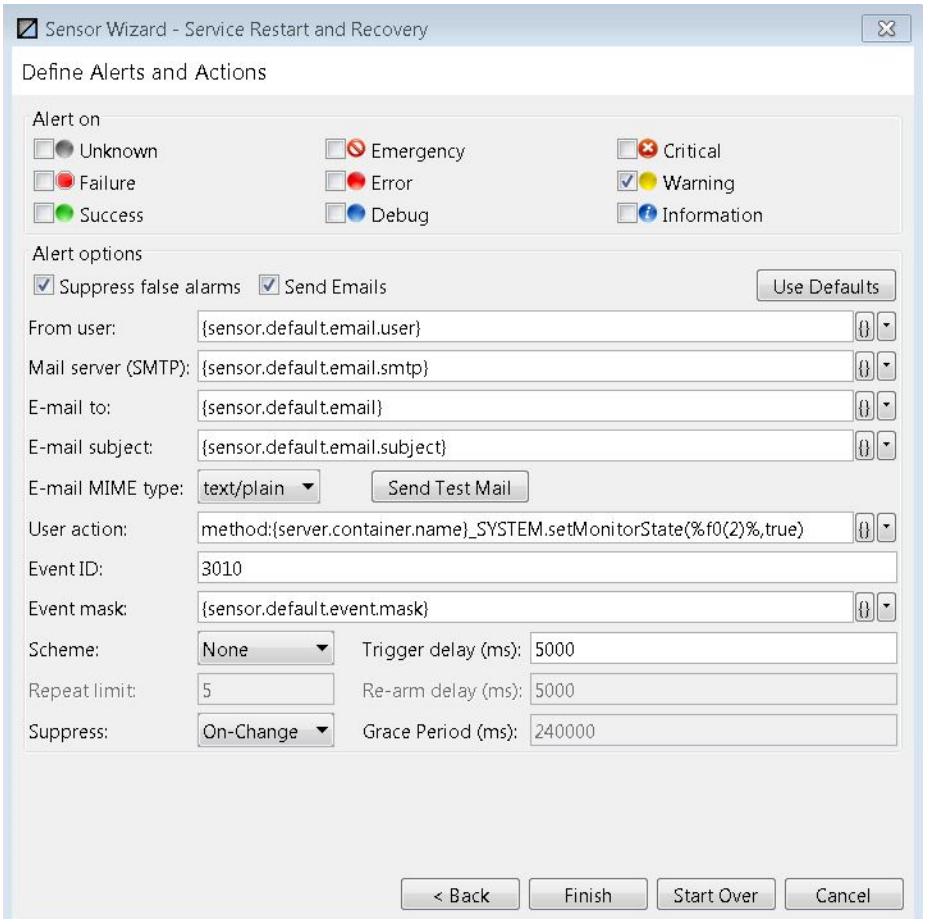


Figure 4-85-b. Policy CEP Server Health Sensor Wizard – Define Alerts and Actions

Table 4-34. AutoPilot System Environment Variables	
Variable	**Description**
%account%	Name of current user context. Used within event mask and sensor action fields.
%cause()	Is an action that gets variable data from a child sensor that triggered the parent to go into alarm. Example: Root Bob Sally Joe Chris If the parent, Bob, wants to generate the event, "My child has gone into alarm" but wants to include the child name it could be done such as "My child %cause(%from%) has gone into alarm". Alternately, there is also parent() too, that is the children want to send the event but reference their parent. "I have gone into alarm, please notify %parent(%from%)"
%date%	Date stamp of the event: **Month, Day, Year**
%desc%	Description of the firing sensor defined in the wizard or properties of the sensor.
%event%	Event message that qualifies the sensor. The message includes the same information that would be logged or viewable in an event viewer.
%facts%	Facts and their values in the form: fact1=value1,fact2=value2,factN=value. The %facts% may need to be enclosed in "" quotes, since it may contain blanks or special characters.
%from%	Component that generated the event (sensor name in this case).
%f#(token)%	Where %f# refers to the fact being monitored by a sensor and # is the fact index number, token is a zero-based token of the fact delimited by "\" delimiter. Example: Given %f0=Expert\TK1\TK2\TK3\TK3\Var Token1: TK1 as %f0(1)%, expert name: %f0(0)%, Token2: %f0(2)% Example: action %sevstr% token2="%f0(2)%"
%health%	Sensor health index from 0.0 to 1.0
%id%	Message id of the sensor, as defined in the *Alert Id* field of the sensor *Alert* properties.
%parent%	Name of the immediate parent sensor.
%party%	List of email addresses parties as specified in the *Email To* field in the *Alert* properties.
%related%	Event message that triggered the sensor. May need to be enclosed in ""quotes, since it may contain blanks or special characters.
%root%	Name of the root sensor (top-most sensor) of the business view.
%sev%	Severity of the sensor that fires the action; integer number from 0-8 representing sensor severities.
%sevstr%	Upper case string representation of the %sev% integer code. Example: WARNING, CRITICAL, SUCCESS.
%srvcaty%	Integer value of service category. Refer to Table 4-28, Service Category.
%srvtype%	String value of service type.
%objtype%	String value of object type.
%time%	Time stamp of the event: **HH:MM:SS AM/PM**
%user%	Name of the user as specified in the *From User* field of the sensor *Alert* properties.

%value%	Current sensor value as specified in the sensor wizard.
{env_var}	Where **env_var** is a java environment property. The complete list of properties varies from system to system. Please refer to http://www.javasoft.com/ and search "java properties".
%ovosev%	Maps M6 severity to HPOVO severity.
%tecsev%	Maps M6 severity to Tivoli TEC severity.

24. *Show value* field defines the value for a sensor. This can be generated using the value of a fact directly or by evaluating an expression.

NOTE You may also use environment variables in the show value field.

25. Set the properties for your *sensor alert* using the options outlined in the table and figure below:

Table 4-35. Alert Properties

Parameter	Description
Severities:	Select the severity levels that will trigger notifications: Unknown: 0.0 (0%), default alert setting Emergency: 0.1 (10%) Critical: 0.25 (25%) Failure: 0.45 (45%) Error: 0.65 (65%) Warning: 0.75 (75%) Success: 1.0 (100%) Debug: 1.0 (100%) Information: 1.0 (100 %)
Suppress false alarms	Enable/Disable ☑/☐ false alarm. Intelligent False Alarm suppression logic. Disables duplicate actions/notifications when sensors are acknowledged/armed.
Use Defaults	Click **Use Defaults** to use the default properties that were defined in domain.properties on each domain server installation. Click **Yes** in confirmation box.
From User	Enter your mail server user id. Usually is the same as your email address.
Mail Server (SMTP)	Enter the name of the local mail server (SMTP server).
E-mail to	Enter the email address of the party to be notified (e.g., %f0:user@nastel.com)
E-mail subject	Enter user-defined e-mail subject line.
E-mail MIME type	Select text/plain or text/html e-mail.
Send Test Mail	Click button to verify e-mail reached the intended recipient.
User Action:	Use to specify an external command, shell script or executable: (e.g., *runaction.bat*). You may also pass business view context information that could be used within a script or action. Refer to section 4.9.4, Automated Actions for a complete list of supported action variables.
Event ID	User assigned event identification number. You can customize the number as needed, but only using numbers, no spaces (example: default = 3010. Custom = 4089). This ID is used to record events into M6 event log.
Event mask	Event format mask used to record events triggered by the sensor. Users may customize the event format. Refer to section 4.9.4, Automated Actions for a complete list of supported variables that can be used within the event mask.
Scheme	Open the Scheme menu and select the specific occurrence that will trigger notifications.

	None: Notification will not be generated.
	Once: Notification generated once per severity status change all severities selected in Alert On section.
	Change: Notification generated with each change in status.
	Repeat: Sensor repeats actions/alerts until the senor goes into a state, which is not checked in the Alert On section. Sensor re-arms until the sensor goes back to an unchecked state. Notification frequency is based on Re-arm Delay intervals.
Trigger Delay	User defined delay times expressed in milliseconds (1/1000th of a second). Default is 60000, or 60 seconds. The sensor will delay notification for the prescribed time.
Repeat Limit	Maximum action/notification repetitions.
Re-Arm Delay	User defined re-arming delay of sensor notifications. Sensor will resample and resend alert notification continuously (per delay time) until condition is corrected. Default is 60000, or 60 seconds.

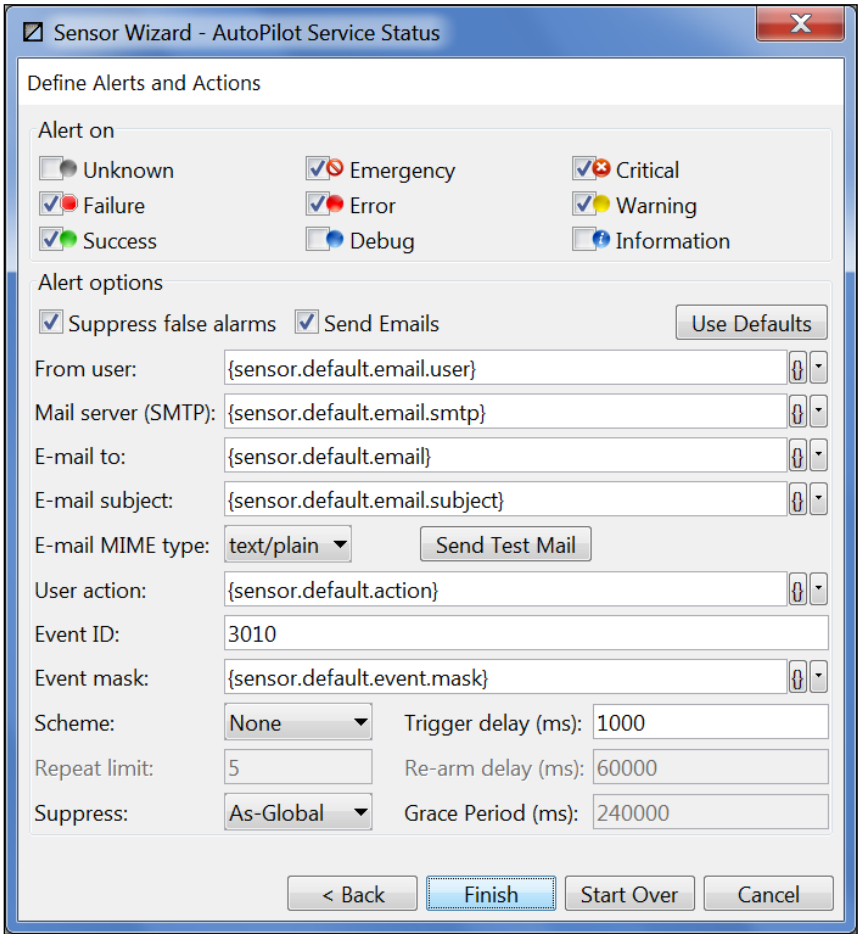


Figure 4-86. Alert Properties

26. Click **Finish** to complete the sensor.

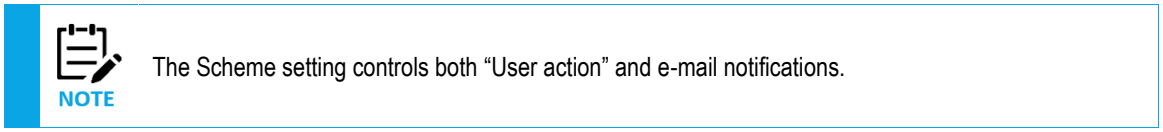
4.9.4 Automated Actions

The sensors in business views monitor one or more facts; based on the rules applied and conditions the sensor will react in a manner you prescribed when you defined the sensor.

By using user actions to invoke automated responses, you can overcome system conditions and failures that would interfere with your operations or process.

Sensors evaluate inputs by applying user-defined (logical) rules. It assigns Severities to (source) Fact/Sensor inputs based on user-defined sensor parameters.

Sensor evaluates the rules and assigns severity when the values of the monitored facts change. It will initiate a number of user-defined actions as well. It may trigger Alerts (example: beeps, e-mail, pager messaging etc.). It can also start automated user scripts within the sensor (example: start procedures or redirecting data to a new collection point).



Sensor *User Action* fields can be used to specify an external command, shell script or executable (example: `myscript.bat %sev% %desc% %facts% %event%`). The User Action is usually taken in response to a problem or event. If you want to receive an e-mail notifying you of a specific event or condition, you can have the event, along with other status or properties of the sensor included by adding action parameters to the User Action to go with the script.

Table 4-36. User Action Parameters

Variable	Description
%account%	Name of current user context. Used within event mask and sensor action fields.
%date%	Date stamp of the event: **Month, Day, Year**
%desc%	Description of the firing sensor defined in the wizard or properties of the sensor.
%event%	Event message that qualifies the sensor. The message includes the same information that would be logged or viewable in an event viewer.
%facts%	Facts and their values in the form: `fact1=value1,fact2=value2,factN=value.` The %facts% may need to be enclosed in "" quotes, since it may contain blanks or special characters.
%from%	Component that generated the event (sensor name in this case).
%f#(token)%	Where %f# refers to the fact being monitored by a sensor and # is the fact index number, token is a zero-based token of the fact delimited by "\" delimiter. Example: Given %f0=Expert\TK1\TK2\TK3\TK3\Var Token1: TK1 as %f0(1)%, expert name: %f0(0)%, Token2: %f0(2)% Example: action %sevstr% token2="%f0(2)%"
%health%	Sensor health index from 0.0 to 1.0
%id%	Message id of the sensor, as defined in the *Alert ID* field of the sensor *Alert* properties.
%parent%	Name of the immediate parent sensor.
%party%	List of email addresses parties as specified in *Email To* in the *Alert* Options.
%related%	Event message that triggered the sensor. May need to be enclosed in ""quotes, since it may contain blanks or special characters.
%root%	Name of the root sensor (top-most sensor) of the business view.
%sev%	Severity of the sensor that fires the action; integer number from 0-8 representing sensor severity.
%sevstr%	Upper case string representation of the %sev% integer code. Example: WARNING, CRITICAL, SUCCESS.
%srvcaty%	Integer value of service category. Refer to Table 4-28, Service Category.

Table 4-36. User Action Parameters	
Variable	**Description**
%srvtype%	String value of service type
%objtype%	String value of object type
%time%	Time stamp of the event: **HH:MM:SS AM/PM**
%user%	Name of the user as specified in *From User* in the *Alert* Options
%value%	Current sensor value as specified in the sensor wizard.
{env_var}	Where **env_var** is a java environment property. The complete list of properties varies from system to system. Please refer to http://www.javasoft.com/ and search "java properties".
%ovosev%	Maps M6 severity to HPOVO severity.
%tecsev%	Maps M6 severity to Tivoli TEC severity.

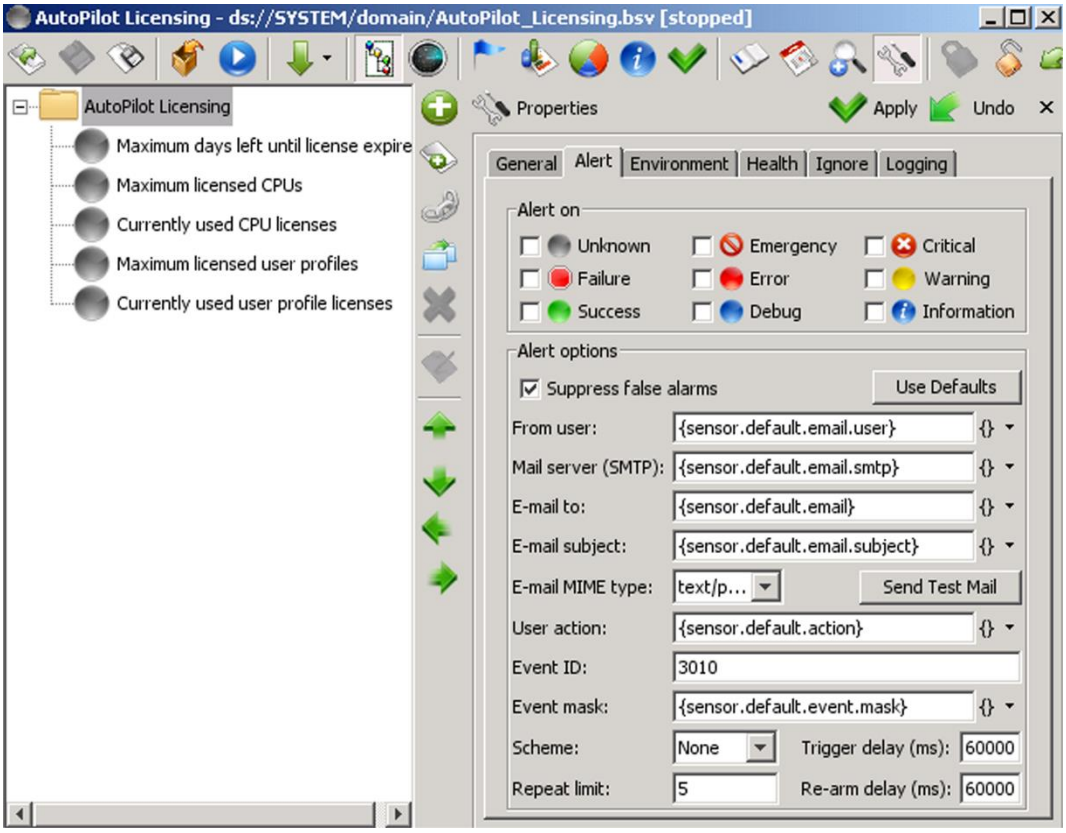


Figure 4-87. Sensor Alert Options

4.9.5 Managing Sensors

The user can add, view, change, copy, and debug sensors by using the sensor options menu.

NOTE The availability of some options depends on the status of the policy.

Right-click a selected sensor to display the sensor options menu.

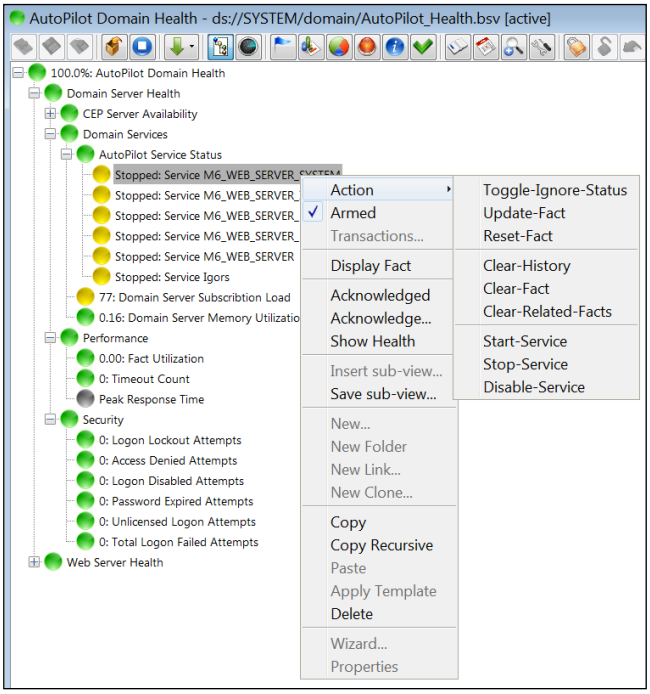


Figure 4-88. Sensor Options

Sensor Menu options are described in the table below.

Table 4-37. Sensor Menu Options

Option	Description
Action	Enables submenu for toggling status, updating, resetting, and clearing facts, and starting, stopping, and disabling services. Submenu will not be displayed on sensors that monitor multiple facts that use filters.
Armed	Disarms/Arms the policy. Action notifications are enabled when Policy is armed. Can be armed without stopping business view.
Transactions	Monitors transactions. Opens the TransactionWorks Explorer.
Display Fact	Opens the Deployment Tool and positions to the fact in the tree. If there is more than one fact, they are listed in a dropdown menu and the user can select one.
Acknowledged	Un-acknowledges/Acknowledges the selected sensor. Only enabled when sensors are started. When acknowledged, ACK is displayed immediately before the sensor name.
Acknowledge…	Opens an *Input* screen where user can acknowledge a sensor with a note so that other users can view it.
Show Health	Displays, in percentage format, the health of the selected sensor.
Insert sub-view	Opens the *Load Sub-View From* screen for selection of Business View to be added. Only enabled when sensors are not running.
Save sub-view	Saves added sub-view to M6 or your local machine.
New	Starts sensor wizard for defining a new sensor. Only enabled when sensors are not running.
New Folder	Adds a new folder to your selected sensor. Click **Properties** to define the new folder. Only enabled when sensors are not running.

Table 4-37. Sensor Menu Options	
Option	**Description**
New Link	Opens the *Link Sensor to Policy* screen to link a selected sensor to a specific existing sensor under a policy. The new link is added to your display. Only enabled when sensors are not running. Sensor links can be expanded within the Master View without having to open a new window.
New Clone	Opens the *Select Model Policy* screen where user can create a model sensor (template) to a previously saved business view. Ability to associate sensor with a model sensor (template). Once sensor is enabled it would load configuration options from the template. (Refer to section 4.9.5.1 for details on this option.)
Copy	Copies parent sensor only. Copying sensors from a running business view to another running business view preserves the context/state of the running sensor.
Copy Recursive	Copies parent sensor and all child sensors.
Paste	Pastes the copied sensor to your selection. Only enabled after *Copy* or *Copy Recursive* is selected.
Apply Template	Contains default service definitions used during new service deployments written in .xml format. Each .xml template should contain only one service definition. Templates are only valid for domain and CEP server installations.
Delete	Permanently removes selection. Only enabled when sensors have been started.
Debug	Immediate debugging of selected sensor. Debug messages are located in the Business View log as configured in the policy logging section. **This option is displayed by holding down the Shift key while right-clicking the started sensor.**
Wizard	Displays the sensor wizard for redefining a sensor. Only enabled when sensors are not running. **This option is not displayed on the parent sensor.**
Advanced	Ability to redefine the selected sensor without using the Sensor Wizard. **This option is only displayed by holding down the Shift key while right-clicking the started sensor.**
Properties	Opens the *Properties* screen to display and/or change your property settings. Only enabled when sensors are not running.

4.9.5.1 Clone Sensor

A clone sensor can be created that will take on the definition of a selected model policy. A model policy is a regular business view containing reusable rules. A clone sensor will morph into the business view it is cloning during runtime, as if that business view had been inserted in its place.

Model policies allow common business logic to be created and maintained in a single place. Changes made to a model policy will be reflected by all clone sensors that reference it in all business views upon restart. If you need to use the same rules in many different business views, you can create a model policy once and simply create a clone sensor where those rules should be used. For example: if you want to monitor five different servers, you can create one server and then create four clone servers.

Model policies are located by default in `ds://model_policies` and can be changed by setting *property server.model.policy.folder=ds://<model policies folder>* in global.properties on domain server installation. Unlike link sensors, model sensors copy a model's definition at runtime when the policy is enabled. Changing a model sensor will affect all sensors that model after it only when the policy is restarted.

Table 4-38. Differences Between Clone Sensors and Link Sensors	
Clone Sensor	**Link Sensor**
Copies a model's definition at runtime.	Links to an already deployed business view.
Model instance of business view.	Runtime instance of policy.

To **create a clone sensor**, do the following:

1. Save the business view to be cloned into the model policies folder.
2. In a different business view, right-click a folder sensor and select **New Clone** or click the *New Clone* link  to display the *Select Model Policy* screen.
3. Select the model business view to be cloned and click **OK**. The *Properties* dialog box is displayed. The only properties that can be changed are on the **Ignore** tab. This allows you to set a schedule different from the model (parent). If a field is not filled in, values from model are used.

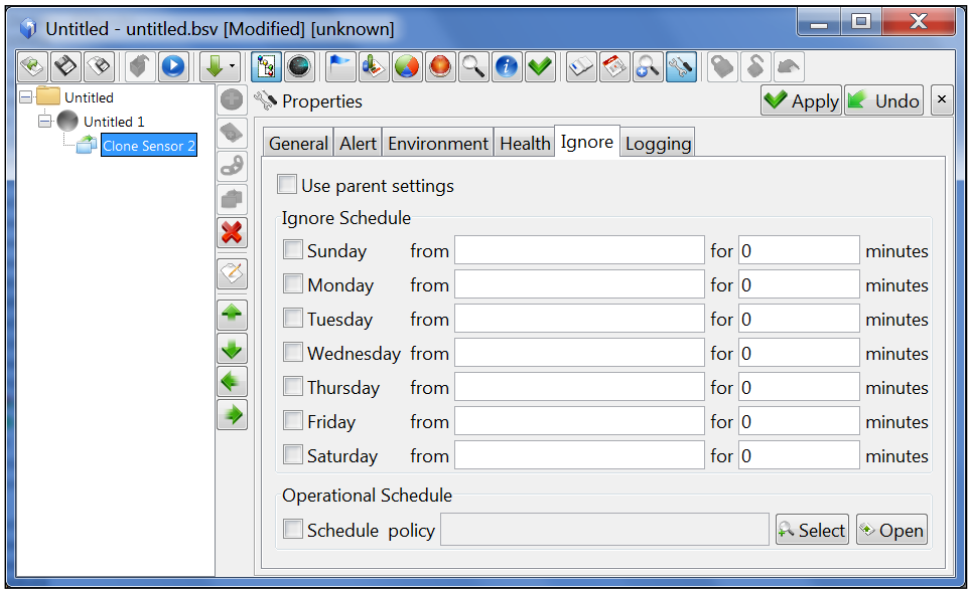


Figure 4-88A. Clone Sensor Properties

To **change an existing clone sensor** to clone a different model policy:

1. Right-click the clone sensor and select **Model From** to display the *Select Model Policy* screen.
2. Select the model business view to be cloned and click **OK**.

To **open the model business view** of a clone sensor, right-click the clone sensor and select **Open Model**.

4.9.6 Business Process

Business Process  is a tool supplied with M6 User Console and can be launched from M6 User Console's *Tools* menu. Business Process is used to create and view business views as a process flow rather than in a hierarchical form. These sensors are set up in the same way as business view sensors with the addition of a graphing capability on the properties screen.

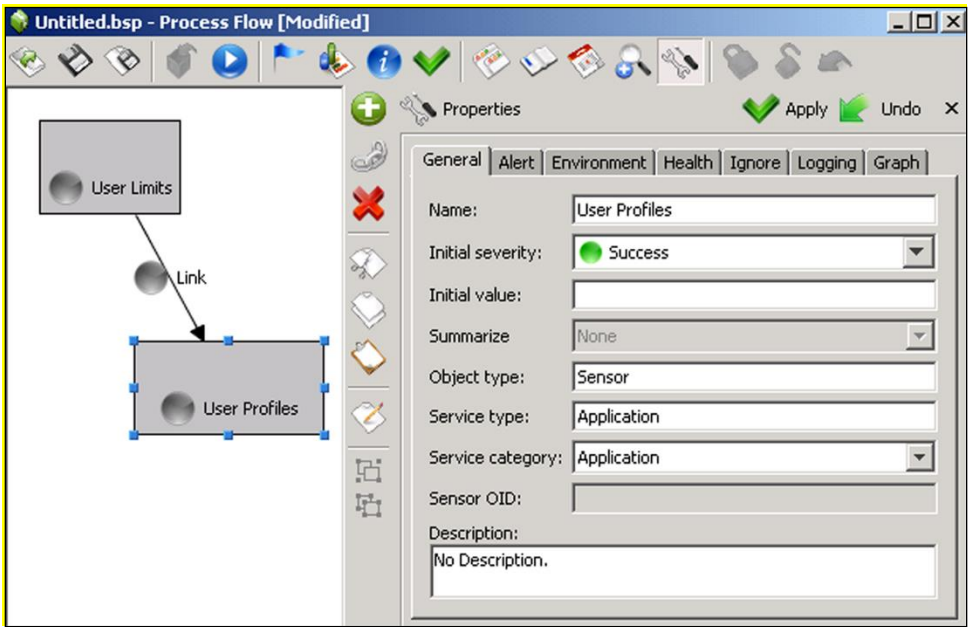


Figure 4-89. Process Flow Graphing Capability

To create a process flow, use the sensor wizard as you would for a business view. Each new sensor created will have its own rectangle with your designated sensor name inside. Connections are added by clicking one sensor on the blue dot and dragging your cursor to the next sensor's blue dot.
Configure or customize your process flow as follows:

Table 4-39. Process Flow Configuration

Icon	Description
Open	Open a previously created business process.
Save	Saves current business process as a .bsp file.
Save As	Saves current business process with file type and name that you select.
New Sensor	Creates a new sensor using the Sensor Wizard.
New Link Sensor	Opens *Link Sensor to Business View* screen to link selected sensor to an existing business view. Sensor links can be expanded within Master View without having to open a new window.
Delete	Deletes your sensor.
Cut	Places selected sensor on clipboard. Sensor is removed.
Copy	Copies selected sensor to your clipboard. Sensor is not removed.
Paste	Pastes the cut or copied sensor to your selection.
Design	Displays the first screen of the sensor wizard.
Group	Groups selected sensors together. To select shapes, click each shape to be grouped while holding the Ctrl key. Click the *Group* icon.
Ungroup	Ungroups sensors that have been previously grouped.
Apply	Applies your last action.
Undo	Undoes your last action.

To save your process flow, click the **Save** button in the toolbar. Your process flow will be saved as a `.bsp` file.

Process flows are also displayed on the M6 Web Server for viewing and monitoring purposes only. They can only be created or edited from the Nastel M6 User Console.

4.9.7 Displaying Sensors Graphically

Business View and Business Process pre-defined sensor values can be displayed graphically for the selected root sensor, including all child sensors. The automatic sensor graph capabilities use charts and bar graphs to display sensor values. The graphs are configured based on which profiles you select. You may use the default properties or change them as you wish. The default properties as well as console profile properties are specified in each profile under `[AUTOPILOT_HOME]\naming\profiles\<prof_name>`. All profiles are stored in Domain Server.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define AutoPilot console menu and tool bar.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; configure graphing options
property console.graph.refresh = 1000
property console.graph.usage.history.size = 5000

; graph dimensions
property console.graph.cols=3
property console.graph.bars=10
property console.graph.bar.ratio=7
property console.graph.min.bar.width=20
property console.graph.rows=10
property console.graph.max.charts=30
property console.graph.spacer=4

; font and stroke
property console.graph.font.name=Arial
property console.graph.font.size=12
property console.graph.stroke.size=3.0

; color coding
property console.graph.line.color.code=true
property console.graph.bar.color.code=true
property console.graph.label.color.code=true
property console.graph.fill=false

; gradient
property console.graph.fill.gradient=true
property console.graph.bar.gradient=true
property console.graph.line.gradient=true

; rendering hints (anti-aliasing)
property console.graph.render.hints=false

property console.view.default.map=false
```

Selecting a previously configured profile

1. Select **User > User Manager** from the M6 toolbar.
2. Right-click profile to be used and select **Properties**.

3. Click **Misc** tab.
4. Type name of profile (use same profile name as specified under `[AUTOPILOT_HOME]\naming\profiles\<prof_name>`) in the *Console Profile* field and click **OK**. (Refer to section 4.2 for further information on managing users.)

Procedure to display sensors

To display sensors graphically, do the following:

1. Open the Business View or Business Process that you want to display sensors for.
2. Select the parent sensor that you want to view.
3. Click the graphs icon. The graphs for parent and children will be displayed on the right of your screen. All display elements are based on the profile that you have selected.

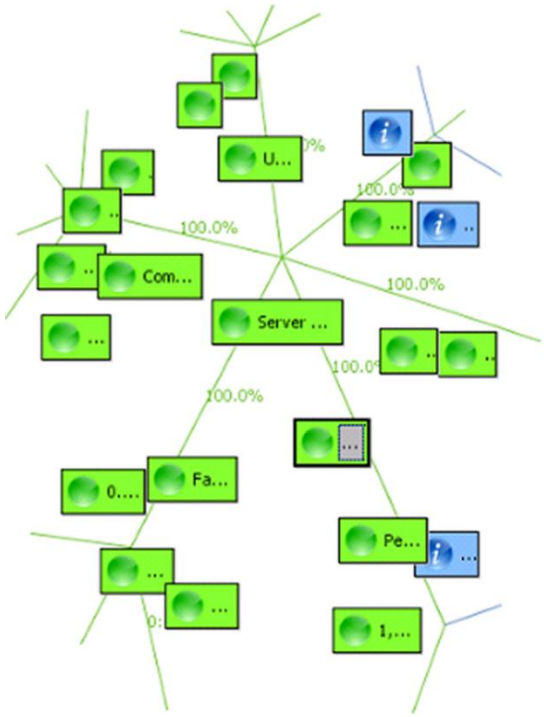


Figure 4-90. Sensor Value Graphical Representation-left side

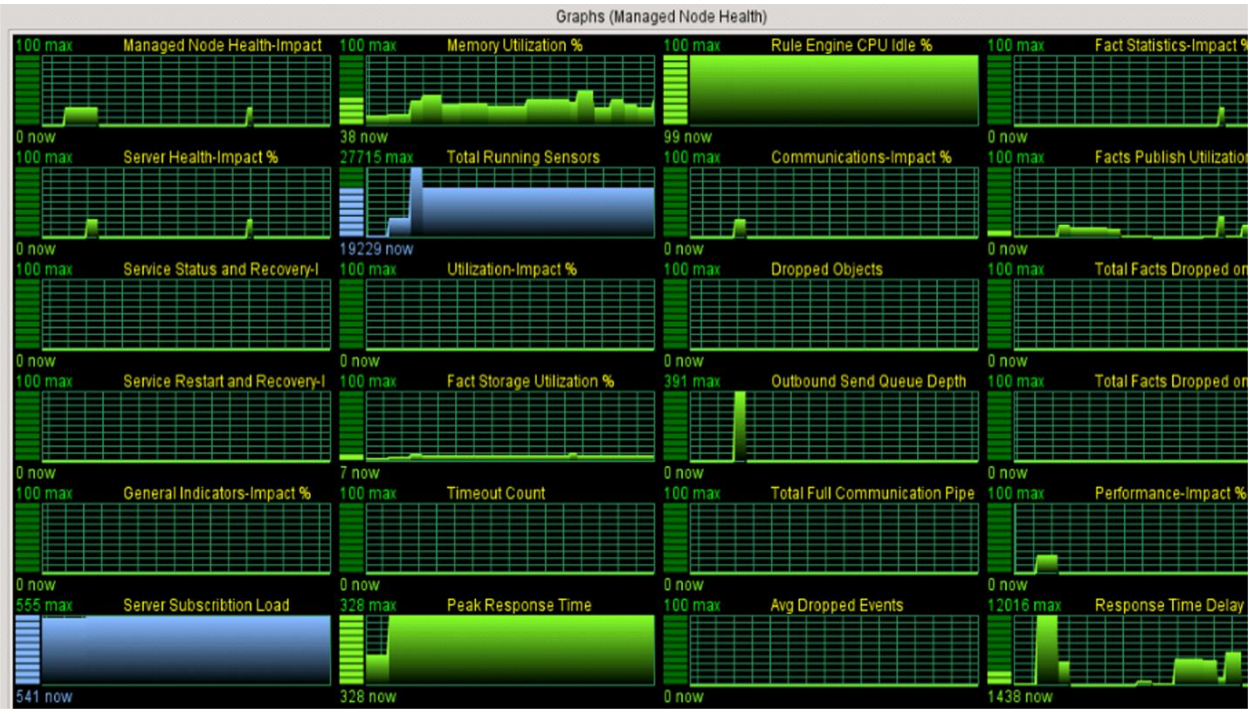


Figure 4-91. Sensor Value Graphical Representation-right side

4.9.8 Setting User Actions

The Action Manager allows users to set predefined user actions (scripts and executable programs) that can be executed in business views. There are three ways to set up an action.

- Through the Action Manager tool
- Through *Properties* screen under the *Alert* tab
- Through S*ensor Wizard* screens under the *Define State* screen.

These user actions must be in the path of the running CEP server or domain server where the business view is deployed, or the path must be specified within the *User Action* field of the Alert Details. Functions may include parameters that are passed on to the user action. Environment variables can be specified using {`env_var`} format, for example, `MYAPPL.BAT {SERVER} param1 param2…paramN`.

Within Action Manager or the action field of sensor wizard, native OS commands/scripts/executables can be executed on any CEP server (servers running AP CEP server) using the following:

@$[*node*]$[*user_action*] For example: @$server1$start_command.sh.

Defining an action using the Action Manager gives the user two distinct advantages. It allows the user to change an action without changing the business view. It also gives the user the opportunity to enable or disable a specific user action for all sensors at one time instead of enabling/disabling a specific user action for each sensor individually – as you would have to do using the S*ensor Wizard* or *Properties* screen. Refer to table and figure below for setting user actions.

IMPORTANT! Parameters must be comma separated. Action invocation will fail if parameters have imbedded blanks.

Table 4-40. User Action Key Words/Prefixes

Key Word/Prefix	Description
Call	Java-based public static method within any classpath in the CEP server. For example: `call:com.nastel.nfc.util.GeneralUtils.getSeparator()`
Method *	Any method already used within M6 with the format: *`ServiceName.methodName(arg_list)`*, where *`ServiceName`* is the name of registered M6 service such as an expert or manager. *`methodName`* is the name of the public method implemented by the service. *`arg_list: type=value1,…type=valueN`* For example: `method:JMX_Monitor.Invoke(String=Domain,String=MBeanName,String=mbean_method,String[]=)`
@	Only used for executables and scripts. Runs action on system where the facts execute the action. The action is run from where the action is, not on the business view.

* Supported method types can be String, Integer, Long, Float, Double, Boolean

Supported array types can be String[], Integer[], Long[], Float[], Double[], or Boolean[]

Array values are specified in the following format ['val1"val2'…'valN'] with no spaces between values. Example: String[]=['This"is"a"test'] is a list of four strings. Empty values can be specified as null.

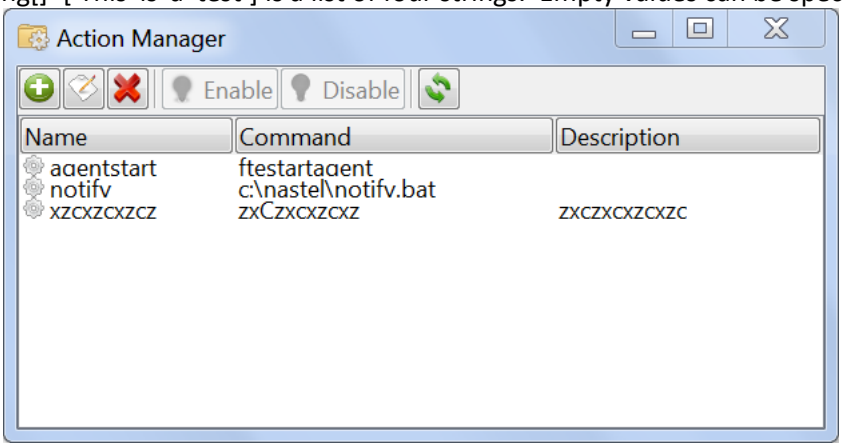


Figure 4-92. Action Manager Screen

To create a predefined user action, do the following:

1. Click the **Action Manager** icon at top of M6 User Console screen. The *Action Manager* screen will be displayed.
2. Click the **New Action** icon at top of this screen to create a new action. A window opens.

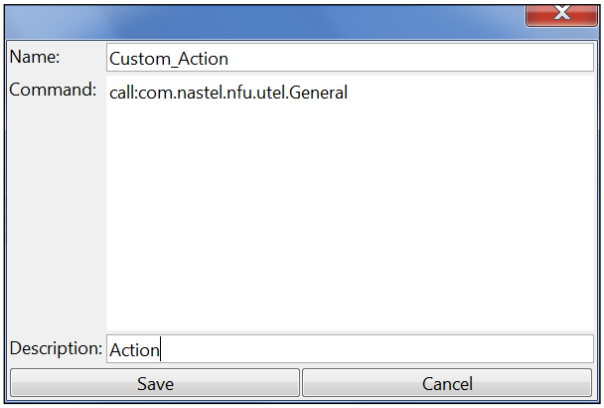


Figure 4-92A. Create Action Screen

3. Enter the name (logical) of the action.
4. Enter the command (physical) for the action.
5. Enter description of the action. (This is not required.)
6. Click **Save**. The action will be listed on the Action Manager screen *(Figure 4-92)*.

Actions can be enabled or disabled as necessary by clicking the appropriate icon. When an action becomes disabled, it turns to grey. To delete an action, click the delete icon. Actions can also be enabled/disabled or deleted by right-clicking and selecting the action. To refresh your screen, click the refresh icon.

> **NOTE**
>
> When actions are executed, they are retrieved from the Domain Server. With high volume actions, this can create high traffic between the executing CEP and the Domain Server. To cache the action, set the following environmental variable:
>
> For Production – **property server.action.cache.active=true**
>
> For Development – **property server.action.cache.active=false** (or undefined). This is the default.

4.9.9 State Change Delay

The *State change delay* field on the *Health* tab (Figure 4-93) designates the amount of time a new condition must be in affect before the state status displayed in the business views changes. This prevents the triggering of an alert when there is a spike in the sensor data. A value of 0 indicates no delay when the state changes.

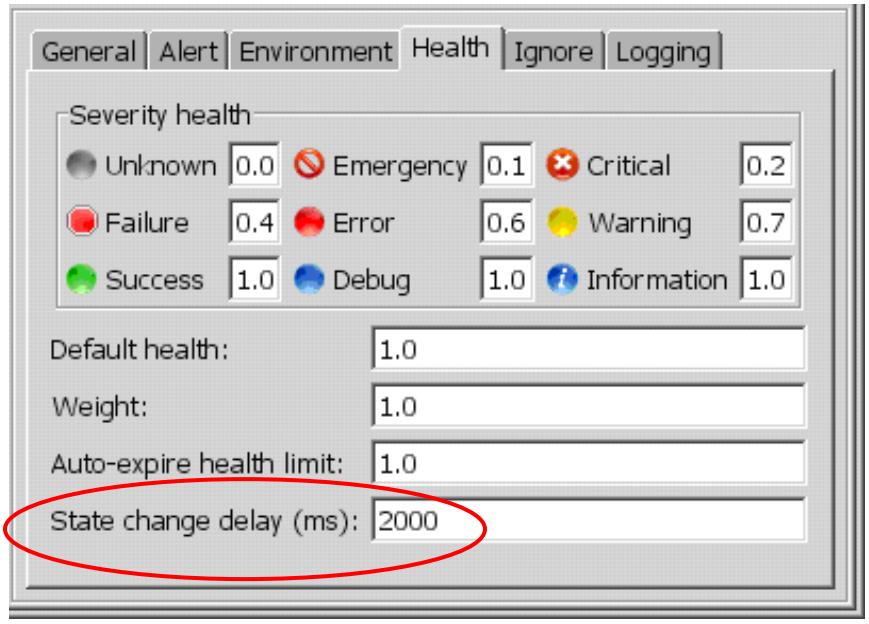


Figure 4-93. Health Tab

The following is an example of the *State change delay* functionality.

In this example (Figure 4-94), the data is updated every 10 seconds and there is a 20-second delay. This means the condition must be true for 20 seconds for the state status to change in the business view. The state status is not necessarily updated every 20 seconds, but rather updated when the condition changes and remains in the new condition or below for 20 seconds. For simplicity, this example has only three conditions/statuses.

NOTE

On the upward severity, all timers between the current state and the new state start at the same time. For example, when the condition goes from green to red, both the yellow and the red timers start, implying the condition is "at least" yellow. If the condition dips down, the timers above the current condition stop, but the rest keep running. For example, if the condition dips from red to yellow, the red timer stops and the yellow timer continues to run. If the condition dips from red to green, the red and yellow timers stop, and the green timer starts. The condition must be true for the time specified in the *State change delay* field on the *Health* tab for the state to change.

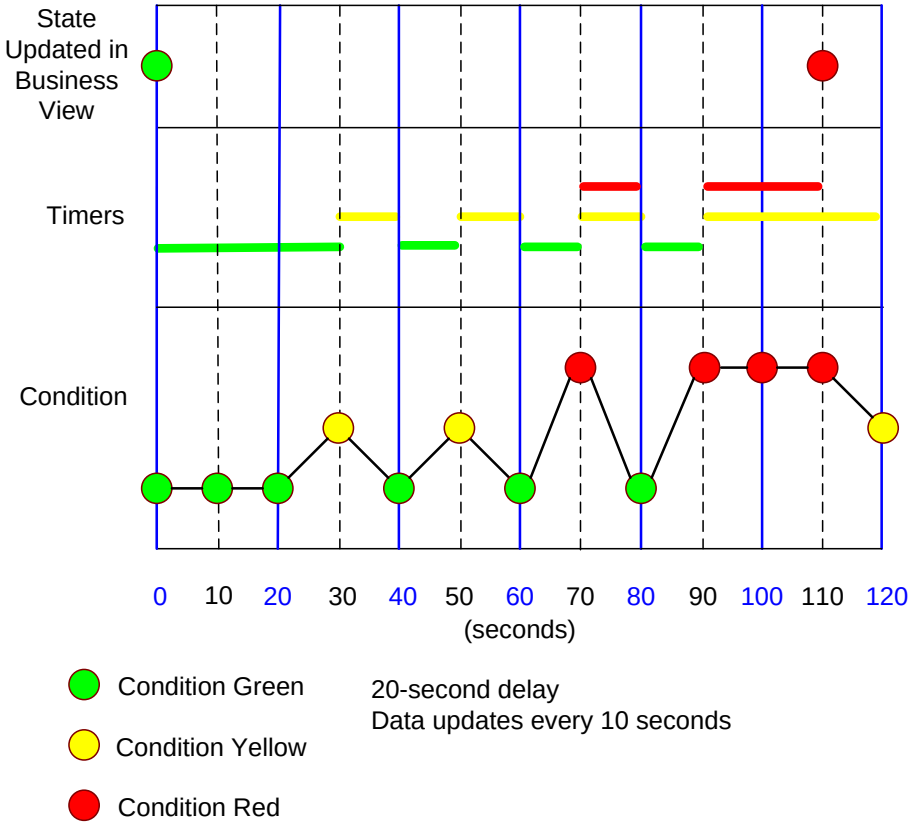


Figure 4-94. State Change Delay

The state starts as green. At 30 seconds it changes to yellow which starts the yellow timer. At 40 seconds the condition dips to green and the yellow timer stops. Since the system was in yellow for only 10 seconds, the state in the business does not change.

The condition continues to fluctuate, but each change is only true for 10 seconds so the state in not updated and remains green.

At 90 seconds the condition changes to red and stays red for 20 seconds causing the state in the business view to change to red at 110 seconds.

NOTE

The *State Change Delay* functionality works in conjunction with the *Trigger delay* setting on the *Alert* tab. The settings on the *Alert* tab do not take effect until the delay has been reached.

4.9.10 Specifying Maintenance Schedule

There are two methods of specifying maintenance schedules in M6. Only one method (schedule) can be selected at a time.

- **Ignore Schedule** – selected sensors within a business view ignore faults and alert conditions during a specified window of time – at a specified time, any faults/alerts are ignored.

- **Operational Schedule** – a user-defined business view is set up to associate an operational schedule with a sensor(s) to control ignore states.

Ignore Scheduling

By specifying day and times periods for sensor maintenance window, alerts and user actions defined for normal operations can be blocked. Time periods set will be recurring until reset. If a sensor is set to ignore its monitoring on Sunday from 6 AM for 360 minutes (six hours, 6 AM to noon) the event that occurs during that time frame will be logged, but alerts and user actions will not be activated. The sensor severity status will not be propagated up the business view hierarchy and the sensor will have **IGN** displayed next to it.

Define maintenance window as follows:

1. Right-click sensor to be ignored.
2. In sub-menu, select **Properties**, the *properties* screen will be displayed.
3. Click **Ignore** tab to open *Ignore* properties screen.

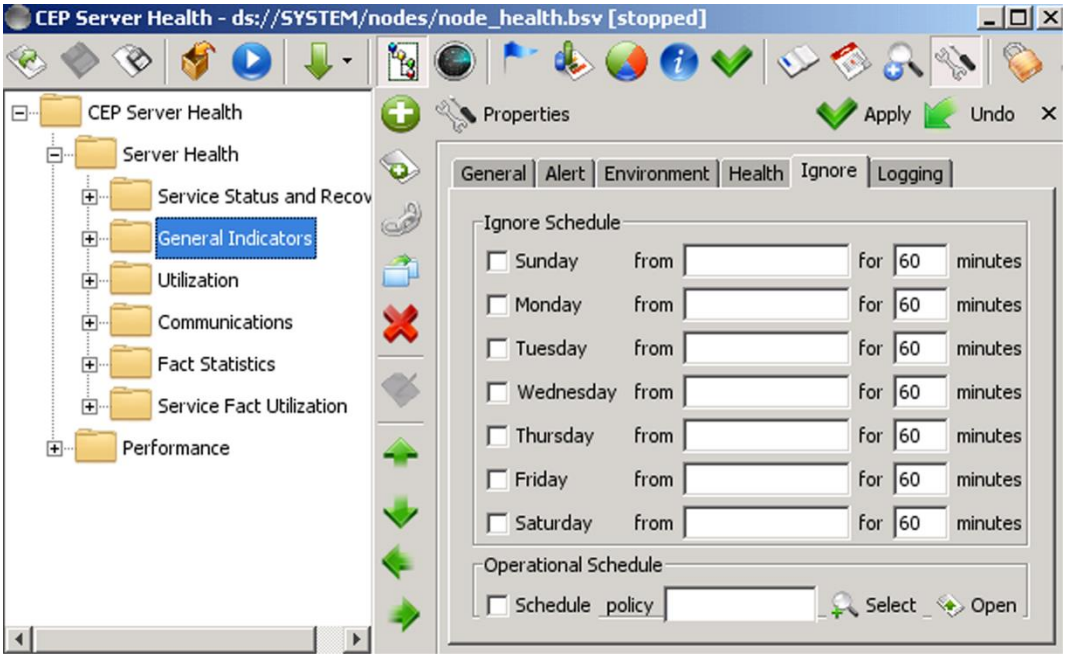


Figure 4-95. Opening Ignore Properties - default screen

4. For days to be ignored:
   a. Enter time for the days you require.
   b. Enter number of minutes you want the sensor to be ignored. (Example: 12 Hours = 720 minutes, 24 hours = 1440 minutes.)
   c. Check (enable) the day. Sensor activity for the day and time specified will be ignored.
5. Ensure all other days are disabled (unchecked) to prevent inadvertent disruption of sensor data. When a day is disabled, the current time (default) for each day deselected is displayed, the day becomes grayed out and the ignore feature is disabled.
6. Click **Apply** when all day and time settings are set. All information for the specified period will be ignored.

The sample illustrated below shows a sensor set to be ignored from 5 PM, Friday until 8 AM, Monday.

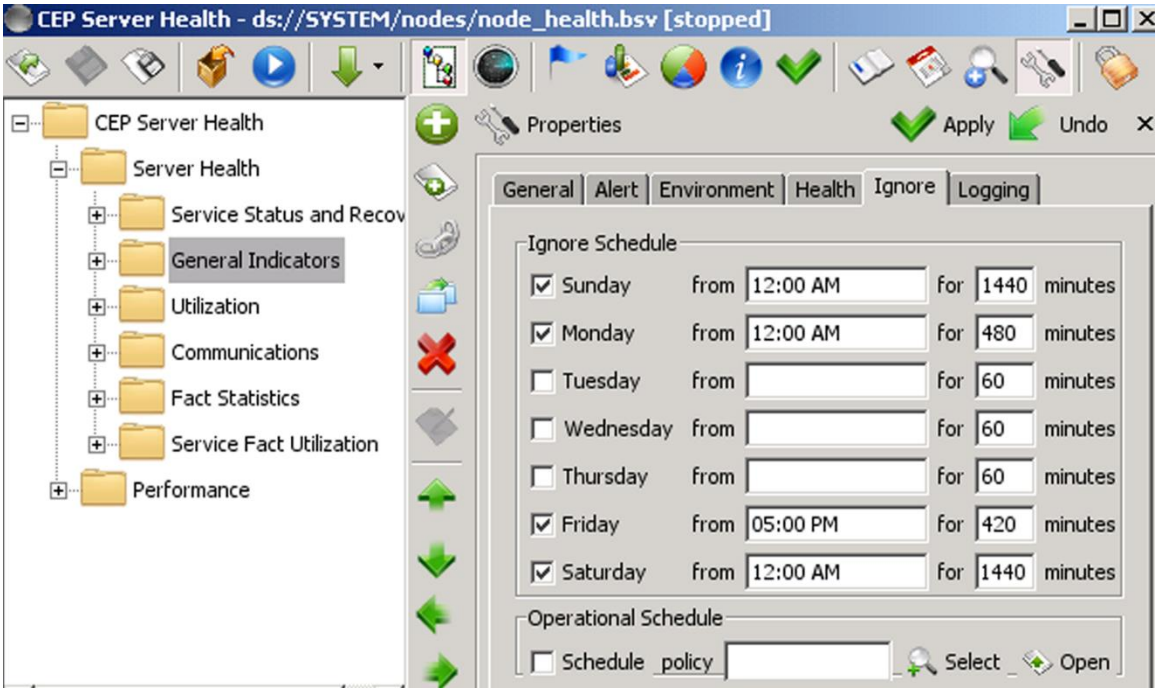


Figure 4-96. Maintenance Window -- Ignore Settings

Operational Scheduling

Users can define a rule-based operational schedule to select sensors to ignore faults and alert conditions as specified by a user-defined business view. This can be set up to change ignore settings as conditions change. If success (green icon) the sensor runs, if not success (icon any color other than green), the sensor is ignored. This provides more flexibility for the user to control his ignore settings.

The following is an example of a policy that can be used for an operational schedule:

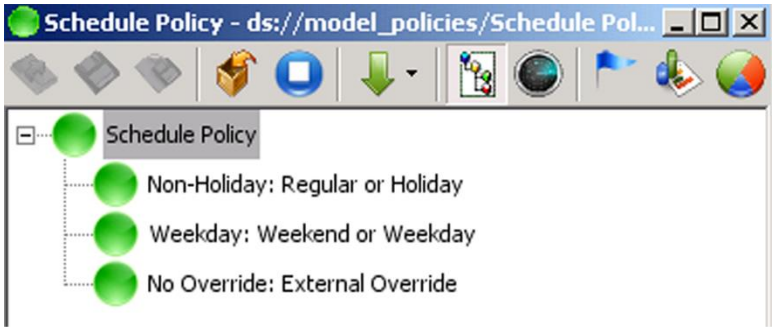


Figure 4-97. Example of a Schedule Policy

In this policy, a success (green) status is returned when not a holiday, weekday, and not overridden. The first two sensors are based on the date and day of the week. The Override is a value that can be sent with apfact to turn off the schedule at any time – such as a previously unplanned event. In addition to using apfact, various ways could be used to override, including the new SQL query capability. Any policy can be used as an operational policy, and it does not need to be limited to date attributes. This policy was run at the Domain Manager for a central point of control but can be run at any CEP Server.

Set up an Operational Schedule as follows:

1. Create a customized policy to fit your operational schedule. (This policy will be selected in Step 5.)
2. Right-click sensor to be ignored.
3. In sub-menu, select **Properties**, the *properties* screen will be displayed.
4. Click **Ignore** tab to open *Ignore* properties screen.
5. Click **Select** at bottom of screen. The *Select Policy as Operation Schedule* screen is displayed. Select policy and then click **OK**. (Click **Open** if you want to review your policy.)
6. Check (enable) the schedule.
7. Click **Apply**.

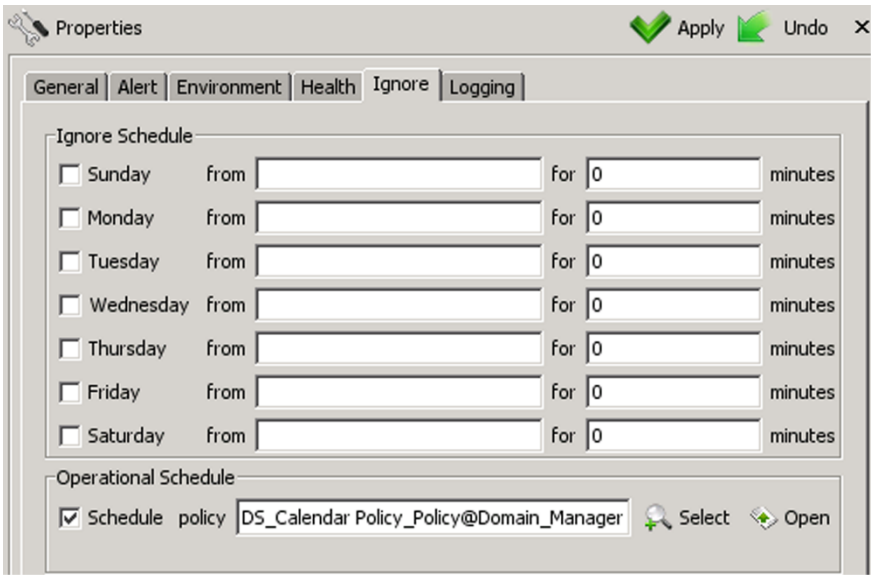


Figure 4-98. Maintenance Window – Operational Schedule

4.9.11 Ignoring Facts

A fact is ignored at the metric level. When a fact is ignored, actions and notifications (alerts) for all sensors that reference the fact are disabled. However, actions placed on the Sensor Evaluation page continue to be invoked even on an ignored fact.

To place a fact in an ignored state:

1. Right-click the selected fact.
2. Click **Action**. A sub-menu will be displayed.
3. Click **Toggle-Ignore-Status**.

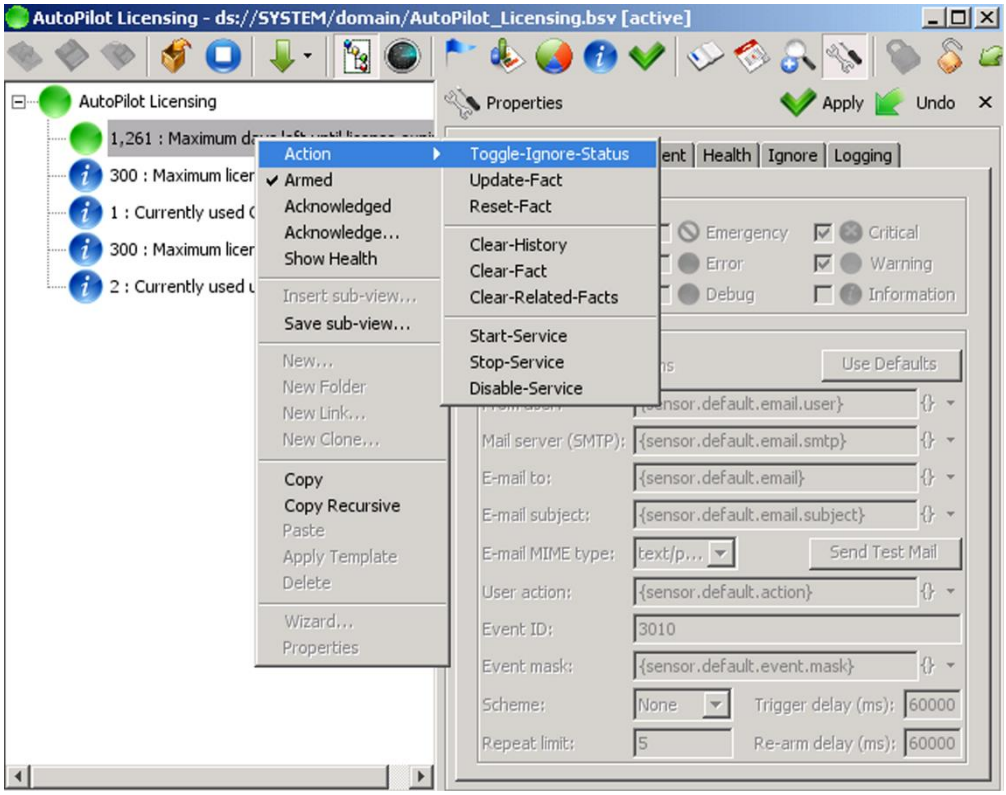


Figure 4-99. Toggle-Ignore-Status Menu

After the action is executed, **IGN** is displayed next to the selected fact. Every sensor that references this fact will stop sending actions and alerts. To reinstate the fact, repeat steps 1-3. **IGN** will no longer be displayed.

4.9.12 Maintaining Sensor History

Business views allow logging of sensor information to database and local flat files. This is useful when historical behavior of variables is required. Once logging is enabled, business view charts can be used to show sensor behavior over a period. The charting option will also be enabled within M6 Web Console. It is recommended that key sensors record historical information. It will also help users understand trends and forecast future behavior.

The sensor logs are configured as follows:

Table 4-41. Log Sensor to Database

Property	Description
Exception based logging only	Click disable/enable button to log sensor information only when the sensor switches to any of the checked states under the *Alert* tab. When button is unchecked, sensor information is logged every time the value of the sensor changes.
Log sensor status to database	Click disable/enable button to log sensor status to selected database.
Use root sensor settings	If the sensor is a model, it will inherit the parent logging settings; that is, the database the parent logs to.
Database	Select a database to receive the log from the menu: • Oracle • SQL Server • Sybase • DB2 • Hypersonic SQL

	• ODBC Data Source • Informix • MySQL • Derby SQL Note: JDBC drivers for the Derby SQL database must be installed separately.
Database server	Enter the name of the database server.
Database name	User-defined file name.
Database table	The description is set when the format is selected.
Database user ID	The user ID is locally managed. See your DBA for local policy on DBA access and user ID. This user needs to be able to create a session, table, sequence, and triggers.
Database password	See your DBA for local policy on DBA access and password.
Test Connection button	Click **Test Connection** to verify your connection to your database. If the test is successful a *Connection Test Succeeded* screen will be displayed.
Create Table button	Click **Create Table** to create a new data table. May not work for some databases that do not support "number" as an SQL data type during table creation. Refer to section 4.9.12.2, Creating a Table Manually, for more detailed information.
Use Defaults button	Click **Use Defaults** to use default properties that were defined in domain.properties on each domain server installation. Click **Yes** in Confirmation box.

When logging is configured, click **Apply,** in upper right corner, to save the logging properties. The business view will begin logging once the business view is deployed. The frequency of recording depends on the frequency of change in sensor status, which is triggered either by changes in facts or changes in child sensors.

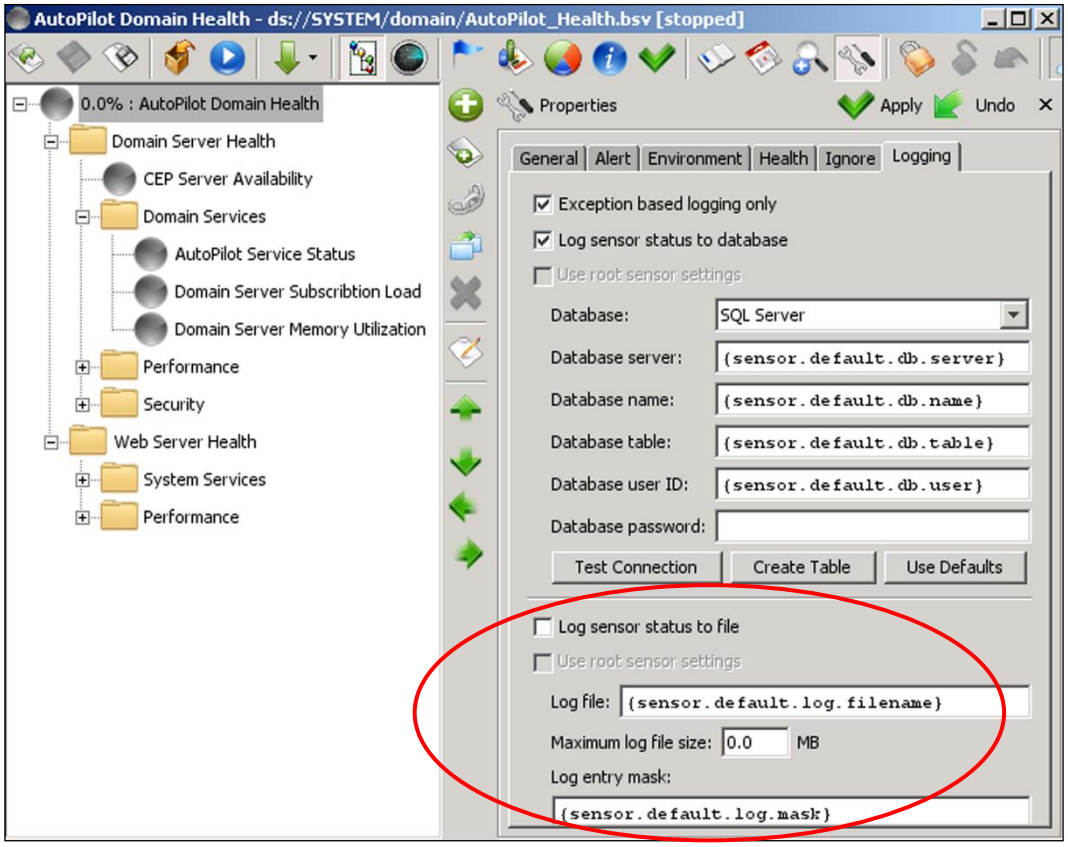


Figure 4-100. Sensor Logging Database Options

Table 4-42. Log Sensor Status to File	
Property	**Description**
Log sensor status to file	Specifies to log sensor status to file. The default file format is EVT, the Nastel event log format.
Use root sensor settings	Specifies to use the root sensor settings as the default setting.
Log file	Enter a logical log file name with a .log extension and complete file address. (Example: *health.log*)
Maximum log file size	Define a maximum file size expressed in megabytes. (Example: 1.0 MB)
Log entry mask	Comma separated list of user-defined tokens or variables such as %sevstr%, %f0, value, etc. in order to log additional entries.

4.9.12.1 Setting Up Pruning

Data can be summarized (daily) and purged from the sensor history tables. The process is called Aggregation. By default sensor history aggregation is disabled (server.sensor.history.aggregate.retain. days=0). When enabled it is kicked off every day at 00:01:00, where data older than five days is summarized and purged from sensor history tables. Aggregation is kicked off on every start and then each subsequent day at 00:01:00.

Scheduling parameters can be changed by defining/modifying the following properties:

// controlling the initial aggregation task (initiated on start only)
server.sensor.history.aggregate.schedule.first=10 (minutes)
server.sensor.history.aggregate.schedule.first.delay=30000 (ms)

// controlling fixed-rate scheduled aggregation task
server.sensor.history.aggregate.schedule.hour=0
server.sensor.history.aggregate.schedule.minute=1
server.sensor.history.aggregate.schedule.second=0
server.sensor.history.aggregate.period=86400 (seconds in 1 day)
server.sensor.history.aggregate.retain.days=2 (retain 2 days' worth of history) setting to 0 disables aggregation altogether.

Additional aggregation metrics have been added under:

<Server>_Facts\Logging\Aggregation

The aggregation automatically determines the tables the need to be summarized by querying "db_services" table. All summary data is recorded into db_sensor_summary table, which must be created first as follows:

MySQL:

```
CREATE TABLE `m6_sensors`.`db_sensor_summary` (
  `SID` int(11) NOT NULL,
  `SensorName` varchar(255) NOT NULL,
  `SensorTable` varchar(255) NOT NULL,
  `SensorDate` timestamp NOT NULL,
  `Records` int(11) NOT NULL,
  `Average` float NULL,
  `Maximum` float NULL,
  `Minimum` float NULL,
  `Deviation` float NULL,
  PRIMARY KEY (`SensorName`,`SensorTable`,`SensorDate`,`SID`)
);
```


SQL Server:

```
CREATE TABLE [dbo].[db_sensor_summary](
   [SID] [int] NOT NULL,
   [SensorName] [nvarchar](255) NOT NULL,
   [SensorTable] [nvarchar](255) NOT NULL,
   [SensorDate] [datetime] NOT NULL,
   [Records] [int] NOT NULL,
   [Average] [float] NULL,
   [Maximum] [float] NULL,
   [Minimum] [float] NULL,
   [Deviation] [float] NULL,
  PRIMARY KEY (SensorName,SensorTable,SensorDate,SID)
);
```

4.9.12.2 Creating a Table Manually

Refer to Appendix F tables to develop your table headings for manual table creation while logging sensors to a database.

The Service Category Table is defined below.

Table 4-43. Service Category

Service Category Number	Service Category Name
0	Hardware
1	Network
2	Server
3	Operating System
4	Middleware
5	Database
6	Application Server
7	Web Server
8	Web Service
9	Client
10	Application
11	IT Service
12	Business Service
13	Transaction
14	Policy
15	Miscellaneous
16	Other

4.9.13 Sensor Performance Counters

Sensor Performance Counters are published under [*managed_node*]_Facts/Facts/Sensor category. All counters are computed since last reset and must be stopped and started to be reset.

Table 4-44. Performance Category

Category	Description
absolute_rate_rules_per_sec	number of absolute rate rules per second
absolute_rate_sensors_per_sec	number of absolute rate sensors per second
last_sensor_exec_time_ms	time in ms. taken by the last sensor execution
max_sensor_exec_time_ms	maximum sensor execution time in ms
min_sensor_exec_time_ms	minimum sensor execution time in ms
total_processed_rules	total number of processed rules since last reset
total_processed_sensors	total number of processed sensors since last reset. (Sensors may execute one or more rules)
rate_rules_per_sec	processing rate of rules per second
rate_sensors_per_sec	processing rate of sensors per second
sensor_idle_percent	percent of time spent outside of sensor processing
sensor_busy_percent	percent of time spent processing sensors and rules
total_sensor_time_ms	total time in ms. spent processing all sensors
total_processed_rules	total number of rules processed
total_processed_sensors	total number of processed sensors
sensor_turn_around_time_ms	measures the time it takes to deliver facts to sensors

Table 4-45. Runtime Category

Category	Description
sensor_arrival_rate_per_sec	total number of sensors arriving per second
sensor_delivery_rate_per_sec	total number of sensors being delivered per second
sensor_total_actions	total number of executed actions (sensor user actions)
sensor_failed_actions	total number of failed actions (sensor user actions)
sensor_total_notifications	total number of successfully sent SMTP messages
sensor_failed_notifications	total number of failed notifications
sensor_rule_backlog	total number of sensors backlogged
sensor_total_expiring_threads	total number of expiring threads
sensor_total_action_threads	total number of outstanding action threads
sensor_total_running	total number of sensors running

4.9.14 Documenting Sensor Information

Sensor information including facts, logic, conditions, alerts, etc. can be saved/exported in the following formats:

- HTML
- PDF
- RTF
- XML

1. Click **Business View Explorer** from the **Tools** menu bar.
2. Expand the folder that contains the Business View to be saved.
3. Right-click the business view and select *Create Sensor Documentation* and then the desired format.

This is especially useful for viewing by other users that are not owners of the business view.

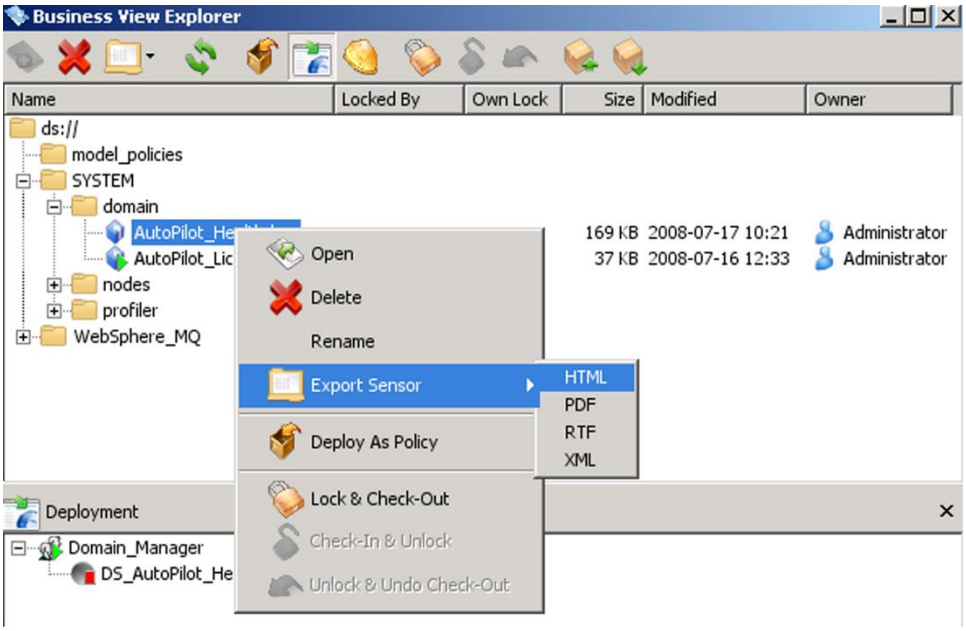


Figure 4-101. Creating Sensor Documentation

4.10 Real-Time Monitoring

This section will discuss the real-time monitoring of business processes, events, and performance issues. M6 has two distinct real-time monitoring capabilities: monitoring of business views and performance monitors. Real-time monitoring enables users to proactively monitor system-wide performance, diagnose system and application bottlenecks, efficiency, congestion, and latency using real time and historical statistics.

Monitoring your environment will affect system performance; the degree of performance degradation depends on user defined sampling rates, size of the environment and number of monitored resources.

4.10.1 Monitoring Business Views

Business views are a collection of rules that define a desired state of an *e*Business environment. Your deployed business views give you real-time views of the information in the form you defined.

Monitoring Business view lets you see the status of applications and identify the impact of failures on the overall environment, enabling you to take prompt actions in response to the events as they happen.

There are many options available when viewing business views. Based on your needs when monitoring business views you can select from several options that offer a good deal of information. You can monitor any single view, or any combination of views of any number of business views. By opening multiple instances of the same business view, you can view multiple detailed areas simultaneously. Open any business view by right clicking it and then selecting *Open* in the sub-menu.

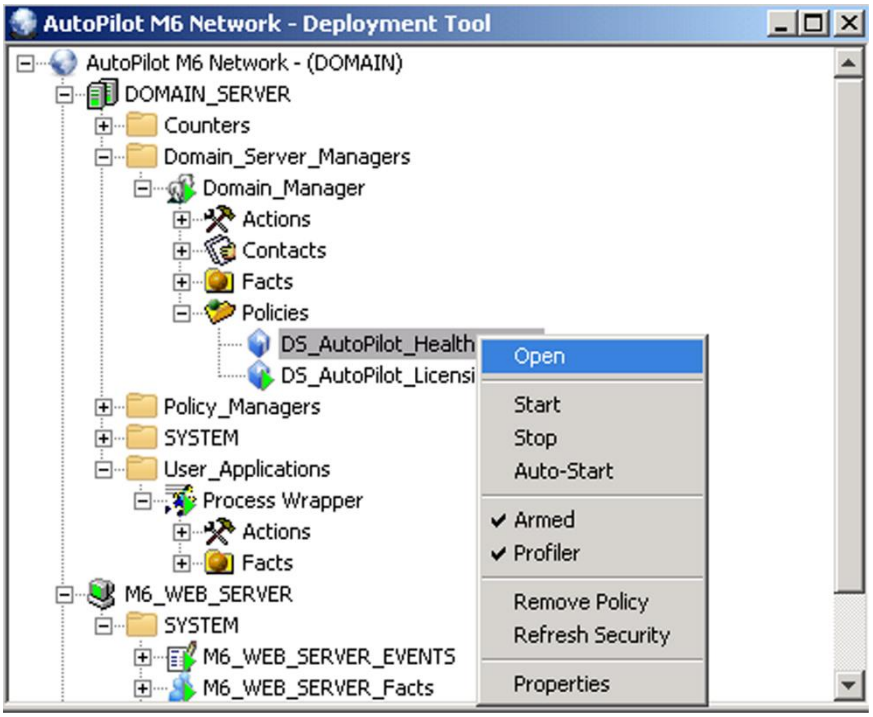


Figure 4-102. Open Any Business View

Typically, the business view will display the configuration from when it was last viewed. In the sample below the *Severity*, *Health* and *State* are all displayed. Business views offer a variety of monitoring options. By selecting options from the tool bar you can open and close, any required monitoring screens.

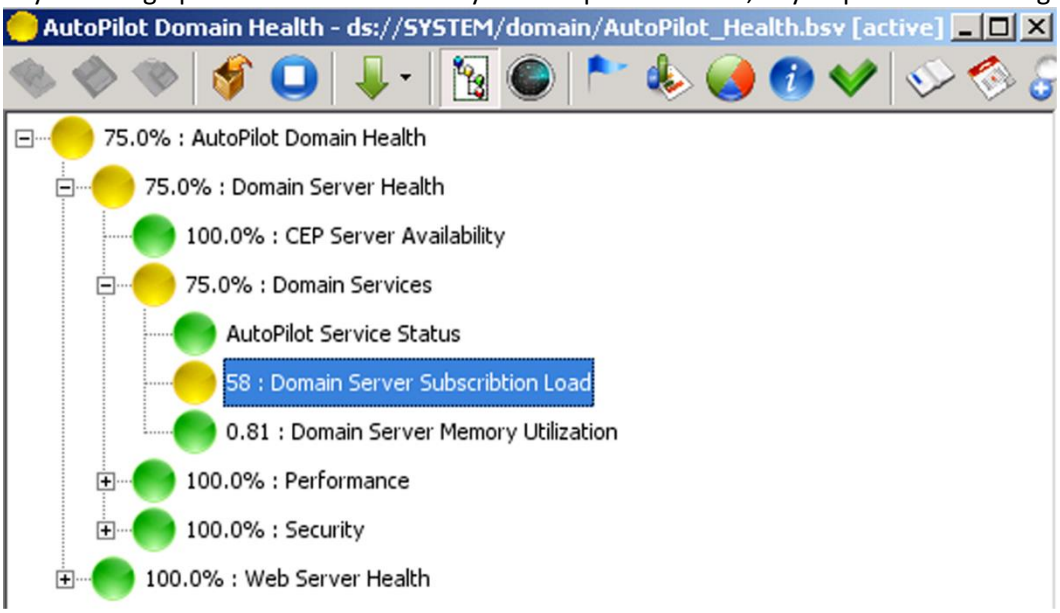


Figure 4-103. Monitoring a Working Business View

4.10.1.1 Event Logs

Event views in business views works exactly the same way as in the console. Open the *Event Viewer* to access the detailed information available. The events files you select will be displayed. It can be the file with events related to the business view displayed, or any other file.

Table 4-46. Business View Event Logs	
Property	**Description**

Severity	Depicted with the relative icon that represents the current health
Date/Time	Displays the date and time of the log entry
Source	Identifies the source of the entry as applicable
Event ID	Provide available event numbers
Account	User account ID for owner of event sources
Message	Contain descriptive information about the event. Contents will vary with the nature of the sensor.
Event Detail	Provide detailed information about the event, source, and status. To open the detail window double click the event you want to view. You can copy the detail by clicking the **Copy All** icon at the bottom right of the screen.

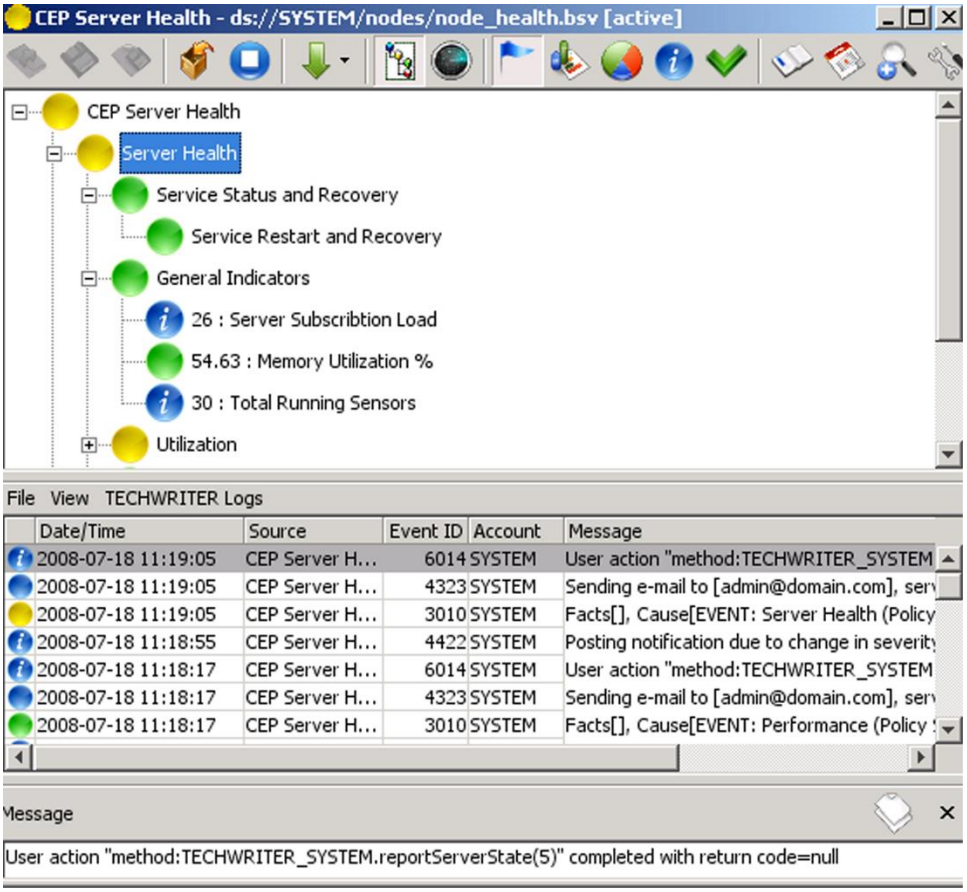


Figure 4-104. Business View Event Log

4.10.1.2.1 Charting Business View Metrics

IMPORTANT! If you want to be able to view the chart of your business view on the M6 Web Console, the logging option must be active in the expert. Refer to section 4.9.12, Maintaining Sensor History.

The charting option available in business views is different than those available in the console charting tool *Open Charting* . Value charting is the default.

NOTE A Business View may be running while adding sensors into the charting index

Drag and drop sensors into the charting index or click the **Add** button. The sensor will be assigned a color for charting.

Table 4-47. Business View Charting Tool

Property	Description
Source	**Real Time:** Plots from current sensor status in real time.
	History: Plots from local log file archives. If the logging option is not active in the expert or manager, there will be no history available.
Data	**Value:** Plots based on the numeric value of the facts monitored.
	Severity: Graphic will display to reflect the severity status of the sensor for the time defined.
	Health: Plot will reflect the health of the business view for the prescribed time frames.
Zoom	**Top:** Move to the right to scroll across the chart from left to right.
	Bottom: Use to expand or enlarge the charted are for a given time to give greater detail.
Type	There are seven charting formats available*: Step, Scatter, Bar, Mixed, Line, Area, and Area-Mixed*. Click the format that best suits your needs. Charting formats can be changed at any time without interrupting the charting process.
Add	Click to add selected sensor to the charting index.
Update	Click to update the chart in real time.
Period	The time intervals at the bottom of the draft represent the time line of the chart being viewed. Time can be measured in minutes, hours, days, or weeks. If the chart is from archived files it will display the times as relative to the file, not current time.
Legend	Correlates color to actual sensor being monitored.
Sensor	Displays sensor state, value, and name.
Value	Displays current sensor value rounded to the nearest hundredth.
Min	Displays the minimum value since charting was started.
Max	Displays the maximum value since charting was started.
Count	Displays how many samples were taken since charting was started.
Average	Displays the average value since charting was started.

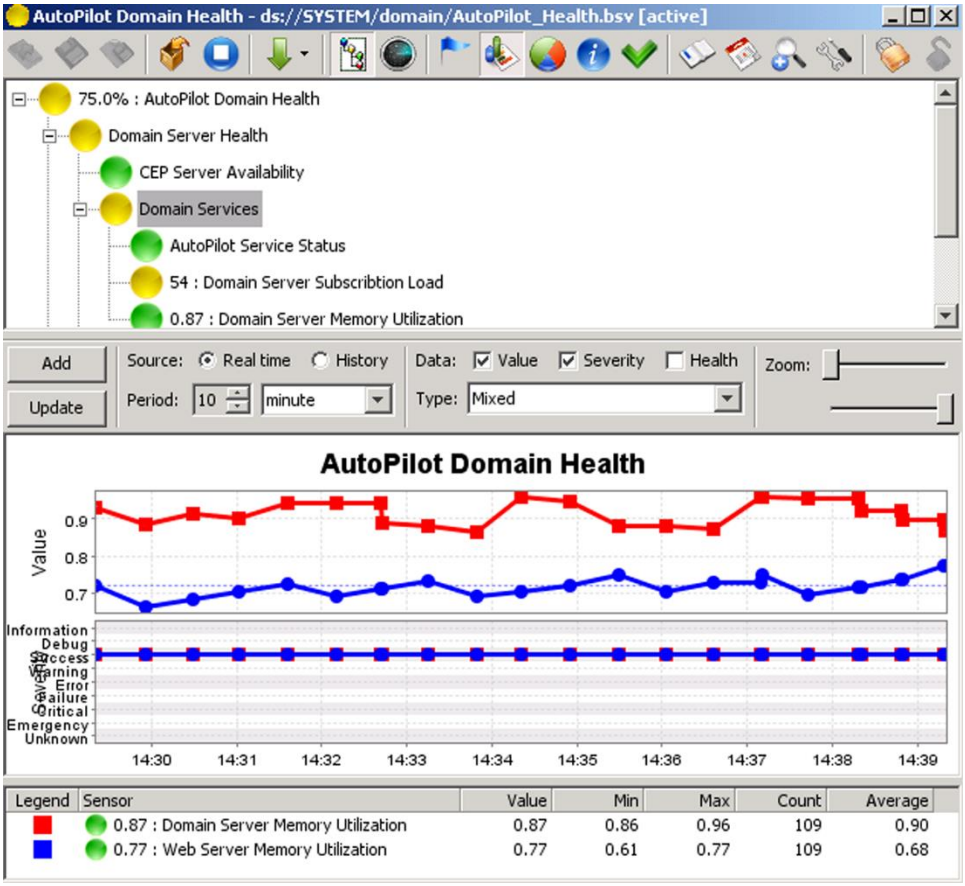


Figure 4-105. Business View Charting Options

Plotting Severity

All the sensors being charted with the same status will be represented by a single line on the chart. The look varies with the type of chart selected.

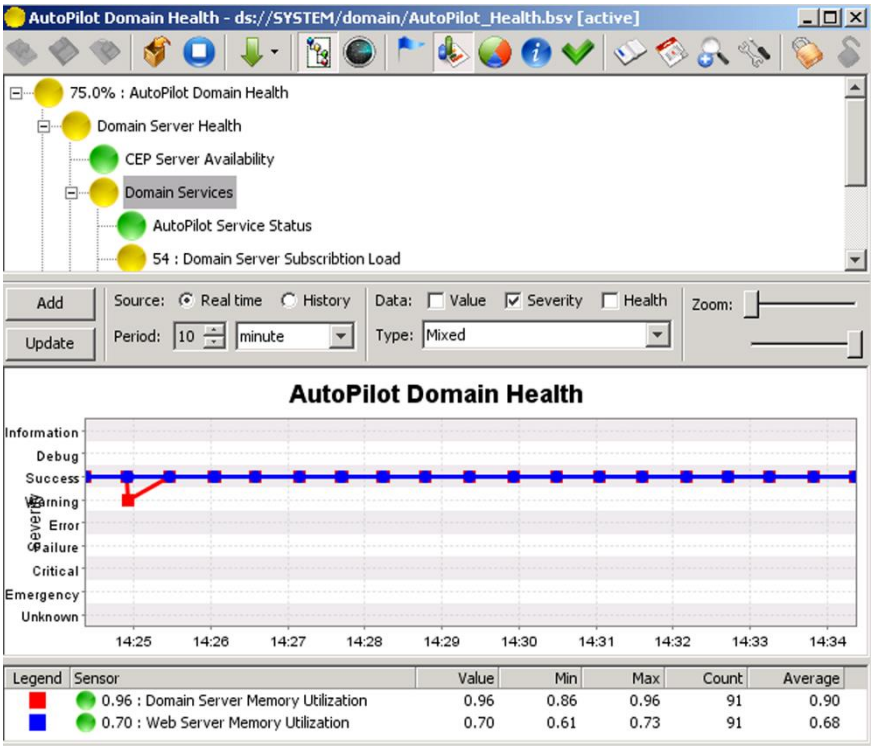


Figure 4-106. Severity Chart Format Samples

Plotting Health

As with severity the common health levels are grouped, the chart reflects the percentage of health. The percentages are defined in the health parameter in the sensor properties.

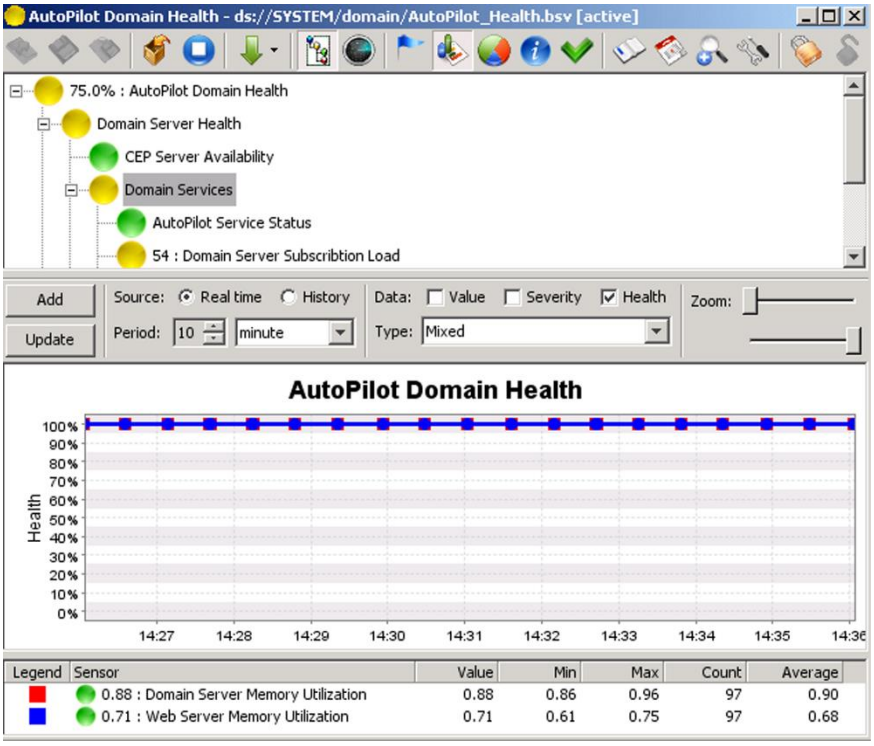


Figure 4-107. Health Chart Format Samples

Plotting Value

Charting with value is only effective with numeric sensor results. As you can see in the sensor index the Memory Utilization sensors are prefixed with a numeric value. Each sensors value is plotted in the chart as the chart refreshes.

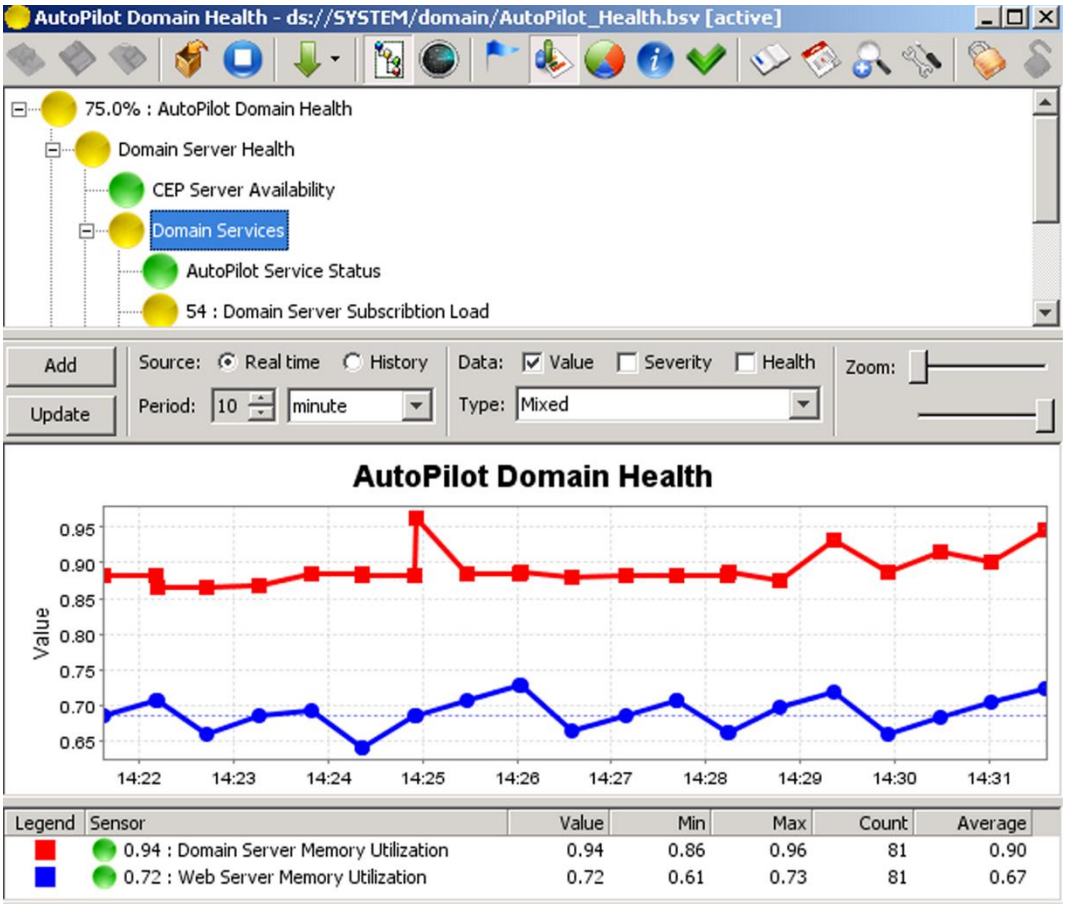


Figure 4-108. Value Chart Format Samples

4.10.1.3 Chart Options

Right click in the charting area of the screen to access the sub-menu. There are several additional options available to you there.

Table 4-48. Chart Options

Option	Description
Properties	Open the chart properties. The chart properties allow you to customize the look of your charts and alter the ranging options.
Save As	Captures the chart graphic and saves it to a user designated file.
Print	Prints the chart and legend to local printer.
Zoom In/Out	The zoom enables you to view the chart in greater or less detail.
Auto Range	Allows you to range in or out on both axes together or either horizontally or vertically, independently.

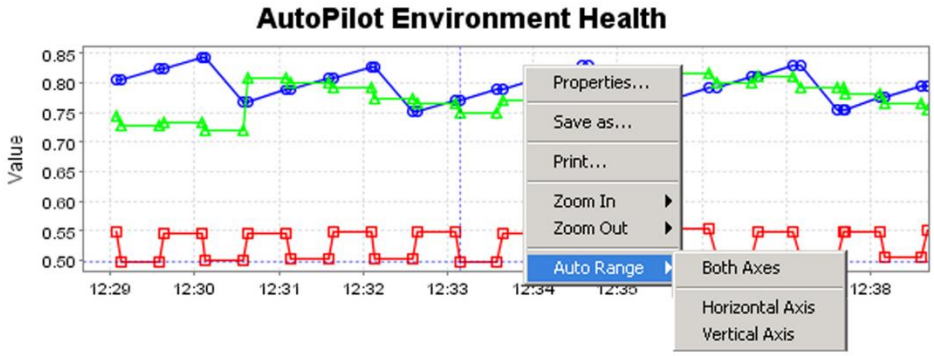


Figure 4-109. Chart Options

Hovering the cursor over any data point in the scattered, mixed, or area mixed chart mode will open a flag that displays relevant data and status. Clicking on that point will set crosshairs and allow you to zoom in or out from that spot.

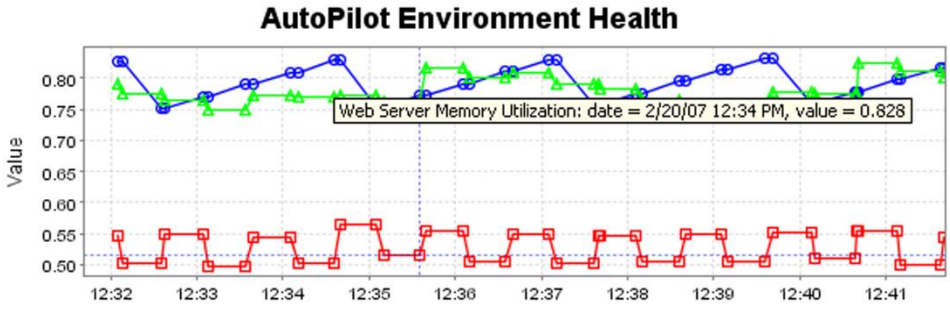


Figure 4-110. Charting Options

4.10.1.4 Sensor Overview

The Sensor Overview provides a summarized look at the business view configuration. It identifies key settings within the sensors that were defined when the sensor was developed or modified.

Table 4-49. Business View Sensor Overview	
Sensor	The identity and status of the sensor
Alert	Identifies if there are alerts specified in the sensor
Environment	Identifies if environmental variables are applied to the sensor
Ignore	Identifies if the sensor is to be ignore during any period
Logging	Identifies if the events are being logged.

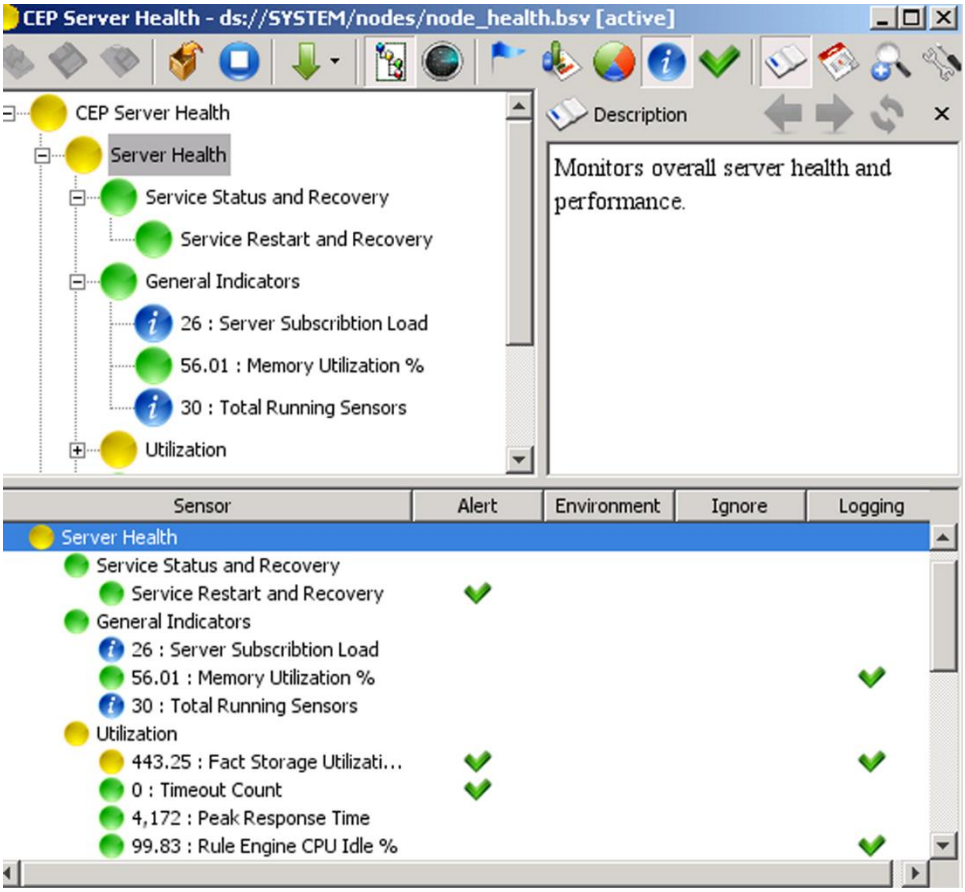


Figure 4-111. Business View Sensor Overview

4.10.1.5 Checking Sensor Integrity

Click the Integrity button to view the status of the business view and all of its sensors. Checks for common user errors associated with each sensor. The following conditions are checked:

- Sensor description is missing
- `[Field Name]` contains undefined environment variable(s)
- Fact `[#]` refers to an undefined service
- Fact `[#]` refers to an unavailable service
- Fact [#] included but not being used
- Fact [#] is undefined

If a sensor does not have an error, the description column displays a Success alert. If a sensor has an error, the description column displays a warning alert and the reason for the warning. Click **Check Integrity** to refresh the screen after making changes to the sensor.

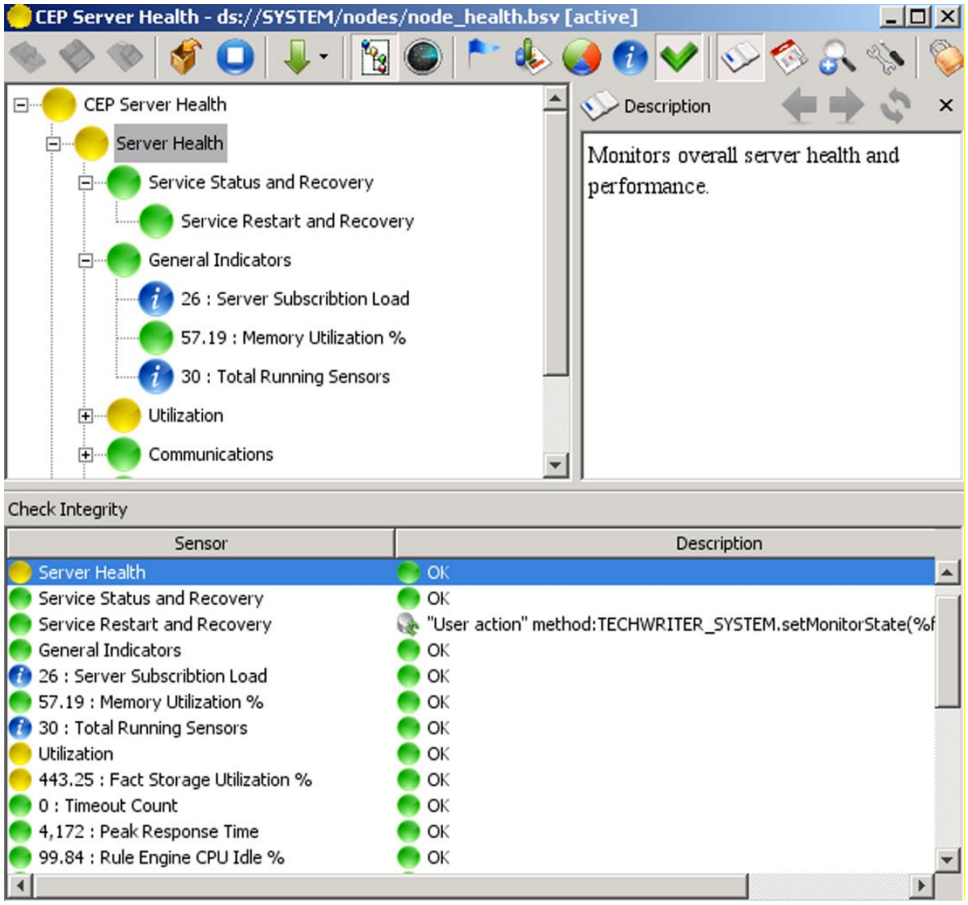


Figure 4-112. Business View Sensor Integrity

4.10.1.6 Sorting Temporary Dynamic Sensors

Click the **Sort Dynamic Sensors** button to sort temporary dynamic sensors by Name, Severity, Value, Health, and in Ascending Order and Descending Order. Only sensors selected as *temporary* from the sensor wizard can be sorted.

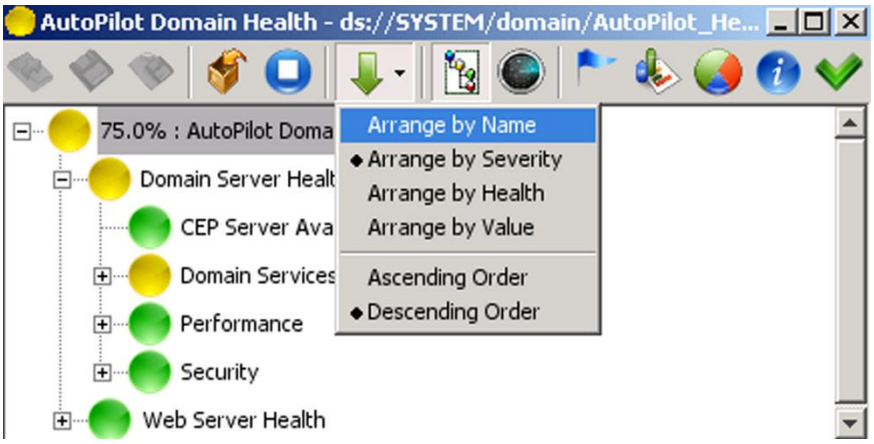


Figure 4-113. Arranging Dynamic Temporary Sensors

4.10.1.7 Showing Description

Display the **Description** to view the functional description of the sensor or business view. The developer generally provides the description. The description displayed is of the selected or highlighted item. By scrolling through sensors each unique description will be displayed.

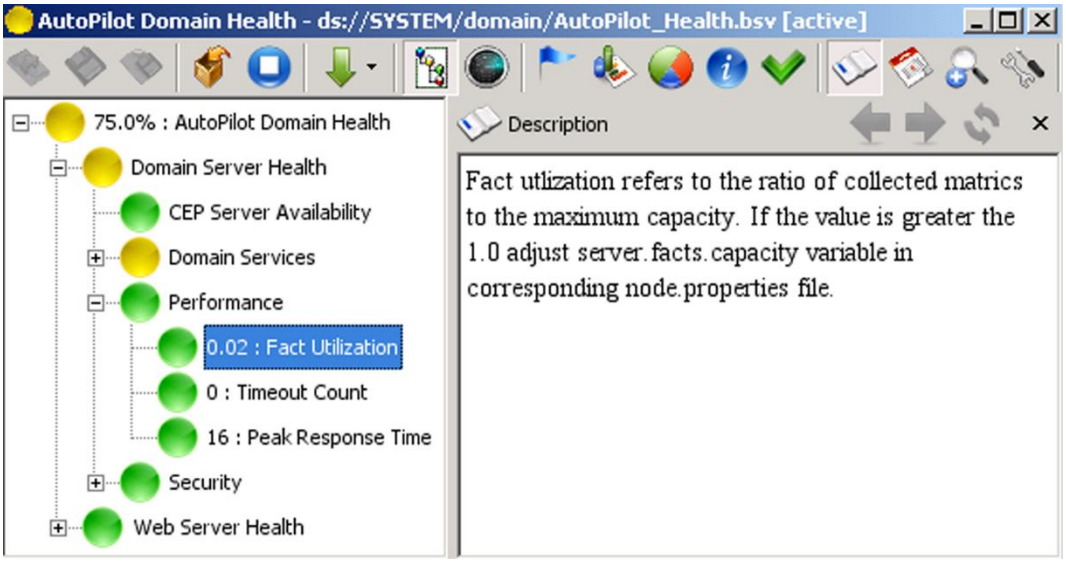


Figure 4-114. Business View Sensor Description

4.10.1.8 Showing Facts

Displays the **Facts** used in a given sensor. If the business view is currently active the facts displayed will reflect the real-time status. Real Time facts view is limited to 1024 display entries to offset display problems associated with too many entries. The facts displayed when the sensors are inactive reflect the last active state.

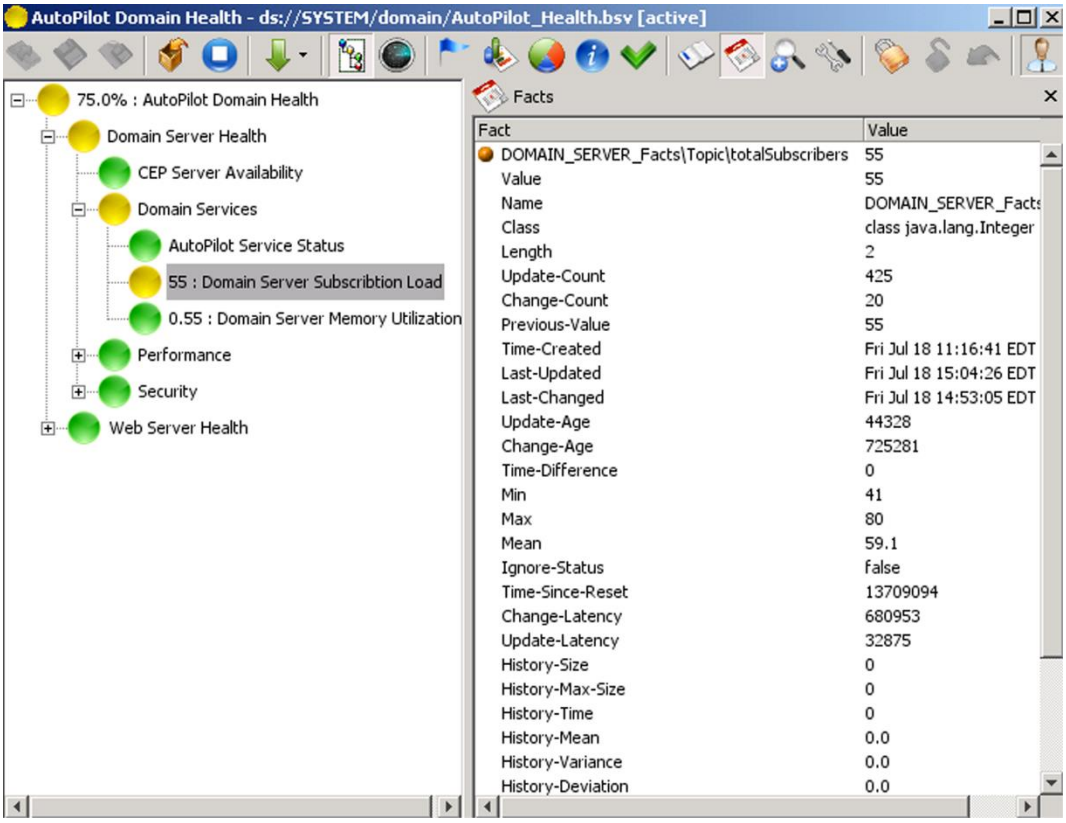


Figure 4-115. Business View Sensor Facts

4.10.1.9 Related Views

Open **Related Views** button to show all active business views that are related to the current view. Related views are those that monitor similar set of metrics. M6 servers perform this correlation automatically.

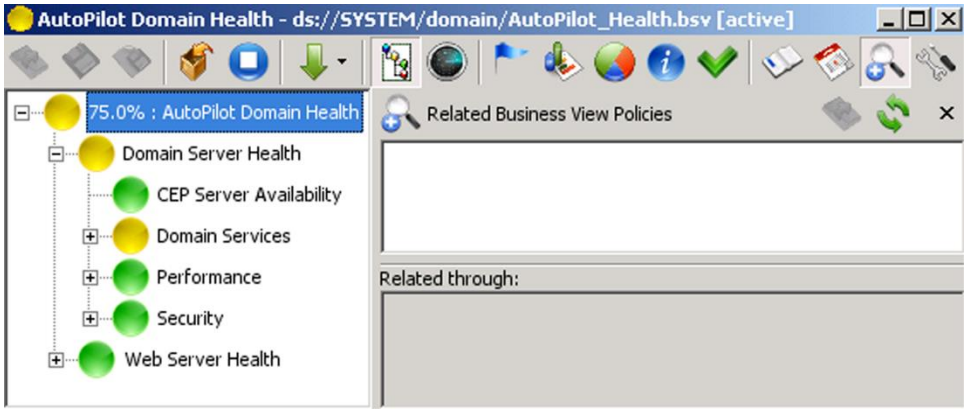


Figure 4-116. Related Views

4.10.1.10 Business View Policy Properties

Display the **Properties** of the business view/policy to review or change when needed by right clicking on the business view/policy, then click **Properties.** The Business view file is specified as: `ds://folder/file_name.bsv`, where `ds://` is a URL extension for M6 Domain Server. Use Business View explorer tool to create folders, save business views and deploy them to the appropriate manager(s). It is recommended that all business views are stored at the domain server and deployed from

the domain server out to the CEP servers. This reduces business view maintenance, improves performance and productivity. It also eliminates business view file synchronization issues.

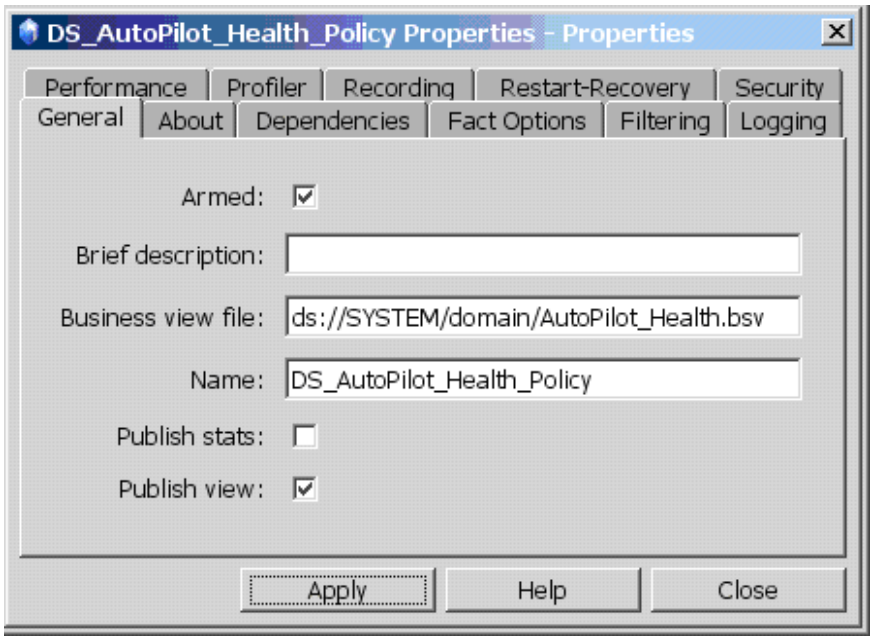


Figure 4-117. Business View Properties

4.10.1.11 Configuring Version Control

To prevent another user from editing your business view, you can lock and unlock your business view as needed. Click one of the following icons described below to control another user from overwriting your changes.

Table 4-50. Version Control

Icon	Description
	Locks and checks-out your document. The Check-Out command retrieves writable files from VSS and places them in your working folder, so that you can modify them. Document can only be opened by another user in Read-Only mode.
	Unlocks and checks-in your document. Document is saved and put back into VSS. To return the file to VSS, you can use the Check-In command to confirm your changes and copy your local, modified file into the VSS database.
	Reverses check-out and unlock. Cancels your check-out and removes the writable version of the file from your working folder.

4.11 Event Logging

Event logs contain data about the operation and status of your system. Business views record information into M6 event logs, on every alert or action that is taken on behalf of a user or user defined rules.

The logging data is based on the logging characteristics defined by the user while configuring experts, managers, policies, and business views. Event logs track system activities such as the start and completion of jobs, device status and health, system events, alerts, and notifications.

The collected system data is compiled and recorded in user defined M6 generated event logs. Event logs help you monitor and control system activity, analyze and correct problems. M6 installations maintain a

set of logs for all installed components in [*AUTOPILOT_HOME*]\logs directory. For more information, refer to Chapter 6: Troubleshooting Techniques.

The event logs have an .EVT file extension (example: DOMAIN_SERVER_SYSTEM$0.EVT).
Event logs have the *Logical* and *Physical Name*:

- **Logical Name:** User known name that is displayed in the event viewer title bar and in individual service Logging Properties. Default Logical Names are *System Unnamed*, and *Services*. There are system logs and user defined logs.
- **Physical Log Name:** Actual system stored file name. Each log file has a physical file name; the actual file is stored on the file system. The physical name convention is: [*MANAGED_NODE_NAME*]_[*LOGICAL_NAME*]$0|1.EVT

  (Example: DOMEGAX_SYSTEM$0.EVT). Since logs are circular, M6 may allocate two logs, for example: DOMEGAX_SYSTEM$0.EVT, when log is full, the $0 log is moved to DOMEGAX_SYSTEM$1.EVT and the original file is truncated (deleted and recreated).

System Logs

- **SYSTEM:** M6 system services record to this log.
- **SECURITY_ACTIVITY:** (Domain Server only) security service records information on security action. The log is created only when "service activity" is enabled on the DOMAIN_SERVER_SECURITY service.
- **ACTION_ACTIVITY:** Contains failed user actions
- **UNNAMED:** Should always be empty. This log is maintained when an event is written to a non-existent log. Any events recorded into this log indicate an internal error and should be reported to 515HNastel technical support.

User Defined Logs

- **SERVICES:** Where all management services record failures, errors, and warnings as well as trace messages.
- **User Defined:** The log name can be changed/created when you modify individual service's *Logging Properties*.


NOTE

All log files are stored as plain text files, viewable using any text editor.

The events in event logs are formatted as follows:

- **key:**{time=*timestamp*}{type=*5*}{evid=*id*}{account=*user*}{from=*component*} {event=*message*}. See *Historical Logs* for a detailed description of the event log format.
- The log is formatted to provide all the necessary data for the event displayed. Individual fields can be expanded to display the full detail, or you can use the detail window to display all event details from all fields.

Table 4-51. Event Log Properties

Field	Description
Severity	The related icon for the severity state or health is displayed.

Date/Time	The time and date of the log entry in year, month, day, hour, and minute.
Source	The source service the entry was received from.
Event ID	System defined Event number. User defined Event ID is defined when used.
Account	Identifies the account, which logged the event.
Message	The detail description of the logged event.
Message Detail	The event detail box provides full descriptive data for a selected event.

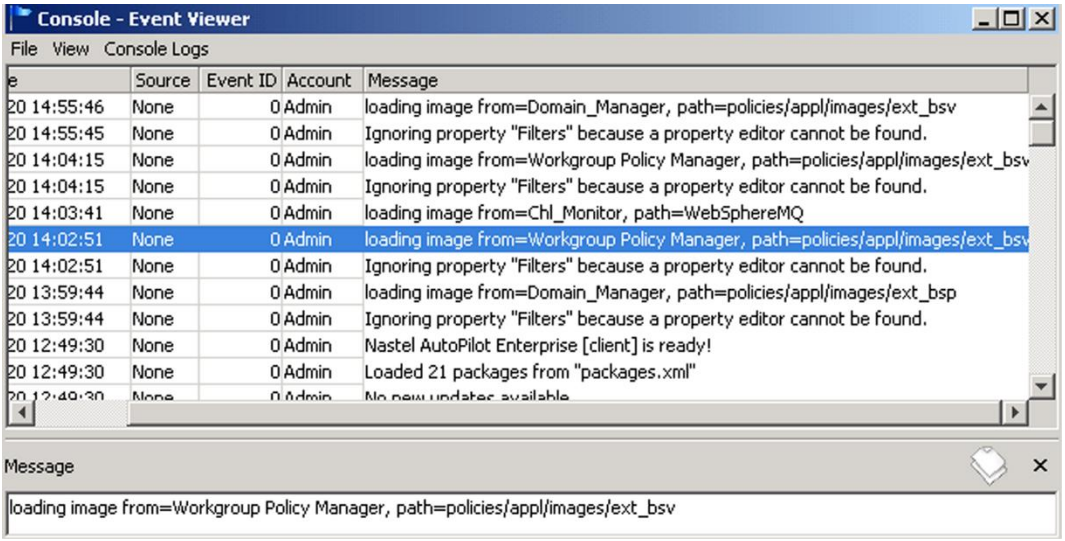


Figure 4-118. Sample Event Viewer and Log

The definable logging characteristics for managers, policies, and experts are common.

Table 4-52. Logging Characteristics

Property	Description
Audit	Select checkbox to enable the Audit Trace option. The audit trace is used when finite accountability of object, process, and user actions is required. The information collected will be logged in the event file specified in the associated experts.
Log Name	The name of the system assigned default event log where events are recorded. You should provide a logical name for each policy/manager to prevent the date from being logged in the generic system logs.
Log Service Activity	Click disable/enable button to disable or enable the log service activity. Disabling the activity will prevent the associated events from being logged in the event logs. If the logging for a manager is disabled only the Managers events will be blocked, if the expert logging is enabled the events generated will be logged in the file specified.
Log size	Log size in bytes. Real log size is the maximum value of server.log.size and logsize.

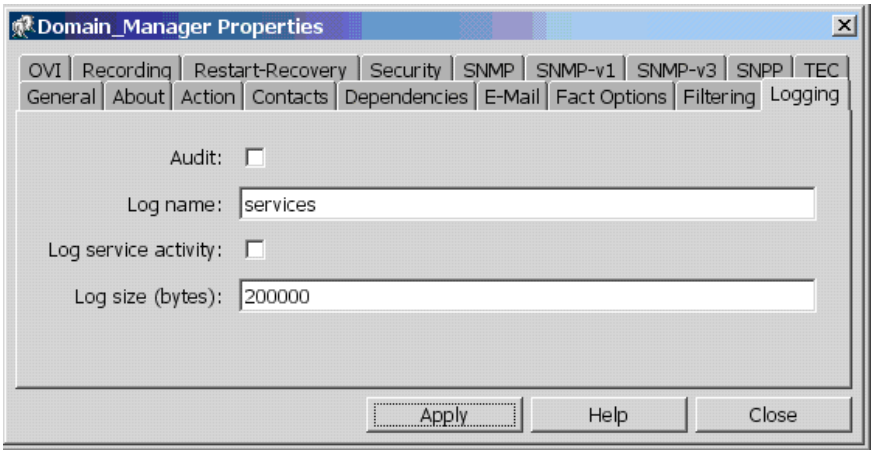


Figure 4-119. Logging Characteristics

4.11.1 Viewing Event Logs

Event logs are viewable from the event viewer, or the text files can be accessed with most text editors. Each business view logs all events, errors, triggers, and actions it executes into an M6 event log. To open the history of an event for a specific business view:

1. Open your business view.
2. Click **Event Viewer** to open the viewer at the bottom of the business view screen.
3. Click **File**, and then click **Open** to display the listing of available logs.
4. Click the log that matches the business view selected in step 1. The log will be displayed.

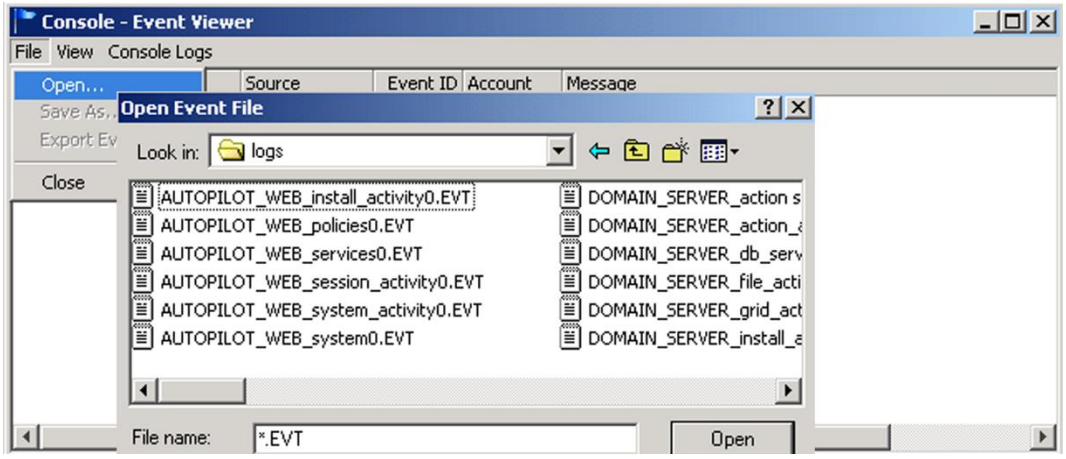


Figure 4-120. Opening Event Log in Event Viewer

The information presented in the events is the same as when viewed in a stand-alone event view or in the business view event viewer. Event data is formatted as follows:

Table 4-53. Event Logs

Property	Description
Severity	Depicted with the relative icon that represents the current health
Date/Time	Displays the date and time of the log entry
Source	Identifies the source of the entry as applicable
Event ID	Provide available event numbers

Account	Name of the account which logged the message or event
Message	Contain descriptive information about the event. Contents will vary with the nature of the sensor.
Event Detail	Provide detailed information about the event, source, and status. To open the detail window double click the event you want to view. You can copy the detail by clicking the **Copy All** icon at the bottom right of the screen.

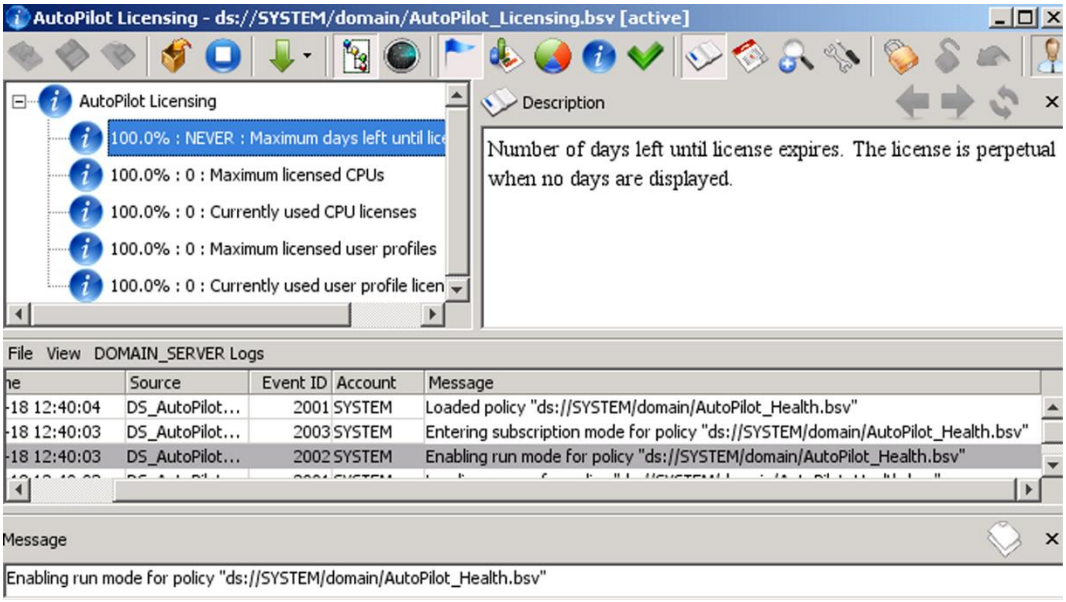


Figure 4-121. Sample View of an Event Log

4.11.2 Event Log Options

Saving and Exporting Log Files

The content of the event logs can be read in any text editor with the limitation that the time/date stamp is specified in milliseconds since OS specific date. Exported **.TXT** file contains events in comma-delimited format, where time is translated into absolute date-time. See sample entry below:

2003-05-08 17:21:56, time=1052428916619, type=8, evid=1001, account=Admin, from=Service, event=Starting service XYZ,

You may want to save a particular log for later use. You can use the *Save As* or *Export Events* functions to archive the current log.

Save As: With the event log open the file sub-menu and click **Save As**. When *Save Event As* screen is displayed, assign a file name to the log.

Export Events: With your event open or closed, open the *File* sub-menu, and click **Export Events**. The events exported will be a snapshot reflective of the (current) status of the event log.

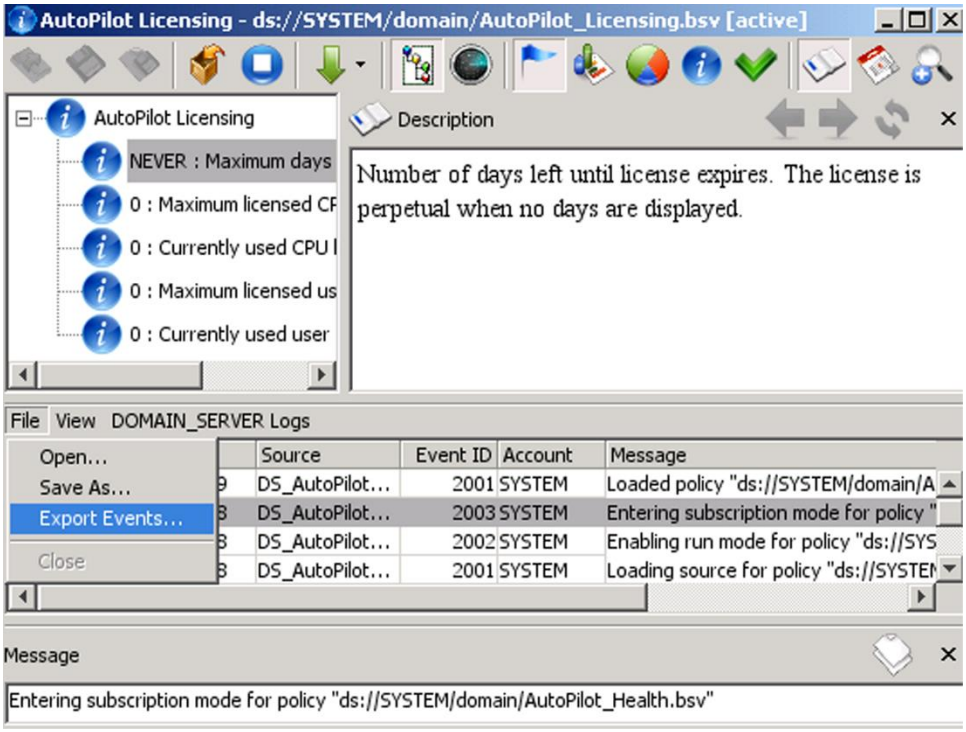


Figure 4-122. Event Log File Options

Viewing and Sorting Event Logs

Event Log viewers have various options that will allow you to customize your logs to best suit your needs. You can use filtering, sorting, and search tools to help identify and define log contents you need.

Event Log Viewing Options: The event log view can be altered to suit your needs by selecting from the following options:

Table 4-54. Event Log View Options

Property	Description
All Events	Displays all events in the log
Filter Events	Allow you to filter events to display only those events you require.
Newest First	Sort events by chronologically from newest to oldest
Oldest First	Sort events by chronologically from oldest to newest
Find Event	Use to locate events by severity, ID, and source.
Event Detail	Opens the event detail dialog box at the bottom of the event viewer. The viewer will remain open until manually closed. Provides detailed information about the event, source, and status. To open the detail window you can also double click the event you want to view.
Refresh	Manually refreshes the event log if active.
Auto-Refresh	Check the *Auto-Refresh* button to enable. The auto-refresh will remain active until turned off.

Figure 4-123. Event Log Option Menu

Event Filter: Use the filter to customize the log for your needs. You can filter a specific log or all logs using the parameters below:

Table 4-55. Set Event Filter

Property	Description
Type	Select severity/health levels to define filter.
From	Specify the date and time to search from. Can only be selected when not being initiated from an existing log.
To	Specify the date and time to search to. Can only be selected when not being initiated from an existing log.
Source	Select one or all event logs to filter for results.
Account	Account name. A blank field will reflect all accounts.
Event ID	Specify event IDs to filter. No entry will reflect all events. (Example: 1003)
Event Filter	Name based on event message type to be filtered (example: Starting*). "Starting" will list only those messages that begin with starting. No entry reflects all messages.

Figure 4-124. Set Event Filter

Find Events: Use this tool to search all event logs or a specific event log using the parameters below:

Table 4-56. Find Event	
Property	**Description**
Type	Select severity/health levels to define search criteria.
From	Specify the date and time to search from. Can only be selected when not being initiated from an existing log
To	Specify the date and time to search to. Can only be selected when not being initiated from an existing log
Source	Select one or all event logs to filter for results.
Event ID	Specify event IDs to filter.

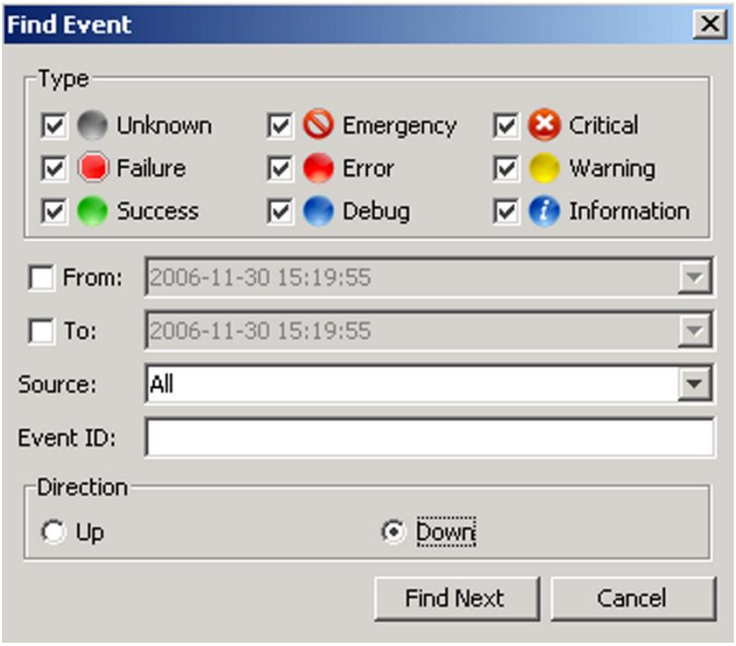


Figure 4-125. Find Event

Log Viewing and Setting Options

NOTE

Only one log can be opened in a viewer. If you need to view multiple logs, open additional instances of the event viewer.

The Console Log menu options are only available on the event viewer. This menu is not accessible from the business view event viewer.

Table 4-57. Log Viewing and Setting Options	
Property	**Description**
Event Log	The *Console Logs* menu lists the active event logs that are available for viewing. Click the log you want to view to open. (Example: SESSION_ACTIVITY, SYSTEM, UNNAMED).
Clear All Events	Use to clear your log of all events.
Log Settings	Use to set the maximum event log size, denoted in KB. Min= 100KB, Max= 10000KB (10MB)

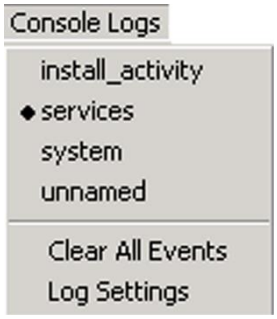


Figure 4-126. Console Logs

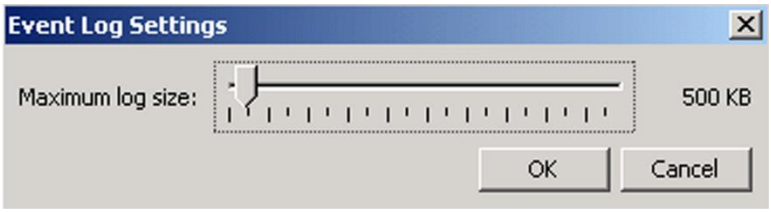


Figure 4-127. Log Size Settings

4.12 Performance Monitoring

4.12.1 Creating Chart Profiles

New charting profiles can be developed at any time. The performance and operation of M6 is unaffected by the development of profiles or performance monitoring.

1. Open a performance monitor by clicking the **Performance Monitor** button.

> **NOTE** The performance tool will only allow facts with numeric values to be listed in the table.

2. Select facts, one at a time, to be monitored from the hierarchal menu. Click **Add** to add the highlighted fact to the table. The monitor will create legend entries.
3. Repeat for each fact you want to monitor. Optionally, click **Delete** to delete a highlighted fact from the table. The monitor will delete the legend entry.
4. Set the refresh rate and the duration points. The refresh cycle can be set from one to 60 seconds Refresh: 20 seconds. The duration can be set from 50 to 300 points Points: 206. Each point represents a refresh cycle that will be displayed on the chart screen. (Example: Refresh Cycle of 10 seconds and Points set at 30 equal five minutes of charting time displayed). The time references at the bottom of the chart reflect the time frame of the current chart screen.
5. Save the chart profile if you intend to use it again. Give your chart profile a logical name that reflects its nature. If you close the performance monitor without saving the profile, it will not be saved.
6. Start by clicking the **Start Charting** button.
7. Check the *Available* column in the chart table to ensure the facts are active. Inactive facts, even though they are listed in the table and legend, will not be plotted. A green icon is displayed when the fact is available for charting. Items that are not available will not be charted.

8. If is necessary to make changes to an active chart you must pause the chart by clicking the **Stop Charting** button. The facts available status indicator will show red when paused. Make the change or addition and then restart when you have completed your changes.

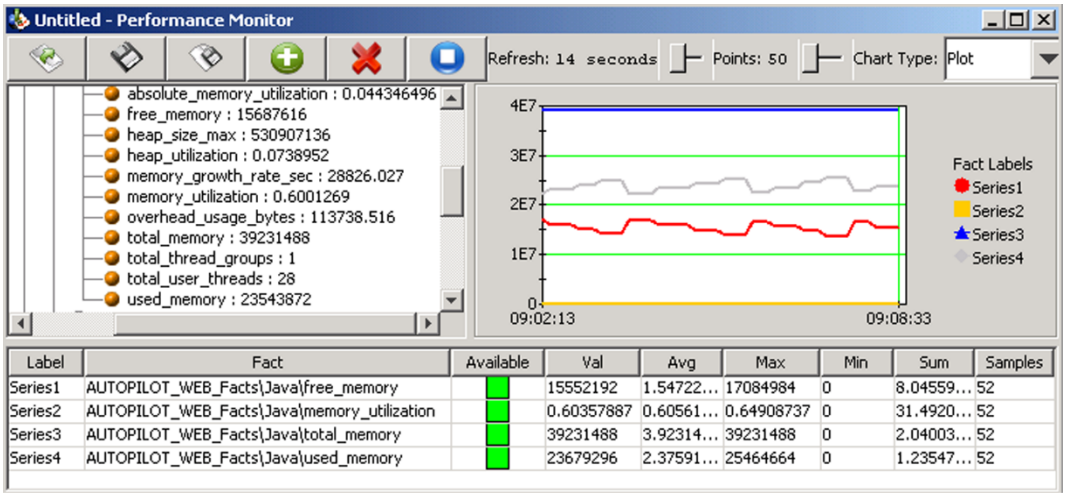


Figure 4-128. Typical Performance Monitor

4.12.1.1 Using Existing Chart Profiles

Use existing chart profiles to repeat monitoring the same performance statistics, or use an existing profile as the template for a new profile.

1. From M6 User Console open the Performance Monitor.
2. Click **Open Chart Profile** to access the menu. Select a chart profile file (.pmv). Click the **Open** button.
3. Make any required changes, add/delete facts, and change refresh or duration settings. Click **Add** to add the highlighted fact to the table. The monitor will create legend entries. To remove an item, click the fact, when it is highlighted, click **Remove** to delete item. The monitor will update the legend to remove the delete fact. Set the refresh rate and the duration points. The refresh cycle can be set from one to 60 seconds Refresh: 20 seconds. The duration can be set from 50 to 300 points Points: 206. Each point represents a refresh cycle that will be displayed on the chart screen. (Example: refresh cycle of 10 seconds and Points set at 30 equal five minutes of charting time displayed). The time references at the bottom of the chart reflect the time frame of the current chart screen.
4. If changes are made you should save the change before starting. Unsaved profile setting will be lost. Alternately, use **Save as** to create a new profile, preserving the original. Click **Save** or **Save As** to save the profile or create a new profile.

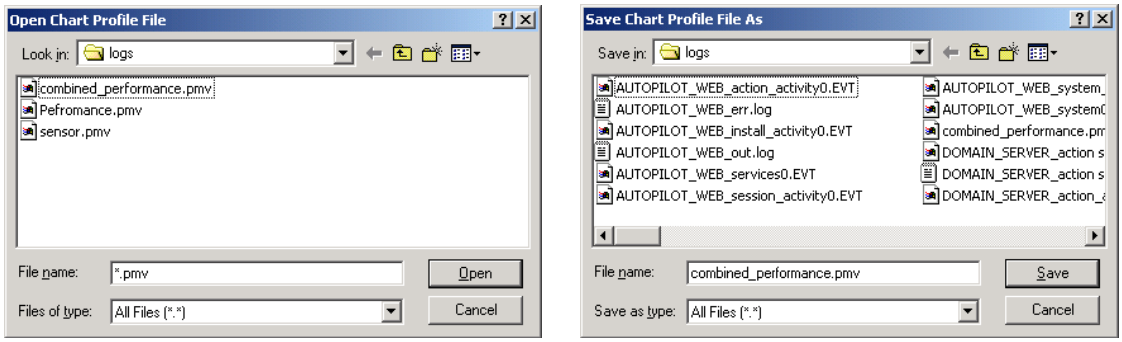


Figure 4-129. Open Chart Profiles and Save Chart Profile As

5. Click the **Start Charting** button to begin.
6. Check the *Available* column in the chart table to ensure the facts are active. Inactive facts, even though they are listed in the table and legend, will not be plotted. A green icon is displayed when fact is available for charting. Items that are not available will not be charted.
7. The chart will automatically stop when the user specified chart duration is complete.

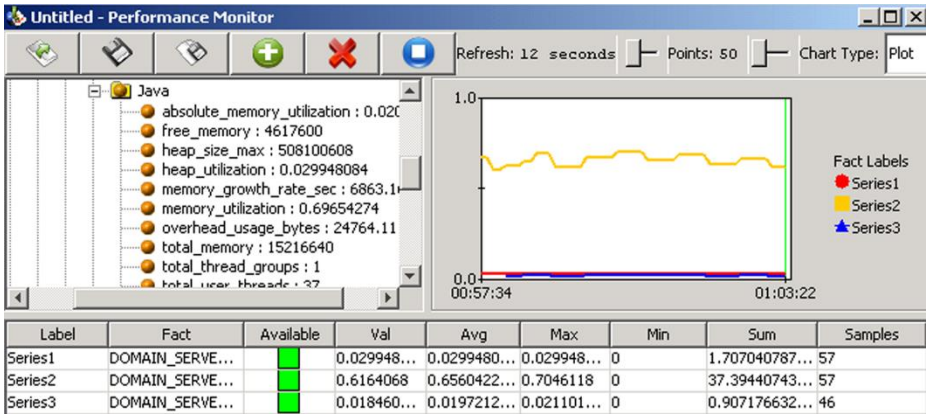


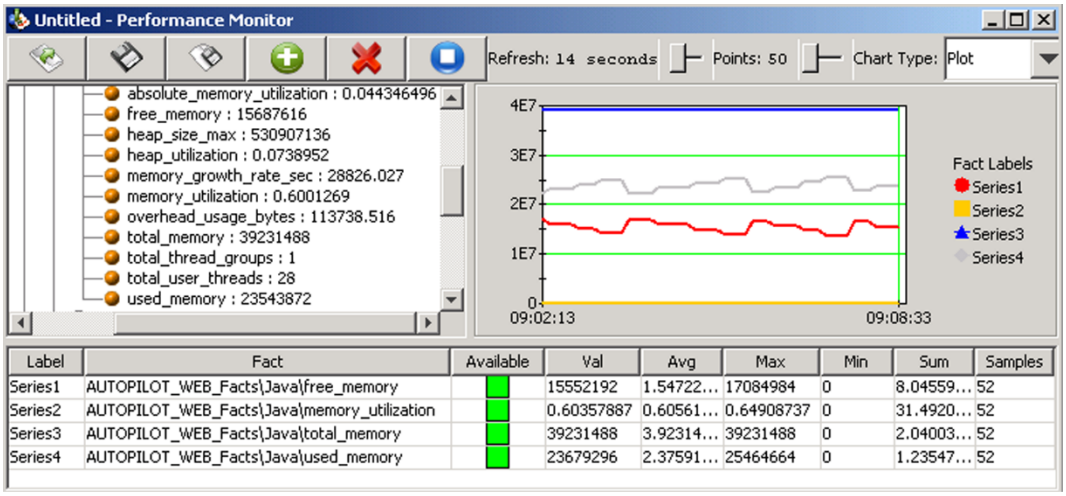
Figure 4-130. Sample Performance Monitor Charting

4.12.2 Monitoring Facts using Performance Monitor

The Performance Monitor starts plotting when activated. The subsequent plots will be posted at the interval you set when the *Refresh* cycle time was set. The total number of plots will be based on the points setting. The plot values are based on the average for the time between refresh cycles. The longer the time between refreshes the less detailed the chart. Short duration events such as memory or processor usage spikes will be averaged and may not show the event in the graph. When short refresh cycles are used the detail gives a more accurate depiction of actual events.

The numeric charting parameters are taken from the facts selected for monitoring. The chart will adjust for each fact added to the table. In the sample below the right side of the plot was set to refresh every four seconds, the left side every 30. The left side shows progressive changes while the right indicates a more accurate performance measurement.

Time on left denotes the time the expiring segment or beginning of chart was recorded. Time on right specifies current time (based on internal clock of host machine). The time span between the noted times is the duration set when you set the refresh and the points.



Label	Fact	Available	Val	Avg	Max	Min	Sum	Samples
Series1	AUTOPILOT_WEB_Facts\Java\free_memory		15552192	1.54722...	17084984	0	8.04559...	52
Series2	AUTOPILOT_WEB_Facts\Java\memory_utilization		0.60357887	0.60561...	0.64908737	0	31.4920...	52
Series3	AUTOPILOT_WEB_Facts\Java\total_memory		39231488	3.92314...	39231488	0	2.04003...	52
Series4	AUTOPILOT_WEB_Facts\Java\used_memory		23679296	2.37591...	25464664	0	1.23547...	52

Figure 4-131. Performance Chart (Plot Format)

Results of monitoring are posted in chart table. They reflect cumulative values since chart was initiated.

Label	Fact	Available	Val	Avg	Max	Min	Sum	Samples
Series1	DOMAIN_SER...		3984944	3510179.2	4099040	0	1.7550896E7	5
Series2	TECHWRITER...		1649032	1659010.0	1797448	0	6636040.0	4
Series3	TECHWRITER...		4947.2	-3813.666750...	6820.533	-13511.2	-15254.66700...	4
Series4	TECHWRITER...		0.66919005	0.6671883625	0.7207304	0	2.66875345	4

Figure 4-132. Chart Table

There are five charting formats available*: Plot, Bar*, *Stacking Bar*, *Area*, and *Stacking Area*. Click the format in *Chart Type* that best suits your needs. Charting formats can be changed at any time without interrupting the charting process.

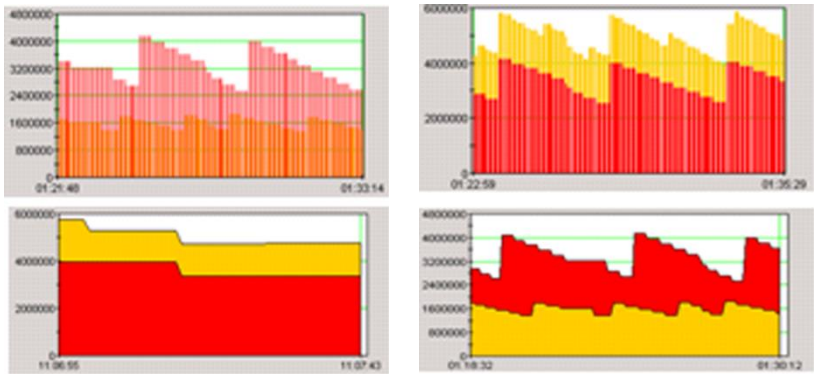


Figure 4-133. Display Chart Formats

4.13 Searching AutoPilot M6 Domain

Search Tool can be used to locate registered management services such as experts, managers, and policies service etc. deployed across the M6 domain based on user-defined search criteria. Search will also look within managers for content if option is selected.

Table 4-58. Search Tool Properties	
Property	**Description**
Search Parameters	
Name	Enter exact name of Object to be found.
Description	Enter a description of the object to be found.
Role	Define maximum file size expressed in Megabytes (example: 7.3 MB).
Property	You can search by using one of the following properties: node, grid_enabled, grid_name, updated, status, auto start, stream facts, stream derived metrics, description, uniquename, resourceHandle, resourcePath, popup_menu_class, popup_window_class, oid, class, roles, owner, and umask. Wildcard (*) can be used as appropriate.
Context	Select the context from the sub-menu. The available context entries reflect the actual context of the existing objects.
Type	Select the type from the sub-menu. The available type entries reflect the actual type of the existing objects.
Find managers with such contact	Check the box to perform an internal search of Managers to locate additional objects.
Search Result Table	
Name	Names object found in the search.
Context	Specifies context of object found.
Type	Identifies the type of object found in the search.
Node	Identifies CEP server or domain server where object was located.
Owner	Identifies owner of object found in the search.

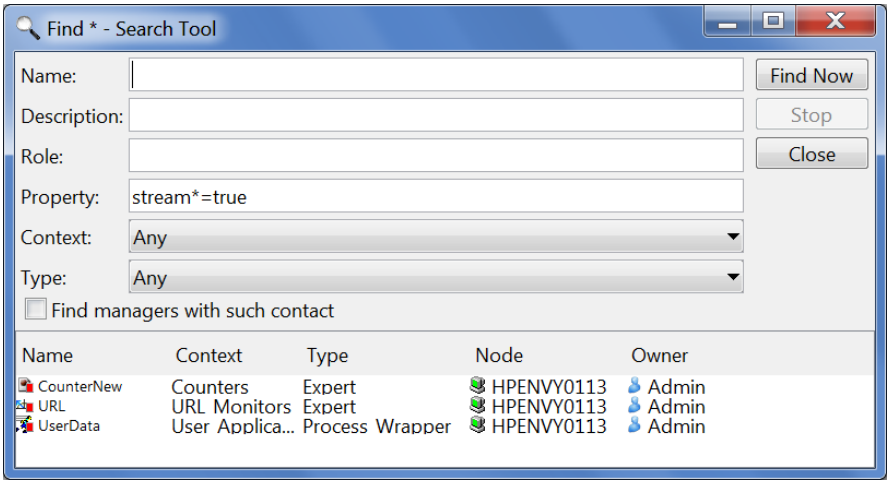


Figure 4-134. Search Tool and Results

Right click any object listed in the results table to access context-sensitive menu.

4.14 Forwarding Events to OVO from Business Views

> **NOTE** An understanding of M6 Business Views and HPOV operations is required to use this section of the manual. Please refer to the Business View sections of this manual and to the pertinent HP OVO documentation for details on these subjects.

Alerts generated by the business views can be forwarded to OVO in several methods. One of these methods is described below.

As shown above, an opcmsg command is issued by the business view which is received by the OVO Agent and formatted as per the relevant message template. Message templates typically match the parameters in the issued opcmsg and/or the pattern in the message text and format it for display and/or issue actions to notify the relevant personnel and generate trouble tickets.

The typical steps are as follows:

1. Sensors in M6 business views generate alerts based on the monitored facts.
2. The sensor passes certain variables and calls the 'opcmsg' command with certain parameters, either directly or through a script. These parameters include severity, object type, application, and message text. The object type and application are based on the facts being monitored. The name of the sensor is usually passed as the message text.
3. The OVO agent running on the CEP server intercepts this call and matches it with the message templates deployed on the node.
4. If the message pattern and parameters match with any of the conditions in the template, then the message is formatted for display on the browser (active or history); used to auto-acknowledge a previous event; highlight the service tree; notify the responsible parties and/or open a trouble ticket.

Use the message template defined by the file bsv_interface.dat. This is a message template that can be used by alerts generated from business views. Please contact your OVO administrator for procedures and guidance.

4.14.1 Issue opcmsg with a Message Text

You can issue an opcmsg with a message text in one of the following formats:

For Agent related alerts:

```
AP_BSV::<*.gm>::<*.node>::<*.objtype>::<*.property>::<*.value>::REASON
```
Ex: `AP_BSV::GM::NODE::wmqagt::state::disconnected::EXRC_AGENT_DISCONNECTED`

The possible reason codes are:

EXRC_AGENT_DISCONNECTED
EXRC_AGENT_CONNECTED

For Queue Manager related alerts:

```
AP_BSV::<*.gm>::<*.node>::<*.objtype>::<*.property>::<*.value>::REASON
```
Ex: `AP_BSV::GM::NODE::qmgr::state::Inactive::EXRC_QMGR_STOPPED`

The possible reason codes are:

EXRC_QMGR_ACTIVE
EXRC_QMGR_STOPPED

For Object related alerts:

```
AP_BSV::<*.gm>::<*.node>::<*.qmgr>::<*.objname>::<*.objtype>::<*.property>:
:<*.value>::REASON
```

Ex: `AP_BSV::GM::NODE::QMGR::APPL.INPUT.QUEUE::queue::Curdepth::15::QDEPTH_THRESHOLD_REACHED`

The possible reason codes are:

EXRC_CMDSRVR_STARTED
EXRC_CMDSRVR_STOPPED
MQRC_CHANNEL_STARTED
MQRC_CHANNEL_STOPPED_ERROR
MQRC_CHANNEL_STOPPED_RETRY
MQRC_CHANNEL_STOPPED_OK
MQRC_CHANNEL_STOPPED_DISABLED
MQRC_Q_DEPTH_FULL
MQRC_Q_DEPTH_HIGH
MQRC_Q_DEPTH_LOW
QDEPTH_THRESHOLD_REACHED
QDEPTH_THRESHOLD_NORMAL

4.14.2 Examples of Sending Alerts

The following are examples of sending an alert when a queue reaches a depth threshold. The alert screen below is from a dynamic sensor that looks at the curdepth fact and generates an alert if the depth is greater than n.

Example 1: The alert will be sent to OVO and can be manually acknowledged by the operator.

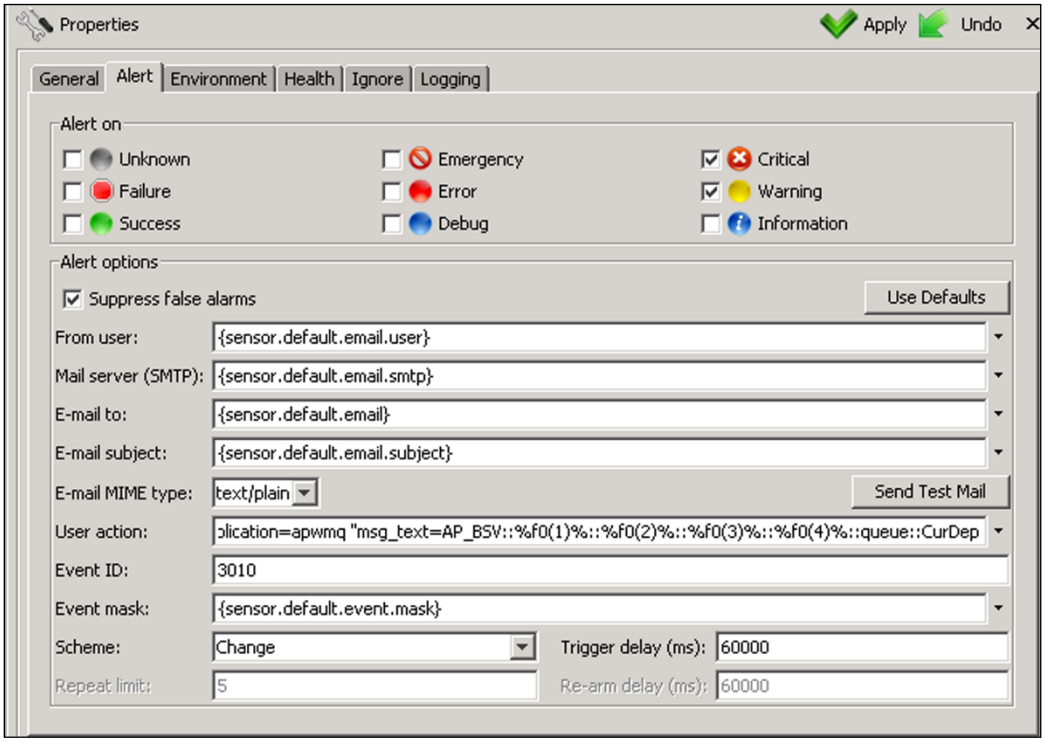


Figure 4-135. Example of Queue Depth Manual Acknowledgement Alert Screen

The possible severities are Critical and Warning. The following alert is issued:

```
opcmsg severity=%sevstr% object=queue application=apwmq
"msg_text=AP_BSV::%f0(1)%::%f0(2)%::%f0(3)%::%f0(4)%::queue::CurDepth:
:%value%::QDEPTH_THRESHOLD_REACHED"
```

Example 2: Alerts will be auto-acknowledged when the business view turns to success.

The business view in the above example can be modified to issue the following alert:

```
object_alert_ovo.bat %sevstr% %f0(1)% %f0(2)% %f0(3)% %f0(4)% queue Curdepth
%value% QDEPTH_THRESHOLD_REACHED
```

The script object_alert_ovo.bat converts the incoming parameters to severity, application, object, and message text.

The script follows:

```
REM *
@echo
REM *
set SEVSTR=%1
set GM=%2
set NODE=%3
set QMGR=%4
set OBJECT=%5
set OBJ_TYPE=%6
set PROPERTY=%7
set VALUE=%8
set REASON=%9

REM *** If business view issues SUCCESS Severity and REASON must be changed
accordingly*
IF %SEVSTR%==SUCCESS goto CHG_REASON
goto ISSUE_ALERT
```



```
:CHG_REASON
set SEVSTR=NORMAL
IF %REASON%==EXRC_CMDSRVR_STOPPED set REASON=EXRC_CMDSRVR_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_ERROR set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_RETRY set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_OK set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_CHANNEL_STOPPED_DISABLED set REASON=MQRC_CHANNEL_STARTED
IF %REASON%==MQRC_Q_DEPTH_FULL set REASON=MQRC_Q_DEPTH_LOW
IF %REASON%==MQRC_Q_DEPTH_HIGH set REASON=MQRC_Q_DEPTH_LOW
IF %REASON%==QDEPTH_THRESHOLD_REACHED set REASON=QDEPTH_THRESHOLD_NORMAL

:ISSUE_ALERT
echo opcmsg severity=%SEVSTR% object=%OBJ_TYPE% application=apwmq

msg_text="AP_BSV::%GM%::%NODE%::%QMGR%::%OBJECT%::%OBJ_TYPE%::%PROPERTY%::%VA
LUE%::%REASON%"
```

The method described above is only one of several methods. You should modify it per your own requirements.

4.15 Registry Tool

The RegistryTool is a Java application that facilitates migration of registry environment from one AutoPilot deployment to another. The tool comes bundled with AutoPilot M6 product (under bin directory).

4.15.1 Requirements

- JDK6 or higher
- AutoPilot M6 SU19

4.15.2 Launching the Tool

1. Change current directory to *AUTOPILOT_HOME\bin*
2. Execute: *java -jar regtool.jar –gui.*

4.15.3 Importing Services (Add to Registry Button)

- Selected services are imported into AutoPilot's registry file which is located in `AUTOPILOT_HOME\localhost\registry.xml`.
- Before being updated, the registry file is backed up in `AUTOPILOT_HOME\localhost\registry.mbk`.
- Imported artifacts are placed in `AUTOPILOT_HOME\localhost\import` directory.
- When the CEP server starts up, the artifacts are pushed to the CEP server's Domain Server.

IMPORTANT! CEP server needs to be shut down before performing the import operation.

4.15.4 Exporting Services (Export to Archive Button)

Selected services may be exported to an archive file (*.pkx). Artifacts referenced from the services are also placed in the archive. If artifacts reside on Domain Server, the user will be prompted for Domain Server username/password.

IMPORTANT!

The Domain Server defined in AUTOPILOT_HOME/global.properties needs to be up before performing export operation.

4.15.5 Artifacts

Artifacts are business views, business processes, and pxml files that are referenced from services in the registry. Physically, artifacts are located either on the Domain Server (for registry files) or in the registry archive.

Artifacts are exported/imported only as part of a service from which they are referenced. For instance, when exporting selected services into an archive file, only those artifacts that are referenced from these selected services will be exported. Likewise, on importing selected services into registry, only the artifacts that are part of these services will be imported.

4.15.6 Connecting to Domain Server

Referenced from registry services, artifacts actually reside on the Domain Server. When exporting artifacts into an archive, the tool needs to connect to the Domain Service to fetch these artifacts. The connection URL to the Domain Service is based on *domain.server.url* property found in AUTOPILOT_HOME/global.properties. The user will be prompted for Domain Server's username and password for Domain Server authentication.

4.15.7 Duplicate Services

If a service that is selected to be imported into registry has the same name as a registry service, the selected service is considered to be a *duplicate*. Before performing the import operation, the tool checks for duplicates and if found, prompts the user to choose whether to overwrite the registry entries with the selected services or to leave the registry entries untouched. If overwriting a registry, a duplicate service with artifact(s) will also overwrite the registry's artifact(s).

4.15.8 Command-line Mode

In command-line mode the tool may be invoked from a script or command prompt. To see available commands and options, execute *java –jar regtool.jar*.

Command options:

-version

Display the tool's build version

-export filename

Export registry to specified archive file. All registry services and their corresponding artifacts will be exported. If the registry contains artifacts, then Domain Server username and password need to be specified. (Refer to –user and –password options.)

-import filename

Add to registry services from specified archive file. All services and artifacts contained in the specified archive file will be imported to the registry. If the archive contains services/artifacts that are already present in the registry, the registry's entries will be overwritten.

[-user username]

Domain Server username

[-pwd password]

Domain Server password

4.16 Streaming Data

To stream to Nastel XRay, you can use https://xray.nastel.com or your local XRay installation.

4.16.1 Stream Data

In order to stream data, modify the global.properties file as follows:

1. Specify the access token
2. Optionally specify the data center and the geo location
3. Specify the URL of the Nastel XRay Service.

```
; AutoPilot Insight specific options
; set following to true to enable shutdown statistics dump
property tnt4j.dump.on.vm.shutdown=false
property tnt4j.dump.on.exception=true
property tnt4j.dump.provider.default=true
property tnt4j.config={autopilot.home}tnt4j.properties
property tnt4j.token.repository={autopilot.home}tnt4j-tokens.properties
property tnt4j.source.DATACENTER=your_datacenter  <- 2
property tnt4j.source.GEOADDR=your_geo  <- 2
property tnt4j.source.hosturl=https://data.jkoolcloud.com  <- 3
property tnt4j.source.access.token=your_accesstoken  <- 1
```

These settings along with those specified on the expert properties and tnt4j.properties are used to control the streaming options.

4.16.2 Stream AutoPilot Facts

In order to stream AutoPilot facts (metrics), you configure the destination and which facts to stream.

1. Right-click on any AutoPilot service such as an expert or a policy manager and select **Properties** to open the Properties dialog box. You can also set these properties when deploying a new service.

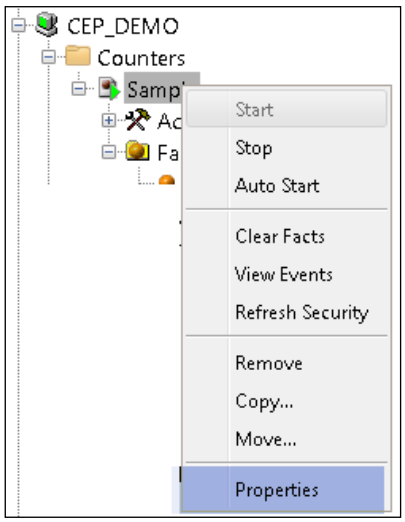


Figure 4-135A. Service Properties

2. On the **Streaming Options** tab, configure the options required as described below. The **Streaming Options** tab controls which facts get streamed and the location they stream to. Many of the fields are optional and have default values provided in the **`tnt4j.properties`** file that is part of the Nastel XRay installation package. However, they can be overridden.
   a. For streaming the facts as they update, select **Stream Facts**.
   b. For streaming periodic derived metrics, select **Derived Metrics** and set a frequency by entering a time interval in milliseconds in the **Interval of derived metrics** field.

      NOTE: It is recommended to set the exclude/include filter to control which facts are streamed.

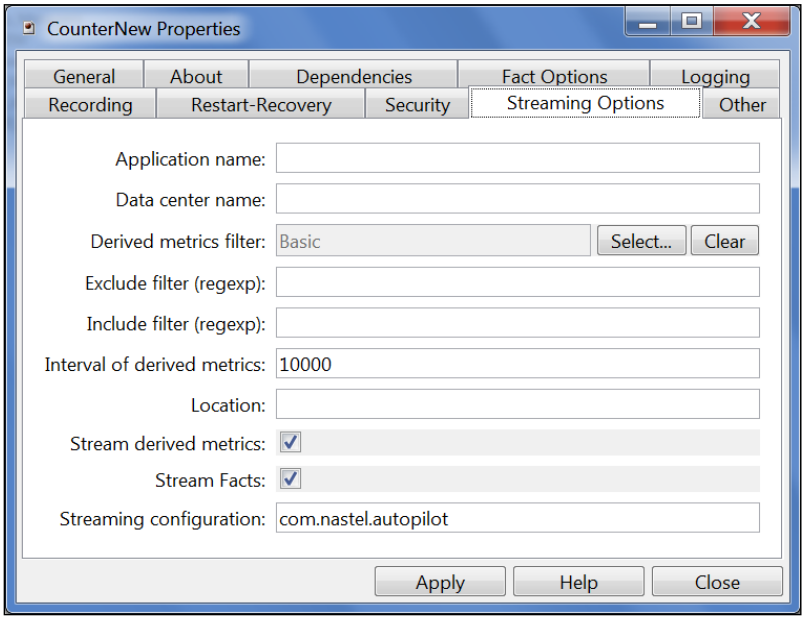


Figure 4-136. Streaming Options

Table 4-59. Streaming Options Properties

Property	Description
Application name	Sets application name if different from the default set in the **`tnt4j.properties`** file.
Data center name	Sets data center name if different from the default set in the **`tnt4j.properties`** file.
Derived metrics filter	Click **Select** to select an existing filter or create a new one. (See Figure 4-137.)
Exclude filter (regexp)	Ignore facts that match specified regular expression; that is, do not stream facts that match the regexp.
Include filter (regexp)	Only stream the facts that match specified regular expression.
Interval of derived metrics	Time interval, in milliseconds, to send fact derived metrics.
Location	Sets server location if different from the default set in the **`tnt4j.properties`** file.
Stream derived metrics	Enable/disable derived metrics streaming. System facts are generated: total_derived_fact_processed and total_derived_fact_sent.
Stream Facts	Enable/disable fact streaming (requires TNT4J streaming framework). System facts are generated: total_fact_processed and total_fact_sent.
Streaming configuration	Indicates where the data streams. This value must match a stanza in the **`tnt4j.properties`** file. The default is **com.nastel.autopilot**.

Create a New Filter

Filters are used to specify which derived metrics you want to stream.

1. After clicking the **Select** button from the **Streaming Options** tab (Figure 4-136), the *Filters* dialog box is displayed. Click the plus sign icon to display the *New Filter* dialog box (Figure 4-138).

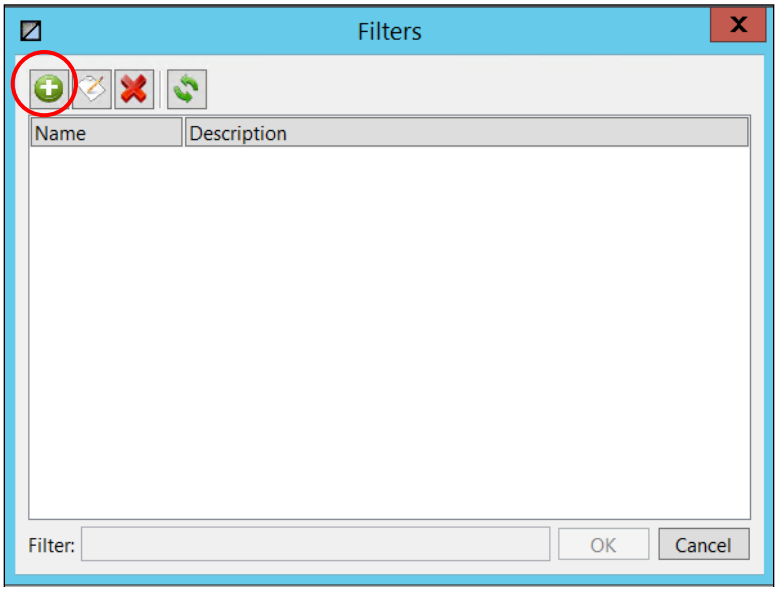


Figure 4-137. Create a Filter

2. From the *New Filter* dialog box:
   - **Filter name** – Enter a name for the filter.
   - **Derived metrics** – Select **General**, **History**, **Statistics**, or **All** to get a list of related metrics.
   - **Description** – Enter a description for the filter.
   - **Available metrics** – Select a metric and use the **Add** button to move it to the **Selected metrics** column on the right. You can use **Add all** to move all the metrics. Similarly, you can use **Remove** and **Remove All** buttons to remove metrics. (Refer to Appendix H for a list of derived metrics.)

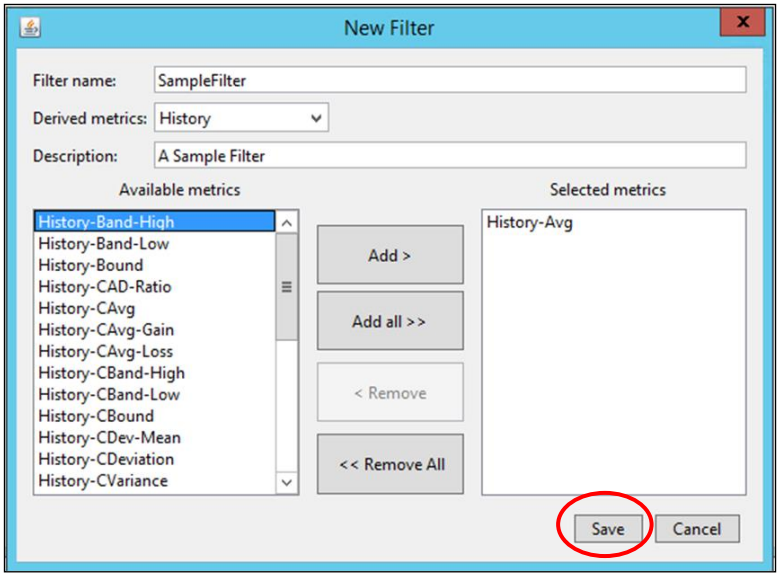


Figure 4-138. New Filter

3. Click **Save** to save/create your filter. It is now listed on the *Filters* dialog box (Figure 4-139) and can be selected for future use.

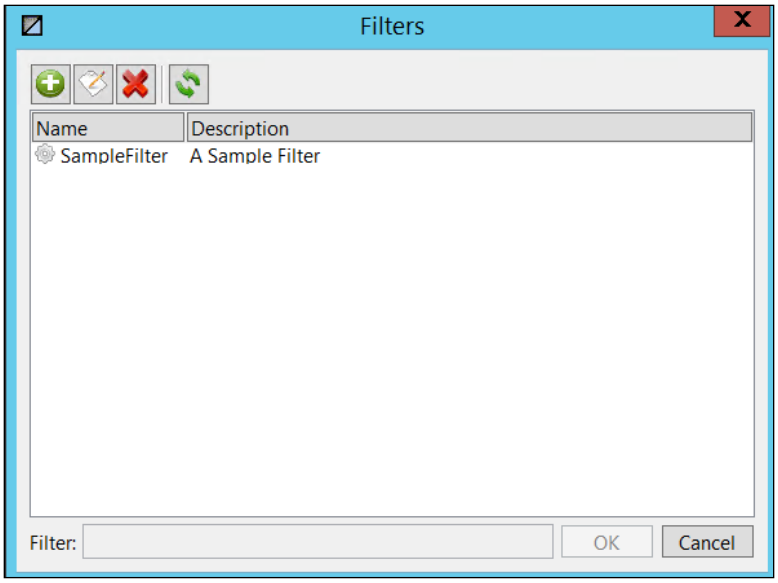


Figure 4-139. Filters Dialog Box

4.16.3 Logging Policies

Before streaming events, the steps in 4.16.1 must be completed. Then, the log4j2.xml file must be configured to forward AutoPilot policy and other events.

```
<!-- uncomment the following to stream policies events to Nastel XRay -->
<!--Logger name="policies" level="info" additivity="false">
    <AppenderRef ref="Tnt4j"/>
</Logger-->

<!-- eventstream events are streamed to Nastel XRay -->
<Logger name="eventstream" level="info" additivity="false">
    <AppenderRef ref="Tnt4j"/>
</Logger>
```

To stream all policy events, uncomment the 3 lines shown by removing "!—" from the beginning and "—" from the end as shown below

```
<!-- uncomment the following to stream policies events to Nastel XRay -->
<Logger name="policies" level="info" additivity="false">
    <AppenderRef ref="Tnt4j"/>
</Logger>

<!-- eventstream events are streamed to Nastel XRay -->
<Logger name="eventstream" level="info" additivity="false">
    <AppenderRef ref="Tnt4j"/>
</Logger>
```

You can repeat this process for any service logs you want to stream, as shown in the example for eventstream.

The **Logging** tab can be configured to specify the log file. The **Logging** tab can be found on the **Property** dialog box for the policy manager or for specific policies by accessing the **Properties** tab for it. The latter is the option which will stream by default (policies).

1. Right-click the policy manager or specific policy and select **Properties** from the drop-down menu.

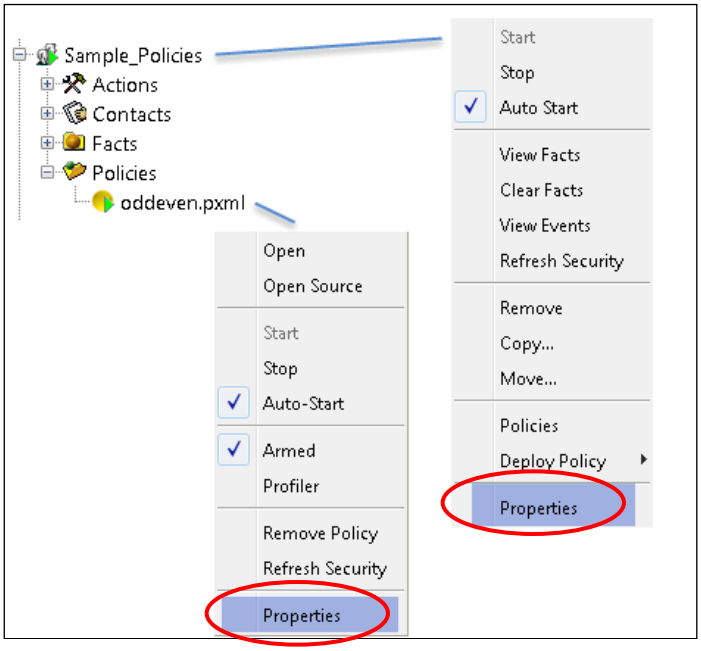


Figure 4-140. Policy Manager and Policy Menus

2. The **Logging** tab is identical regardless of which method is used. The values specified here for **Log name** must match the value specified in **log4j2.xml** in order to stream.

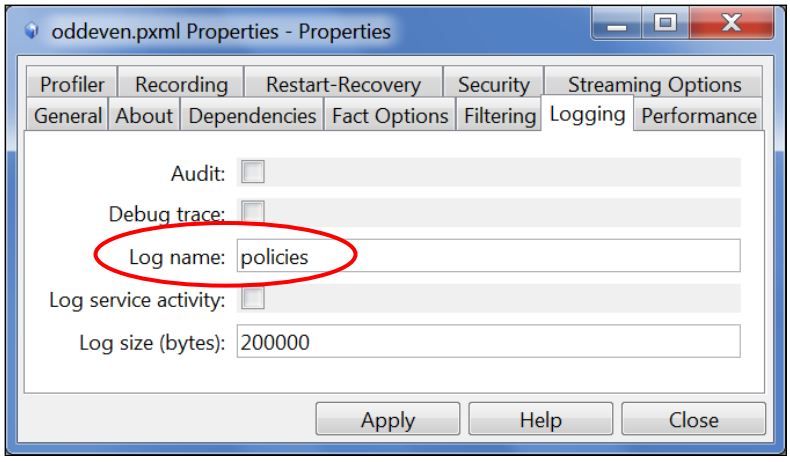


Figure 4-141. Logging Tab

AutoPilot Events are represented in Nastel XRay as Activities, Events, Snapshots, and properties.

The activity represents the "event" generated by AutoPilot and ties the event activity together. The events represent the different types of actions that AutoPilot can take as a result of an event, including sending email, running methods, and executing user actions. The snapshot contains the parameters associated with the state of the AutoPilot event.

Chapter 5: Customizing AutoPilot M6

This chapter describes customization options of various M6 components. Customization should be performed by M6 administrator, which must have admin privileges to the file system where M6 is installed.

This section applies to M6 components such as CEP server, domain server, and M6 User Console.

5.1 Overview

M6 customization can be divided into the following categories:

- **Operational Properties** control operation run-time behavior of each component. Operational properties are maintained in node.properties and .LAX files.
- **Performance Properties:** Alters performance behavior of M6 components. Each component (CEP server, domain server and M6 User Console) is customized via **node.properties** files and corresponding ATP*.LAX file. There is a separate **node.properties** file for details. (Refer to table below.)
- **Registry Configuration:** all deployed service definitions are maintained in registry files. Registry files are .XML files, which are dedicated for each domain server, CEP server and web server. Registry files can be moved from one CEP server to another for migration purposes. They can also be merged by placing a registry file into a corresponding **import** directory. Once merged by a CEP server, the registry file is renamed into an **import\<file>.xml.merged.** Registry files with .DAT extension are binary files and do not follow the same rules as the XML files.

You can use the "include" command to include property files within other property files. For example, include../plugin.properties.

Table 5-1. Server Property Files (apwmq.properties)

Components	M6 Properties	JVM Properties Files
Domain Server	naming\node.properties	ATPNAMES.LAX
CEP Server	localhost\node.properties	ATPNODE.LAX
M6 User Console	mconsole\node.properties	ATPCONS.LAX
M6 Web	jakarta-tomcat\webapps\autopilot\node.properties	N/A

Table 5-2. Server Registry Files

Registry	Locations	Description
registry.xml	**Domain Server**: \naming **CEP Server**:\localhost **M6 Console**:\mconsole **M6 Web**:\jakarta-tomcat\webapps\ autopilot	Contains all deployed service definitions for the corresponding server or console. It includes experts, managers, and policies.
naming.xml	Domain Server: \naming	Contains domain directory registration entries. All registered services in the domain maintained in this file.

security.dat	Domain Server: \naming	Non-XML binary file that contains domain account definitions. File must be secure. Loss or corruption of file will result in loss of all account information.
actions.xml	Domain Server: \naming	Definitions of all globally defined user actions.
fileacl.xml	Domain Server: \naming	Access control list for all user defined policies (.bsv, .pxml and .bsp files).
filelock.xml	Domain Server: \naming	Policy lock file maintains information about users who hold policy locks.
apwmq.properties	Domain Server: [autopilot_home]	Used to define abbreviations that can be used in fact names of a business view. Example: property GROUP = NASTELPRDM6 property WSMON = WS_Monitor property NODMON = Node_Monitor property QMMON = QM_Monitor (See figure 4-141-b below)

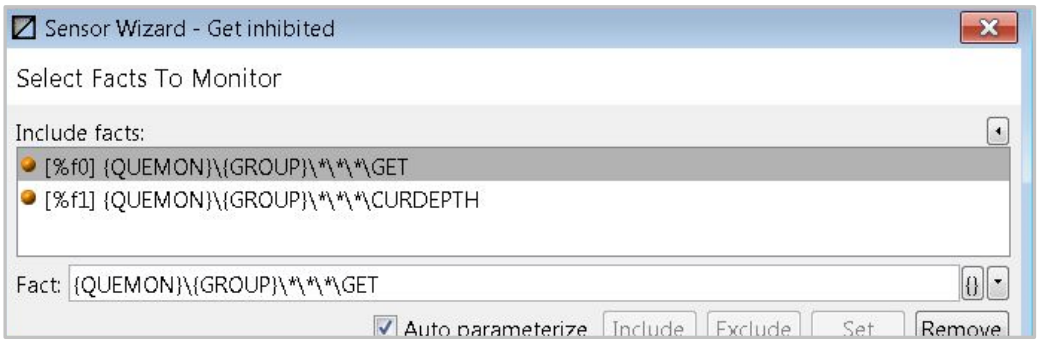


Figure 4-141-b. Sensor Wizard – Use of Aliases

Registry XML files can be edited by hand with any text or XML editor. However, this should be done with extreme caution. This could be useful when making global changes that otherwise might require too many console procedures.

> **NOTE** Do not edit or remove `security.dat` registry file. Make sure all registry files are backed up periodically and can be restored in case of a problem, loss, or corruption.

5.2 Java Runtime—LAX File Customization

LAX file is associated with every M6 executable (except for AutoPilot M6 Web Server). Process name has a corresponding file with `.LAX` extension (example: `ATPNODE.lax`). `.LAX` files define Java run-time environment such as JRE, path, CLASSPATH, memory limitations and other Java start-up parameters. Most parameters do not require change, with the following exceptions:

- **lax.nl.java.option.java.heap.size.initial:** Initial heap size (bytes) of JVM where M6 component is executing. Default setting (32MB) is system dependent. If CEP server(s) are monitoring large environments (10000 or more facts), default number can be increased.

> **NOTE** JVM maximum heap size is determined by JDK1.4 or higher.

- **lax.nl.java.option.java.heap.size.max:** Maximum heap size (bytes) of JVM where M6 component is executing. Default heap size is 32MB (system dependent). If CEP server(s) are monitoring large environments (10000 or more facts), default number can be increased. In general, large configurations may require up to 512MB of maximum heap size.
- If properties above are not defined, include by adding lines to .LAX files as follows:

```
// initial heap in bytes
lax.nl.java.option.java.heap.size.initial=64000000
// maximum heap in bytes
lax.nl.java.option.java.heap.size.max=256000000
```


NOTE

The lax.nl.current.vm file points to the location of the Java file. However, if the location is wrong or missing AND you are running SU23 AND Java exists on your machine, then it is ignored.

5.2.1 Using Server JVM

For optimal performance, it is highly recommended to replace and then enable the property **lax.nl.java.option.additional=-server** with the following on Windows, Solaris and Linux systems running with Sun JR1.4 or higher. Replace this property in ATPNODE.lax and ATPNAMES.lax

lax.nl.java.option.additional=-server -Xnoclassgc - XX:SurvivorRatio=16

-XX:PermSize=128m -Xmn256m

This enables the use of server JVM which allows M6 to take advantage of multiple CPUs and enhanced memory and CPU performance.

To enable server JVM on JR1.3 platform for Solaris and Linux (Sun JVM only):

- Stop all M6 services and applications
- Rename [*AUTOPILOT_HOME*]/jre/bin/client to [*AUTOPILOT_HOME*]/jre/jvm_client
- Rename [*AUTOPILOT_HOME*]/jre/bin/server to [*AUTOPILOT_HOME*]/jre/client
- Restart all services.

To enable server JVM on JR1.3 platform for Windows (Sun JVM only):

- Stop all M6 applications and services.
- Install JDK1.3.1_14 or later.
- Rename [AUTOPILOT_HOME]/jre to [AUTOPILOT_HOME]/old_jre
- Copy [jdk1.3.1]/jre to [AUTOPILOT_HOME]
- Rename [*AUTOPILOT_HOME*]/jre/bin/client to [*AUTOPILOT_HOME*]/jre/jvm_client
- Rename [*AUTOPILOT_HOME*]/jre/bin/server to [*AUTOPILOT_HOME*]/jre/client
- Restart all services.

To enable server JVM on JR1.3 platform for other operating systems, consult specific JVM documentation for details.

Troubleshooting Server JVM

When CEP servers are not responding or key CEP server backlog parameters are growing, generate JVM stack traces to troubleshoot JVM problems by doing the following:

1. Start CEP server in console mode:

- For CEP server, [*AUTOPILOT_HOME*]/localhost/ATPNODE –console
- For domain server, [*AUTOPILOT_HOME*]/naming/ATPNAMES –console

2. Generate a stack trace while the problem is occurring.
   - For Windows: Press **Ctrl-Break** within the running window.
   - For UNIX: kill –QUIT [*java_pid*]

Send stack traces to Nastel support support@nastel.com for analysis.

5.3 Server Runtime – PROPERTY FILES

M6 maintains five property files as follows:

- [*AUTOPILOT_HOME*]/global.properties – properties shared by all run-time components such as domain server, CEP servers, console, and web server.
- [*AUTOPILOT_HOME*]/domain.properties – global properties shared by all CEP servers within a domain. File is loaded by domain server only and cannot be refreshed while domain server is running. Domain server must restart to apply changes. It is recommended to use "shared." prefix for all properties to distinguish global properties from local ones. For example: shared.prop1=test shared.prop2=test2
- [*AUTOPILOT_HOME*]/naming/node.properties – properties shared by domain server and all deployed services within the domain server.
- [*AUTOPILOT_HOME*]/localhost/node.properties – properties shared by CEP server and all deployed services within the CEP server.
- [*AUTOPILOT_HOME*]/mconsole/node.properties – properties shared by the Console.

These supported properties can be defined and overwritten by administrators by copying into [*AUTOPILOT_HOME*]/global.properties or individual node.properties to take effect.

Each node.properties file is identical in structure but may define different properties for various components. This file defines runtime configuration that defines server operational parameters.

Properties are defined as: *property property_name= value* format. User-defined properties can be defined as well.

For example:

```
property server.facts.capacity = 100000
property server.net.sessions.poolsize = 3
property server.net.agents.poolsize = 1000
property server.log.size = 2000000
```

Table 5-3. General Server Properties

Property	Description	Values
autopilot.home	M6 install directory set automatically during installation. **DO NOT** change or define this variable.	String. [install_dir]
server.type	Type of CEP server. **DO NOT** change this property.	String. [*type*] Server , Domain, Client
installation.update.groups	List of groups M6 installation is subscribing to for updates.	String. Console, Mandatory, Service_Updates

Table 5-4. Console Properties

Property	Description	Values
console.external.browser	Launch external web browser within business views.	Boolean. Default is true.
console.pwd.notice	Early expiration of password notification.	Boolean. Default is true.
console.pwd.notice.days	Number of days until password expiration.	Integer. Default is 15.
gui.lookandfeel	Look and Feel of the GUI. Default – default Java look and feel Native – native OS look and feel SkinLF – skinnable look and feel based on supplied theme packs.	Boolean. Default is Native.

To enable one of the theme packs listed below, set gui.lookandfeel to SkinLF and uncomment one line.
gui.skinlf.themepack=../themes/aquathemepack.zip
gui.skinlf.themepack=../themes/macosthemepack.zip
gui.skinlf.themepack=../themes/beosthemepack.zip
gui.skinlf.themepack=../themes/bbjthemepack.zip
gui.skinlf.themepack=../themes/whistlerthemepack.zip
gui.skinlf.themepack=../themes/modernthemepack.zip
gui.skinlf.themepack=../themes/themepack.zip
gui.skinlf.themepack=../themes/xplunathemepack.zip
gui.skinlf.themepack=../themes/toxicthemepack.zip

Table 5-5. Policy Performance Properties

Property	Description	Values
server.policy.buffer.size	Facts are buffered and flushed when limit is reached.	Integer. Default is 500.
server.policy.buffer.grace period	Facts are buffered before flushing after defined grace period regardless of buffer size.	Integer. Default is 1000.
server.sensor.delivery.flowpct	Disables/enables flow control.	Integer. Default is 0.
server.sensor.delivery.batch	Processes sensor events.	Integer. Default is 2000.
server.sensor.db.queue.limit	Maximum queue size of all entries.	Integer. Default is 1000.
server.sensor.db.queue.flowpc	The percentage at which logging is re-enabled after reaching 100% utilization. In order to make sure that capacity is not immediately exceeded again when it is turned back on, logging is only re-enabled when the utilization drops down to this threshold. Default: 60%.	Integer. Default is 60.
server.sensor.profiler	Enable sensor profiler globally for computation of sensor profiling metrics.	Boolean. Default is false.
servlet.cache.timeout	Cache timeout in seconds.	Integer. Default is 5 secs.
server.topic.delivery.limit=server.facts.capacity	Maximum undelivered published queue level. Messages are dropped if level is exceeded.	String: server.facts.capacity/10
server.pubsub.filter.asis	If set to true, increases filter specificity which eliminates circular filter references among CEP servers.	Boolean. Default is false.
server.pubsub.remote.lookups	Improves policy performance for policies subscribing to remote facts. If set to false, disables remote fact lookups when topic has not been initialized yet.	Boolean. Default is false.

Table 5-6. Rule Engine Performance Properties

Property	Description	Values
server.sensor.table.size	Maximum number of expected child sensors for each parent sensor.	Integer. Default is 1011.
server.sensor.batch	Maximum sensor in the pipeline before forcing rule execution.	Integer. Default is 20.
server.sensor.idle	Maximum idle time in ms. before forcing rule execution.	Integer. Default is 3000.
server.sensor.rule.thread.debug	Enable rule engine debug mode.	Boolean. Default is false.
server.sensor.process.nulls	Enables/disables node.properties to use null processing within rules for processing fact disconnection.	Boolean. Default is true.

server.sensor.comp.mode	Disables/enables null processing within rules.	Boolean. Default is false.

Table 5-7. Communication Properties

Property	Description	Values
server.agent.timeout	Processing time in ms. allowed for requests executed remotely.	Integer. Default is 35000ms.
server.socket.pipe.trace	Enable trace for all socket communication. (Enabled only for debugging and tracing.)	Boolean. Default is false.
server.classloader.debug	Enable all network class loading trace.	Boolean. Default is false.
server.socket.option.TCP_NODELAY	Enables/disables TCP algorithm; true turns it off, false turns it on	Boolean. Default is true.
server.socket.option.SO_KEEPALIVE	Enable TCP keep alive for server running behind a firewall.	Boolean. Default is false.
server.socket.option.SO_RCVBUF	Override TCP receive buffer size in bytes	Integer. Default is 65536 bytes.
server.socket.option.SO_SNDBUFF	Override TCP send buffer size in bytes.	Integer. Default is 65536 bytes.
server.net.heartbeat	Enable heartbeat exchange. Heartbeat exchanges prevent firewalls from dropping idle connections	Boolean. Default is true.
server.net.heartbeat.interval	Heartbeat interval in ms.	Integer. Default is 60000.
server.net.heartbeat.trace	Enable heartbeat trace	Boolean. Default is true.
server.io.input.buffer.size	Control input buffer sizes.	Integer. Default is 8192 bytes.
server.io.output.buffer.size	Control output buffer sizes.	Integer. Default is 8192 bytes.
server.net.sessions.poolsize	Maximum number of parallel sessions between each CEP server. More sessions will improve performance but will require more memory.	Integer. Default is 3. 2 are minimum.
server.net.agents.poolsize	Number of parallel requests that can be issued simultaneously.	Integer. Default is 1000.
com.nastel.nfc.net.trace	Enable all low-level communication trace.	Boolean. Default is true.
server.net.connection	Override with the specified host name, IP for communication. Should always have the value of the public interface.	String. hostname,ipaddress
server.pipe.processor.limit	Maximum number of queued objects for each socket connection. Connection is disabled and objects will be dropped when the limit is reached.	Integer. Default is 20000.
server.pipe.delivery.flowpct	Enables/disables flow control.	Integer. Default is 60.
server.process.timeout	Business Views will unblock and continue after the specified timeout value.	Integer. Default is 1000.

Table 5-8. Service Properties

Property	Description	Values
server.services.facts.expire	Default fact expiration for all services. Should only be used to enable server-wide expiration for all deployed services.	Integer. Default is 0.

Table 5-9. Security Properties

Property	Description	Values
server.security.algorithm	Security algorithm. **DO NOT** change after user accounts are created.	DSA
server.security.provider	Security profile provider. **DO NOT** change after user accounts are created.	Sun
server.aip.port	Enables AIP communication support. The property can be added to any CEP server's node.properties file.	Integer. Default is port_number.
server.aip.trace	Enables AIP trace.	Boolean. Default is True.
server.security.storage.provider	Security profile storage provider. **DO NOT** change.	String. com.nastel.nfc.security.SecurityBoard
server.unmask	Default user permission mask applied to every new user account.	String. Base/Control Base/Read Base/Change Base/Execute, Base/Read
server.security.token.expiry	Security session token expiration in ms.	Integer. Default is 10000ms
server.security.password.length	Minimum password length.	Integer . Default is 5.
server.jaas.realms	Space-oriented list of domain names that CEP servers and console clients may authenticate with.	String. [Domain 1] [Domain 2]
java.security.krb5.conf	Defines properties for default realm.	String. {autopilot.home}krb5.conf

Table 5-10. Domain Server Properties

Property	Description	Values
server.domain	Logical name for the domain serviced by the domain server.	Boolean. Default is DOMAIN.
server.naming.url.port	TCP port for the Domain Server directory service.	Integer. Default is 2325.
domain.server.url	URL of the Domain Server where port should equal value of **server.naming.url**.port	String. Default is: bbns://localhost:port
domain.server.name	Name of domain server as registered in the directory service.	String. Default is DOMAIN_SERVER.
domain.server.failover.url	Failover domain server location. Edit and uncomment the following line to enable domain server failover where port should equal the value of the alternate **server.naming.url**.port.	String. Default is: bbns://failover-server:port
installation.update.folder	URL location of all M6 software for distribution to all M6 installation. ds://equals the root directory in business.view.dir or file.system.root properties.	String: Default is ds://software

Table 5-11. Server Properties		
Property	**Description**	**Values**
server.work.dir	Working directory of running server. Read Only.	String. [*aphome*]/naming
server.user.url.port	TCP port for server-to-server communication.	Integer. Default is 3005.
server.log.dir	Logs directory where all logs are being written.	String. [*aphome*]/logs
server.import.dir	Server registry import directory.	String. [*workdir*]/import
server.snmp.mibs	Location of all SNMP mibs.	String. [*aphome*]/mibs
server.template.dir	Location of service templates.	String. [*aphome*]/templates
server.debug	Server wide debug mode for extra information.	Boolean. Default is false.
server.sampling.rate	Sampling rate for server wide statistics for [*SERVER]*_Facts expert.	Integer. Default is 30000.
server.auto.gc.memory.usage	Force garbage collection when memory utilization hits specified number.	Integer. Default is 0.82.
server.auto.gc.timeout	Minimum time before auto gc cycles.	Integer. Default 600000.
server.facts.capacity	Maximum number of facts/properties that can be handled by CEP server. CEP server may handle more facts than specified by the property, but performance may degrade significantly. CEP server will not shutdown if limit is exceeded.	Integer. Default is 10000.
server.topic.cache.size	Maximum size of the cache (number of objects). This cache is used to cache remote objects. Also see server.topic.cache	Integer. Default is 1101.
server.topic.request.limit	Maximum number of outstanding requests for the topic.	Integer. Default is 10000.
server.publish.force	Force publishing a bypass duplicate check. Reduces network bandwidth utilization and CPU and memory consumption.	Boolean. Default is false.
server.facts.history.maxsize	Server-wide history size for all collected facts. Maintains history for number of samples selected for all facts.	Integer. Default is 0.
server.facts.history.maxtime	Server-wide history time for all collected facts. Maintains history for time-base collection of samples in milliseconds for all facts.	Integer. Default is 0.
server.facts.expire	Server-wide facts expiration in ms.	Integer. Default is 0.
server.facts.log.incr	Enable only incremental logging for facts that are added when logging is enabled.	Boolean. Default is false.
server.facts.expiry.rate	Rate at which expiration of facts is checked, lowering value decreases performance.	Integer. Default is 60000.
server.facts.expiry.debug	Enable/disable fact expiry debug trace.	Boolean. Default is false.
server.registry.backups	Maximum number of backups for registry files.	Integer. Default is 5.
server.auto.save.timeout	Force auto save after registry change within specified ms.	Integer. Default is 3000ms.
server.system.facts.expire	Set expiration for system facts in ms.	Integer. Default is 0.
server.max.stdout.size	Maximum log size in bytes for stdout and stderr output.	Integer. Default is 500000 bytes.
server.service.facts.logging	For Experts and Managers ONLY. CEP server records all facts produced by the service which can be played back using apfact utility.	Boolean. Default is true.
java.awt.headless	For UNIX ONLY. Enable AWT headless mode to prevent AWT/X11 related activity for server applications which can generate exceptions at runtime.	Boolean. Default is true.

Table 5-12. Grid Properties		
Property	**Description**	**Values**
server.grid.folder	Location of grid definitions.	String. [AUTOPILOT_HOME]/ naming.grid
server.grid.vote.sample	Wait interval before selecting primary grid server.	Integer. Default is 120000.
server.grid.vote.delay	Wait before voting on a primary after primary failed.	Integer. Default is 15000.
server.grid.dynamic.services	All services in the grid are dynamic – nonpersistent. Removed from directory when hosting server is not available.	Boolean. Default is false.

Table 5-13. JDBC Properties		
Property	**Description**	**Values**
server.jdbc.oracle.driver	Oracle Driver	oracle.jdbc.driver.OracleDriver
server.jdbc.oracle.url	Oracle URL	jdbc:oracle:thin:@
server.jdbc.sqlserver.driver	SQL Driver	net.sourceforge.jtds.jdbc.Driver
server.jdbc.sqlserver.url	SQL URL	jdbc:jtds:sqlserver://
server.jdbc.sybase.driver	Sybase Driver	net.sourceforge.jtds.jdbc.Driver
server.jdbc.sybase.url	Sybase URL	jdbc:jtds:sybase://
server.jdbc.db2.driver	DB2 UDB Driver	COM.ibm.db2.jdbc.net.DB2Driver
server.jdbc.db2.url	DB2 UDB URL	jdbc:db2://
server.jdbc.hypersonicsql.driver	Hypersonic SQL Driver	org.hsql.jdbcDriver
server.jdbc.hypersonicsql.url	Hypersonic SQL URL	jdbc:HypersonicSQL:hsql://
server.jdbc.jdbc.odbc.driver	Generic JDBC/ODBC Driver	sun.jdbc.odbc.JdbcOdbcDriver
server.jdbc.jdbc.odbc.url	Generic JDBC/ODBC URL	jdbc:odbc:
server.jdbc.informix.driver	Informix Driver	com.informix.jdbc.lfxDriver
server.jdbc.informix.url	Informix URL	jdbc:informix-sqli://
server.jdbc.mysql.driver	MySQL Driver	com.mysql.jdbc.Driver
server.jdbc.mysql.url	MySQL URL	jdbc:mysql://
server.jdbc.derby.driver	Derby Driver	com.ibm.db2.jcc.DB2Driver
server.jdbc.derby.url	Derby URL	jdbc:derby:net://

5.4 Key Performance Properties

Several key performance properties may require customization:

- server.facts.capacity
- server.agent.timeout
- server.topic.cache
- server.topic.cache.size

The most important property is `server.facts.capacity`, which determines the maximum number of managed properties that can be monitored. Determine if CEP server or domain server requires property changes by checking [*MANAGED_NODE*]`_Facts` as follows:

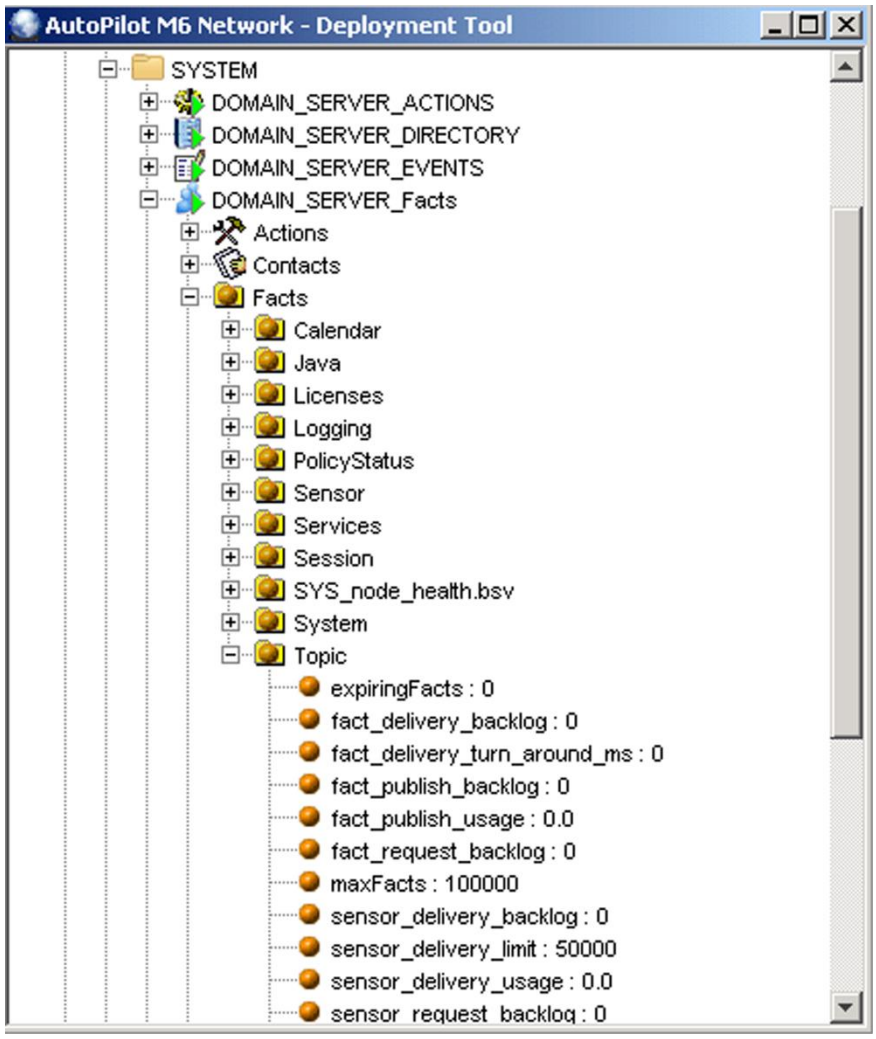


Figure 5-1. CEP Server, Domain Server Facts

If [MANAGED_NODE]Facts\Topic\totalFacts is less then [MANAGED_NODE]_Facts\Topic\maxFacts then:

`server.facts.capacity` will require modification to: *totalFacts + 1/3*totalFacts*.

(Example: Where *maxFacts* = 700 and *totalFacts* = 600, increase *totalFacts* by *totalFacts* plus 1/3 or *totalFacts* = 800).

5.5 Configuring Tomcat and Microsoft IIS

This Document provides instructions for setting up Internet Information Services (IIS) to serve Java Server Pages (JSP) in conjunction with the M6 Web Server (Jakarta Tomcat 3.2).

IIS will function normally, except when a JSP is encountered. IIS will pass the JSP to Tomcat, which will process and send the results back to IIS.

The following file paths will be used in these instructions. If installed in a different directory, then change accordingly. These are the default installation directories for the programs listed, except for Tomcat, which does not specify a default installation directory.

- **AutoPilot M6:** [*AUTOPILOT_HOME*]=c:\nastel\AutoPilotM6
- **Java:** [JAVA_HOME]=[AUTOPILOT_HOME]\jre
- **Catalina:** [*CATALINA_HOME*]=[*AUTOPILOT_HOME*]\jakarta-tomcat
- **isapi_redirect.dll**: `[CATALINA_HOME]\bin`
- **IIS Root:** C:\InetPub\wwwroot

5.5.1 Tomcat Set-up


NOTE

Jakarta Tomcat v3.2.1 is installed during the M6 (AutoPilot M6 Web option) installation under `[AUTOPILOT_HOME]\jakarta-tomcat` directory. Additional information about Tomcat is available at: http://jakarta.apache.org/downloads/binindex.html.

5.5.1.1 Verify Settings in Windows

1. From **My Computer** (on desktop) right-click to open sub-menu. Click **Properties** to display *System Properties* screen, click **Advanced**.
2. Click **Environmental Variables** tab. The *Environmental Variables* screen is displayed.
3. Under *System Variable* verify the following settings:
   - CATALINA_HOME=[*AUTOPILOT_HOME*]\jakarta-tomcat
     (Example: [*CATALINA_HOME*]=C:\nastel\AutoPilotM6\jakarta-tomcat)
   - JAVA_HOME=[*AUTOPILOT_HOME*]\jre

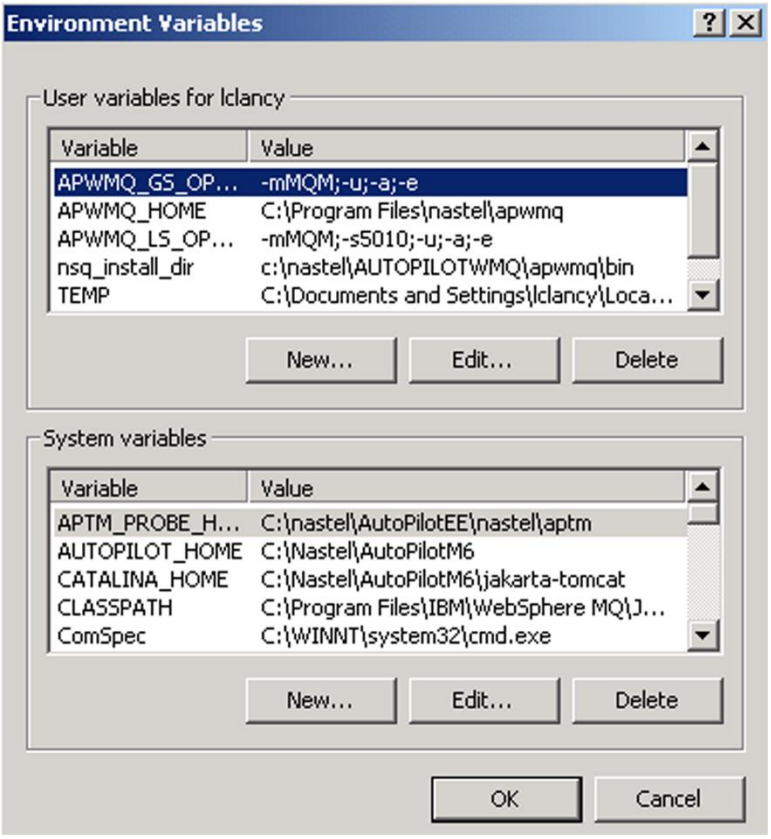


Figure 5-2. Environmental Variables

4. Restart the computer if PATH settings required change.

> **NOTE** In the *Environmental Variables*, ensure `[JAVA_HOME]` is the first directory in your PATH. If other Java programs have been installed, they may initiate their Java Run-Time directory first. This will corrupt the Jakarta-Tomcat installation ability to launch. If another Java application is installed later, ensure `[JAVA_HOME]` remains the first directory in the PATH.

5.5.1.2 Starting Jakarta-Tomcat

This section provides procedures for starting Jakarta-Tomcat as a stand-alone application.

1. Open the Command Prompt.
2. CD to `[AUTOPILOT_HOME]\jakarta-tomcat\bin`
3. Run **startup**. A second *Command Prompt* screen identified as *Tomcat 3.2*. The *Tomcat 3.2* screen will contain the Tomcat status. If this screen closes, Tomcat did not launch correctly. Recheck settings.
4. To verify Tomcat is running properly open a browser to: http://localhost:8080/. The default Jakarta-Tomcat home page will be displayed. Tomcat is properly configured and running normally.
5. Close the browser as desired.
6. At the *Command Prompt*, run **shutdown** to close Tomcat. Close *Command Prompt* screen.

5.5.2 Configuring Tomcat with IIS

This section contains configuration procedures for the following:

- Configuring Tomcat to work with Java Server Pages (JSP) by identifying where the JSP files are located using `server.xml`.
- Configuring IIS to manage all files, other than JSP. Specify the pages that are to be redirected to Tomcat using `uriworkermap.properties`.

1. Open `[CATALINA_HOME]\conf\server.xml` in Notepad.
2. Toward the end of the file, edit Context Path (if required)

   from:

   ```
   <Context path="/admin" docBase="webapps/admin" crossContext="true"
   debug="0" reloadable="true" trusted="true" />
   ```

   to:

   ```
   <Context path="/" docBase="C:/InetPub/wwwroot" debug="0"
   reloadable="true"/>
   ```

3. Open `[CATALINA_HOME]\conf\uriworkermap.properties` with Notepad.
4. Add the following in the file: `/*.jsp=ajp12`
5. All the other lines in `uriworkermap.properties` may remain as is or be removed as desired.
6. Ensure proper operation by opening a browser to: http://localhost and checking that the normal default screen is displayed.
7. Open `C:\InetPub\wwwroot`, insert a test file containing sample code (`test.jsp`) and direct browser to: http://localhost/test.jsp. The file will execute if configuration is correct.

NOTE

If a blank page is loaded, or a prompt to download is displayed, there is a malfunction.

5.5.2.1 Install ISAPI Redirector and Jakarta-Tomcat Virtual Directory

This paragraph covers installation of an ISAPI filter in IIS. When IIS receives a request for pages that meet criteria compliant, it will hand the request off to another program (Tomcat) to handle the processing. Tomcat then returns plain text to IIS. In turn, IIS sends the text to the browser.

This paragraph also covers creating a virtual directory that has the same process.

1. `isapi_redirect.dll` is located in `[CATALINA_HOME]\bin`
2 Double-click `[CATALINA_HOME]\bin\isapi_redirect_nt.reg` or `[CATALINA_HOME]\bin\isapi_redirect_2000.reg` to import the information into the registry. (If other directories have been used, other than specified here, then the `.reg` file will require appropriate editing).


NOTE

Restart the PC after the registry update for changes to take effect.

3. Open *IIS Management Console,* create a new virtual directory named **jakarta,** and make the physical path `[CATALINA_HOME]\bin`. Ensure the virtual directory has **Execute** permissions.
4. In the *IIS Management Console* right-click your machine name, (not the root web). Select **properties**.
5. Click **Edit** next to *Master Properties* for the Web Service.
6 Select ISAPI Filters, click **Add**. Name the filter *jakarta*. The file is located in: `[CATALINA_HOME]\bin\isapi_redirect.dll`.
7. Create the M6 WEB SERVER virtual directory named **autopilot**.
8. Click **Properties**.
9. Click **"A redirection to URL"**, type: http://localhost:8080/m6console
10. Restart the IIS Web Server.
11. Open *Control Panel* (Windows), select *Services*, restart the *IIS Admin* service (ensure Web Publishing Service restarts as well).
12. Return to the *ISAPI Filters* screen; ensure that *jakarta* filter has a green arrow next to it. The green arrow indicates proper operation.
13. Update `workers.properties` in `[CATALINA_HOME]\conf` with
    `worker.ajp12second.host=YourRemoteHostIP`
    `worker.ajp12second.port=8007`
14. Ensure Tomcat is running. Open browser to http://localhost:8080/examples/, a Tomcat index page will be displayed. There are several Java Server Page (JSP) examples which can be used for testing.

Chapter 6: Troubleshooting Techniques

This chapter describes general troubleshooting techniques for any of the M6 components. Since M6 is a distributed network of servers and services, it could make problem determination somewhat difficult. This section describes error reporting and problem determination facilities.

6.1 Overview

M6 Troubleshooting Facility is categorized as follows:

- **Event Logs:** Event/error logs that are local to each M6 installation.
- **Service Activity:** Allows users to enable traces for any management service within the M6 domain for the purpose of problem determination. Traces can be viewed using Event Viewer.
- **Event Viewer:** Centralized Event Viewer allows the user to view, filter, sort events, and errors on any M6 installation, local or remote. Event Viewer can be launched using M6 User Console.

6.2 Event Logs

Each M6 installation maintains a set of logs for all installed components in directory `[AUTOPILOT_HOME]\logs`. Each log has an `.EVT` file extension. Logs have *Logical* and *Physical Names*:

- **Logical Name:** User known name that is displayed in the Event Viewer title bar and in individual service Logging Properties. Default Logical Names are *SYSTEM*, *UNNAMED*, *POLICIES,* and *SERVICES*. There are system logs and user defined logs.
- **Physical Name:** physically all logs are located in `[AUTOPILOT_HOME]\logs` directory and have `.EVT` file extension.

Each log file has a physical file name; actual file stored on the file system. Physical name convention is: `[MANAGED_NODE_NAME]_[LOGICAL_NAME]$0|1.EVT` (example: `DOMEGAX_SYSTEM$0.EVT`).

Since logs are circular, M6 may allocate two logs, for example: `DOMEGAX_SYSTEM$0.EVT`, when log is full, the `$0` log is moved to `DOMEGAX_SYSTEM$1.EVT` and the original file is truncated (deleted and recreated).

6.2.1 System Logs

- **SYSTEM:** M6 system services record to this log.
- **SECURITY_ACTIVITY:** (Domain Server only) Security Service records information on security action. Log is created only when "Service Activity" is set to **true** (on). **DOMAIN_SERVER_SECURITY** service.
- **ACTION_ACTIVITY:** Contains failed/audited user actions.
- **UNNAMED:** Should always be empty. This log is maintained when an event is written to a non-existent log. Any events recorded into this log indicate an internal error and should be reported to Nastel technical support.

6.2.2 User Defined Logs

- **POLICIES:** All deployed policies record events into this log (unless changed by the authorized user). Log name can be changed/created when modifying individual policy's *Logging Properties.*
- **SERVICES:** Where all management services record failures, errors, and warnings, as well as trace messages, log name can be changed/created when modifying individual service's *Logging Properties*.

Users may change log names by setting *Log name* of the *Logging* properties section of every management service.

> **NOTE** All log files are stored as plain text files, viewable using any text editor.

Each event in every .EVT event log is formatted as follows:

- **key:**{time=*timestamp*}{type=*5*}{evid=*id*}{account=*user*}{from=*component*}
  {event=*message*}

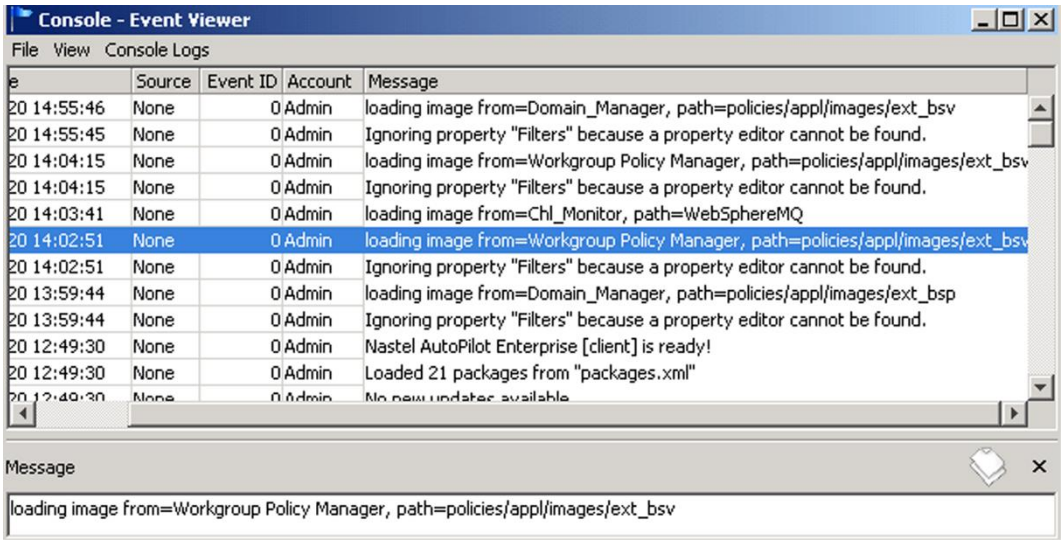


Figure 6-1. Sample Event Log

6.3 Service Activity

> **CAUTION!** Log Service Activity should be enabled only during problem determination. Leaving Log Service Activity enabled will cause significant performance degradation.

Each management service can be enabled to generate Service Activity or detailed trace information. Enable "Service Activity" on a specific management service as follows:

1. Right-click desired management service (Example: expert, manager, or policy) to open sub-menu.
2. Click **Properties**.
3. Click **Logging** tab.
4. **Log Name** is user defined. Logs are system generated if existing log is available.
5. **Log Service Activity**: Click the disable/enable button to enable log service activity. A check mark will be displayed. Select *Audit* to enable audit events recorded every time any user accesses this service.
6. Click **Apply** to apply the changes to the selected service.

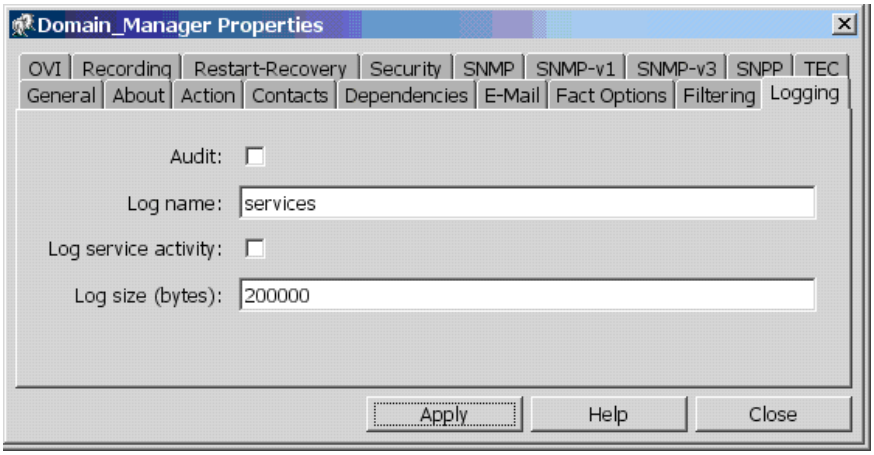


Figure 6-2. Logging, Service Activity

NOTE

When *Log Service Activity* or *Audit* is enabled and applied, logs can be viewed using Event Viewer under the CEP server where service is running.

7. Open the event view by right-clicking the subject manager, expert, or policy, the event viewer will be displayed. Open the event file from the file menu either by clicking **Open,** or from the *Domain Logs* menu as shown in the figure below.

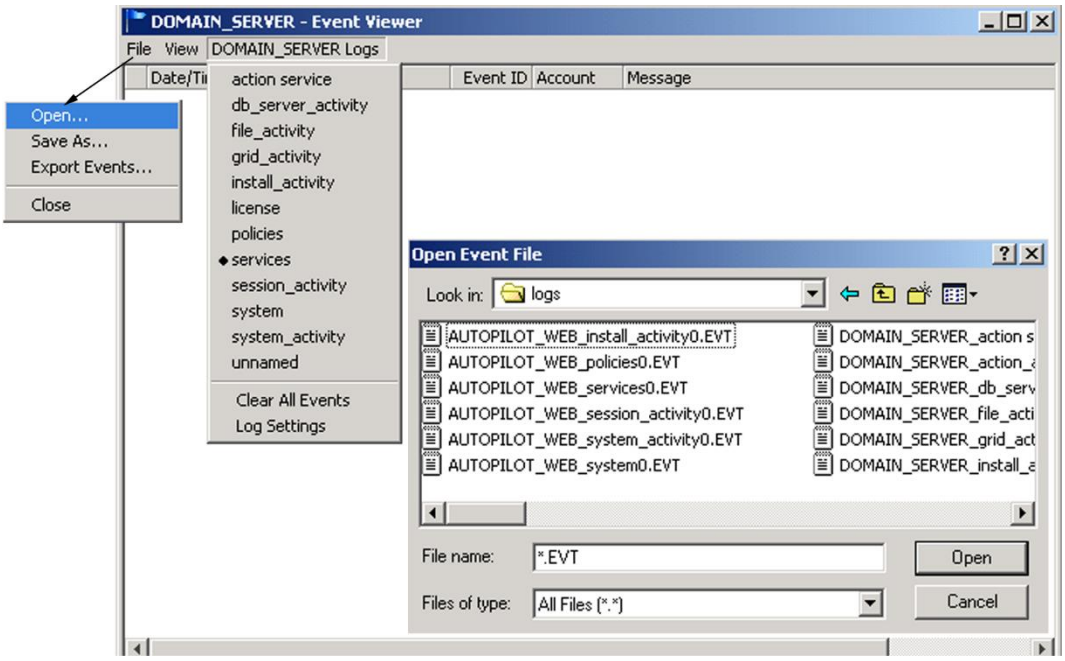


Figure 6-3. Opening Log in Event Viewer

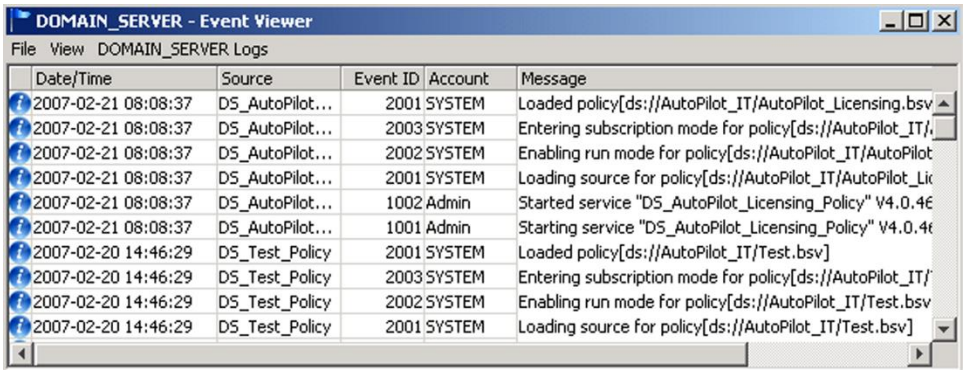


Figure 6-4. Viewing Log in Event Viewer

6.4 Event Viewer

M6 User Console allows users to launch event viewer for each CEP server, manager, expert, or policy. When event viewer is active, users can view all logs generated under specified CEP servers, including remote CEP servers.

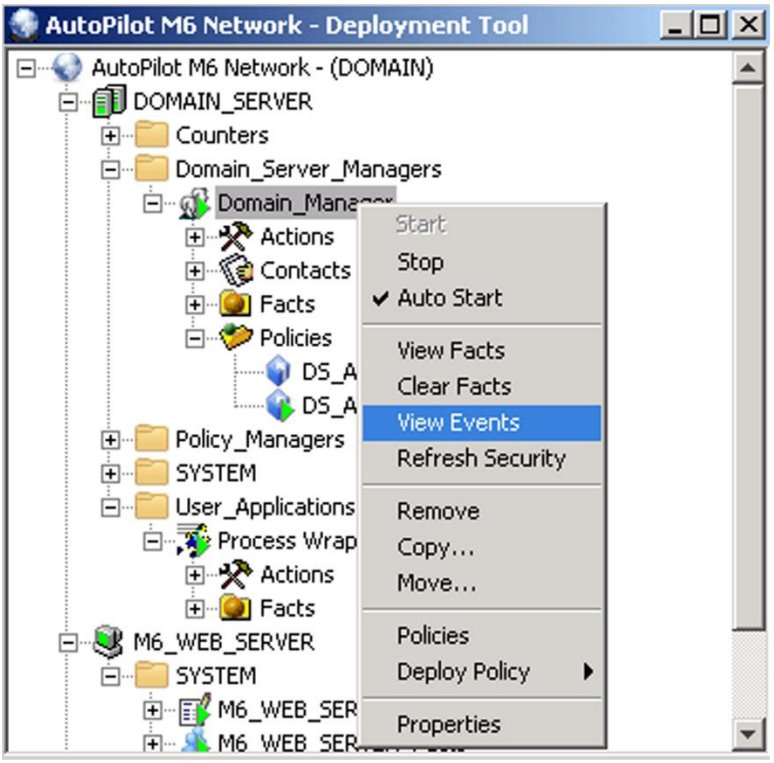


Figure 6-5. Event Viewer for a CEP Server

NOTE Event Viewer is also accessible from M6 User Console (required for troubleshooting M6 User Console activity).

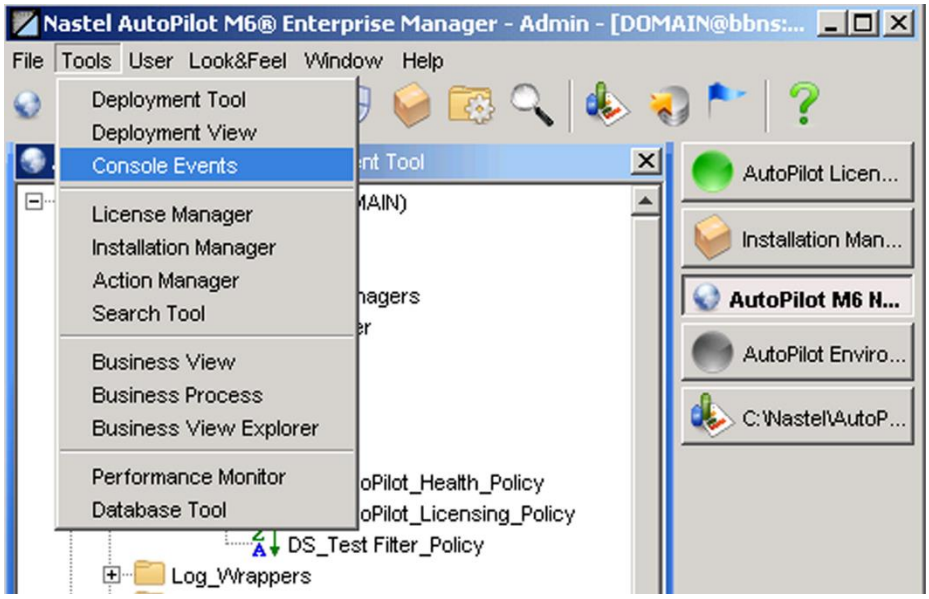


Figure 6-6. Launching Event Viewer from Console Tool Menu

6.5 Generating NRD Files

An NRD file captures the state of the AutoPilot environment. The file is helpful to technical support as an aid in troubleshooting issues. The file is generated automatically during stress but can be generated on-demand using Enterprise Manager.

To generate an NRD file, right-click **dumpRuntime,** then select **Run Action.**

Right-click **dumpRuntime,** then select **Run Action** to generate NRD file.

Figure 6-7. Generate NRD File

Chapter 7: Troubleshooting Procedures

The following chapter identifies specific problems that may occur while configuring/using M6 and gives detailed solutions for these problems. If a problem still exists, after reading this chapter, contact Nastel Technologies as follows:

- To contact Nastel technical support by phone, call 1-800-963-9822 ext. 1, if you are calling from outside the United States, dial 001-631-761-9190.
- To contact Nastel technical support by e-mail, send a message to support@nastel.com
- To access the Nastel automated support system (User ID and Password required), go to http://support.nastel.com/

Contact your local M6 Administrator for further information.

Before requesting support, please verify the following:

- Latest service update is installed.
- All the latest versions of your plug-ins are installed.
- All components are at the same level. For example, CEP Servers, Domain Server, and Domain Console are all at Version: 6.0, Service Level: SU5.
- License has been installed and has not expired.
- CPU license limit has not been reached.
- All services are started (Domain Server, CEP Server, and/or Web Server).

7.1 Problems and Solutions

Problems along with their solutions are grouped into categories as follows:

- Installation
- Business Views
- Facts
- CEP Server
- M6 Web Console Access.

7.1.1 Installation

When running pkgman ../updates/filename command - returns a warning that says: Package database is currently locked. Please try later.

Delete *.lock* file in [*AUTOPILOT_HOME*] directory. This file is created when pkgman is killed before completing.

7.1.2 Business Views

Is there a way to turn on and off monitoring of a specific sensor in a Business View?

M6 has the ability to specify selected sensors within business views to ignore faults and alert conditions during a specified window of time.

1. Right click sensor to be ignored.
2. In the sub-menu, select **Properties**, the *properties* screen will be displayed.
3. Click *Ignore* tab to open ignore properties screen.
4. The default settings have all days selected, with no time specified. This would have no ignore features active. By deselecting a day, default time of "*5 PM*" is displayed, but the ignore feature is disabled.
5. Enter the time, for the day(s) you require.
6. Enter the number of minutes you want the sensor to be ignored (example: 12 Hours = 720 minutes, 24 hours = 1440 minutes).
7. Select (check) desired day(s). Sensor activity for the day(s) and time specified will be ignored.
8. Ensure all other days are disabled to prevent inadvertent disruption of sensor data.
9. Click **Apply** when all day and time settings are set. All information for specified period will be ignored.

Is there a way to view images in the description field of a business view?

The *Description* field is not a fully enabled HTML browser. The recommended approach is to use a link to your pages that contains HTML content including dynamic content, forms, etc. Since business views can also be displayed on the web, it is best to author all HTML pages using a different tool and post them on the web server and then link from a business view description to those pages. The .../images/cs.gif convention would not work since web servers would not be able to locate these images when business views are loaded from the web. Therefore, it is recommended to use off-line web pages and link to them from the *description* field.

How does an administrator of AutoPilot unlock BSV currently locked by someone else?

A BSV can be unlocked by using the Business View Explorer.

1. From AutoPilot Console toolbar, click **Tools > Business View Explorer**.
2. Right-click locked BSV and click **Check-In & Unlock**.

7.1.3 Facts

Is there a way for user to selectively batch remove published facts from the M6 User Console?

1. Right-click on desired expert.
2. In the sub-menu, select **Properties**, the *Properties* screen will be displayed.
3. Click the *Fact Options* tab.
4. In the *Expire facts(ms)* field, enter the desired value in milliseconds for when you want the old fact to be cleared within that expert.
5. Click **Apply**.

Is there a way to publish facts to M6 from your java applications?

1. Import the following package:
   `com.nastel.nfc.util.*`
2. Declare the following:
   ```
   String M6_HOST = "localhost";     // This is host or IP of M6 server.
   int M6_PORT = 6000;               // This is port on which a UDP listener is listening.
   RecordFact sendFact = new RecordFact();
   ```
   RecordFact class is located in `%AUTOPILOT_HOME%\lib\nfc.jar`.
3. Use the reportFact method in your code to send facts like the following:
   ```
   sendFact.reportFact("MyApp\\State=true", M6_HOST, M6_PORT);
   ```

This interface can also be leveraged by non-JAVA applications as follows:

- The interface can be called from the command line and used by C programs using a system function.
- You can also publish a UDP/TCP string to the port in the format accepted by apfact. (See usage display for [*AUTOPILOT_HOME*]`\bin\apfact`). If you start a user application using an M6 Process wrapper expert, the output produced from the application will automatically create facts. The format of the output must match the format accepted by apfact.

7.1.4 CEP Server

CEP Server will not start and generates an error log.

Some possibilities may include:

1. Inconsistency of Service Updates. Ensure the latest service pack is installed for all environments.
2. Incorrect Host Name and/or Port. Modify global.properties file to ensure the correct Host Name and Port are being used.

7.1.5 M6 Web Console

Unable to log onto web console.

Some possibilities may include:

1. Java Runtime Environment (JRE) 1.7 or higher not installed.
2. Internet browser not updated. Internet Explorer 5.5 or higher is recommended.
3. M6 Web Server service is not started.

How do I configure M6 Web Console to only show Business Views?

1. Right-click on desired policy.
2. In the sub-menu, select **Properties**, the *Properties* screen will be displayed.
3. Verify that you are on the *General* tab.
4. Uncheck the *Publish view* checkbox.
5. Click **Apply**.
6. Hide system services by right clicking anywhere on M6 screen and then clicking **Hide System Services.**

Unable to display charting for selected Business View.

Ensure the logging option to database or file is active in the expert.

In User Console:

1. Stop Business View
2. Right-click Business View and in the sub-menu select Properties. The Properties screen will be displayed.
3. Select Logging tab.
4. Select Log sensor status to database and enter database information.
5. Click Apply.
6. Restart Business View.

7.2 FAQs

7.2.1 How to Stop/Start Nastel Services

Domain Server:

Start: [AUTOPILOT_HOME]/naming, type: **nohup ./ ATPNAMES &**

Stop: ps -aef | grep nastel and kill naming process

OR [AUTOPILOT_HOME]/bin, type: **apnet –domain localhost stop DOMAIN_SERVER_SYSTEM**

CEP Server:

Start: [AUTOPILOT_HOME]/localhost**,** type: **nohup ./ATPNODE &**

Stop**:** ps -aef | grep nastel and kill localhost process.

OR [AUTOPILOT_HOME]/bin, type: **apnet –domain localhost stop <NODENAME>_SYSTEM**

Web Server:

Start: [AUTOPILOT_HOME]/apache-tomcat/bin, type: **catalina.sh run &**

Stop: ps -aef | grep nastel and kill tomcat process.

OR [AUTOPILOT_HOME]/jakarta-tomcat/bin, type: **/shutdown.sh**

OR [AUTOPILOT_HOME]/bin, type: **apnet –domain localhost stop AUTOPILOT_WEB_SYSTEM**

For Tworks MQ probe:

Start: nsqtacon –console [options based on TD environment]

Stop: nsqtacmd –stop

7.2.2 What does the icon (gray ball with green refresh arrows) represent?

The icon  means incomplete, meaning sensor does not have enough info to evaluate itself. Usually happens if one or more underlying facts are not available. When this occurs sensor goes into a halt state until all facts can be resolved. This can also occur of the rule itself has a problem causing the sensor to go into a halt state.

Appendix A: References

A.1 Nastel Documentation

Table A-1. Nastel Documentation	
Document Number (or higher)	**Title**
M6/INS 623.001	*Nastel AutoPilot M6 Installation Guide*
M6/WMQ 600.002	*Nastel AutoPilot M6 Plug-in for WebSphere MQ*
M6WMQ-ADM 656.002	*Nastel AutoPilot M6 for WebSphere MQ Administrator's Guide*
M6WMQ-INS 656.001	*Nastel AutoPilot M6 for WebSphere MQ Installation Guide*
M6WMQ/SM 656.001	*Nastel AutoPilot M6 for WebSphere MQ Security Manager User's Guide*

A.2 IBM Documentation

SC33-1872 *WebSphere MQ Intercommunications*

SC33-1369 *WebSphere MQ MQSC Command Reference*

SC34-5456 *WebSphere MQ Using Java*

http://www-306.ibm.com/software/websphere/

A.3 HP OpenView Documentation

http://ovweb.external.hp.com/lpe/doc_serv/

A.4 Java™ 2 J2SE™ for HP-UX Information Library

http://www.hp.com/products1/unix/java/infolibrary/index.html
http://developer.java.sun.com/developer/technicalArticles/Servlets/corba/

A.5 Jakarta Documentation References

http://jakarta.apache.org/site/library.html

A.6 Oracle Online Documentation

http://otn.oracle.com/documentation/content.html

A.7 Tru64 UNIX Online Documentation and References

http://h30097.www3.hp.com/docs/pub_page/doc_list.html

This Page Intentionally Left Blank

Appendix B: Conventions

B.1 Typographical Conventions

Table B-1. Typographical Conventions

Convention	Description
Blue/Underlined	Used to identify links to referenced material or websites. Example: support@nastel.com
Bold Print	Used to identify topical headings, glossary entries, and toggles or buttons used in procedural steps. Example: Click **EXIT**.
Italic Print	Used to place emphasis on titles, menus, screen names, or other categories.
`Monospaced Bold`	Used to identify keystrokes/data entries, file names, directory names, etc.
`Monospaced Italic`	Used to identify variables in an address location. Example: `[`*`AUTOPILOT_HOME`*`]\documents`, where the portion of the address in the brackets [ ] is a variable.
`Monospaced Text`	Used to identify addresses, commands, scripts, etc.
Normal Text	Typically used for general text throughout the document.
Table Text	Table text is generally a smaller size to conserve space. 10-, 9-, and 8-point type is used in tables throughout the AutoPilot M6 product family of documents.

B.2 Naming Conventions

In the redesign of AutoPilot M6, we have defined many elements within the AutoPilot M6 product line.

Table B-2. AutoPilot M6 Related Naming Conventions

Old Name	New Name	Abbreviated As	Link
Nastel AutoPilot	Nastel AutoPilot M6	M6	NA
AutoPilot Web	M6 Web Server	M6 Web Server	http://host:8080
AutoPilot Web Portal	M6 Web Console	M6 Web Console	http://host:8080/m6console
Nastel AutoPilot Business Dashboard	Nastel AutoPilot M6 Business Dashboard	M6 Dashboard	See product documentation.
Managed Node	CEP Server	CServer	NA

This Page Intentionally Left Blank

Appendix C: Command Reference

This Appendix describes M6 command line interface, usage, and options.

C.1 PKGMAN – Product Maintenance

pkgman is a product maintenance utility for M6. It allows users to:

- Display information about all installed components and libraries
- Repair damaged installations
- Verify installed packages
- Install plug-ins, service packs and patches from a local file system or URL location.

All output from **pkgman** is stored in `[AUTOPILOT_HOME]\logs\pkgman.log` in addition to console output, even if executed in GUI mode.

Table C-1. PKGMAN Options

Option	Description
-gui	Runs in GUI mode, opens the Package Manager in GUI mode (default console). (See below.)
-about	Displays product information (see below)
-info	Displays installed product information. Displays installed product component information. (See below.)
-libinfo	Displays library information. (See below.)
-verify	Verifies installed package. Displays the detail of the last installation package. pkgman –verify [*package name*]
-repair	Repairs the installation of the package identified. `pkgman –repair [package name]`
-uninstall	Uninstalls an existing package. `pkgman –uninstall [package name]`
-reinstall	Reinstalls from a package file. `pkgman –reinstall [package name]`
-help	Prints this message.

GUI Mode

The **pkgman** will run as a GUI application when –**gui** command line option is used. It can also be invoked by running **apman** at the command prompt or (Windows only) accessed from the M6 start menu, under the *Product Maintenance* link.

Product Information

Displays the M6 product information when the **–about** option is executed (example: `pkgman –about`).

Installed Product Information

In the sample, the base identified is defined along with the build version and date. Details are displayed by using the **–info** option (`pkgman –info`).

Library Information

The information about installed java packages (jar files) is displayed when using –**libinfo** option (`pkgman –libinfo`).

MIME File Type Recognition

Mime types are searched in the following order:

- The file `.mime.types` in the user's home directory.
- The file [*java.home*]/lib/mime.types.
- The file or resources named [*AUTOPILOT_HOME*]/bin/META-INF/mime.types.
- The file or resource named META-INF/mimetypes.default which is located in `[AUTOPILOT_HOME]/lib/activation.jar` file.

MIME Types File Format

- # comments begin with a "#"
- # the format is [mime type][space separated file extensions]

  for example: text/plain txt text TXT
- # this would map file.txt, file.text, and file.TXT to the mime type "text/plain"

C.2 APNET – Control Utility

apnet allows users to control M6 services from a command line, locally or remotely. **apnet** requires that the domain server be up and running and available on the network. Users don't have to specify the location of the domain server, user, and password if **apnet** is executed on the same server as the domain server.

Table C-2. APNET Options

Option	Description
-useglb	Use settings supplied in `[AUTOPILOT_HOME]/global.properties` (default is false)
-dsname	[*domain server name*] M6 domain server host name (default is DOMAIN_SERVER)
-domain	[*domain server host name*] M6 domain server host name (default is localhost)
-port	[*domain server port*] M6 domain server port (default is 2325)
-user	[*user name*] M6 user name
-password	[*password*] M6 user password
-relation	[*relation_name*] M6 relation name (default is DEPENDS_ON)
-pair	[*ordered_pair*] M6 ordered pair (default is: (*,*) -- all)
Service	Service or policy name. Policy is specified as **policy_name:manager_name**
autostart	Sets the auto-start for the designated service. The service will run whenever the CEP server is active.
start	Manually starts the designated service.
stop	Manually stops the designated service.
disable	Disables the designated service. Service required manual start once disabled.
remove	Removes the designated service.
save	Saves current configurations on the specified CEP server. CEP server is specified as follows: [*NODE_NAME*] _SYSTEM.
query	Queries and displays service registration attributes.
clear	Clears all facts for a specified service. Executed asynchronously, inline and serialized with published facts to preserve order.
get	Queries facts and value pairs for a given pattern.
lookup	Looks up any registered object in the domain server (service and non-service).
change_pwd	Resets account password. apnet must run on the same box as the domain server or user must use Admin User ID/password to reset account password.
reset-pwd Admin	Resets account password for the Administrator.
ignore	Deactivates object monitoring for a limited time (by toggling the fact state to ignore). Useful when users do not want notifications for specific objects. Toggle the fact state as follows: apnet.exe ignore <ServiceName> fact To reenable object monitoring, run the command again for the particular object. Please note, ServiceName should be in double quotes if there are spaces: apnet.exe ignore "Test Counter" counter

Examples:

```
apnet query DOMAIN_SERVER_SYSTEM
apnet stop Domain_Server_Manager
apnet clear Domain_Server_Manager
apnet –domain localhost –port 2325 save DOMAIN_SERVER_SYSTEM
apnet lookup DOMAIN_SERVER
apnet ignore Que_Monitor MQM\KITE\QM\BBB.LQ\CURDEPTH
apnet ignore Que_Monitor MQM\KITE\QM\BBB.LQ\*
```

`apnet` `also executes methods used within M6 services such as experts, managers, and policies.`

Example (when executing JMX MBean methods):

apnet invoke JMXAgent.invoke(Domain,MBeanName,mbean_method.String[]=null)

where:

JMXAgent is the name of the M6 service.

invoke is the name of the JMXAgent method.

Service follows the following format when used with the `invoke` command:

`ServiceName.method(type=value,…)`

where:

type can be a *String, Integer, Float, Double, Boolean, String[], Integer[], Float[], Double[]*, or *Boolean[]*

value is the value of the variable to be passed to the calling method.

Spaces are not allowed unless part of the value.

Array values are specified in the following format ['val1"val2'…'valN'] with no spaces between values. Example: String[]=['This"is"a"test'] is a list of four strings.

C.3 APLIC – License Manager

License Manager displays local license key information. The command should be used on every installation and does not query remote license information.

Table C-3. APLIC Options

Option	Description
-l	Fully qualified file license key file name, such as **license_key.jar**. By default `[AUTOPILOT_HOME]\lib\license_key.jar` is loaded.
-s scope	Displays license information for a specific scope (host name). Use this to validate license for specific host. By default scope is the name of the local host.
-c all|users|component	Shows license information for a specific component. Each component has a name. The default name is Node (CEP server). To display a list of all licensed components use "–c all" option. Use "–c users" option to display user account limitation.
-expired	Shows only licenses that are expired or out of scope.

Examples:

```
Show all available licenses: aplic –c all
Show the maximum number of licensed users: aplic –c users
Show all expired licenses: aplic –expired
Test the license for a specific host name: aplic –s HOST
```


C.4 APFACT – Fact Publisher

apfact utility publishes facts to a process wrapper, allowing users to instrument scripts and user applications without coding. The utility uses UDP protocol to send facts to the Process Wrapper. The Process Wrapper expert must be deployed and configured to listen on a designated UDP port. That port should be specified on the command line of **apfact**. It is recommended to use **apfact** and Process Wrapper on the same machine, due to limitations of UDP protocol. **apfact** and Process Wrapper should run on the local area network in close proximity (preferably on the same host) and should not be separated by routers (many routers and firewalls block UDP traffic).

Table C-4. APFACT Options

Option	Description
-nobanner	Do not show banner.
-verbose	Enables verbose mode.
-host	Host name of the CEP server (default=localhost) apfact –host [*host_name*]
-protocol udp | tcp	Protocol used to publish facts (default=udp)
-port	Listener port of the Process Wrapper Expert (default=6000) apfact –port [*port_number*]
-file	Input file containing a batch of facts apfact –file [*file_name*]
-factor *factor_value*	Acceleration factor or multiplier by which the time delays in the batch file are multiplied. For example, a .5 factor would cause half the time delays. The combination of factor and repeat allows the same batch file to be used to simulate varying fact generation rate and volume conditions.
-repeat *repeat_count*	Number of times to repeat batch (default=1)
-obj	Specifies common fact prefix followed by individual metrics.
-filter *filter_string*	Facts filter applied to batch (default=*)
-fact	fact1=value1[desc][act1:op1+act2=op2..],..., factN=valueN

Example apfacts Sent from the Command Line

1. `apfact -port 6010 -filter "CPU\*" -file input.txt`
   This example specifies a fact batch file is to be used. The filter qualifier specifies to only extract facts from the batch file that start with fact category "CPU."

2. `apfact -port 6010 -fact "CPU\IDLE=50"`
   CPU is the fact category and IDLE is an integer fact within the category that will appear in the Enterprise Manager nested under CPU.

3. `apfact -port 6015 -fact "CPU\IDLE=10[CPU Idle percent],CPU\BUSY=90"`
   Similar to example 2, but a description is included following the IDLE integer fact value. The description will appear in the fact properties. The additional integer fact BUSY will appear nested under CPU and aligned with IDLE.

4. `apfact -host 123.12.34.15 -port 6010 -fact "CPU\IDLE=10[CPU Idle]"`
   Similar to the IDLE fact of example 3, but a specific host IP address versus the default localhost is specified.

5. `apfact -fact "SERVER\Running=false[server state] [Start:START.BAT+Stop:STOP.BAT]"`

Defining Facts in a File

To define a batch of facts in a file, use the -file option and provide a fact file in the following format:

```
time1_in_ms fact1
time2_in_ms fact2
```

Example:

```
100 CPU\Idle=10
10 CPU\Busy=90,Swap Time=50
```

The input file is processed line-by-line. apfact waits for time_in_ms (millisecs) before publishing the fact, which must be specified in the format described by the –fact option. Each fact may require double quotes around it. Examples:

1. `apfact -port 6010 -factor 0.1 -file input.txt`
   Read and send the facts defined in file input.txt, multiplying the specified time delays by a factor of .1. For the example fact batch file shown above, the IDLE fact will be sent every 10 seconds instead of every 100 seconds, and the BUSY fact will be sent with a delay of 1 second instead of 10 seconds.

2. `apfact -port 6010 -repeat 10 -file input.txt`
   Repeat 10 times the reading and sending of facts defined in file input.txt.

3. `apfact -port 6010 -filter "CPU\*" -file input.txt`
   Read and send the facts in fact file input.txt , filtering to select only facts that start with fact category CPU.

C.5 ATPNODE – CEP Server

ATPNODE starts M6 CEP server component and normally runs in the background on all M6 hosts. **ATPNODE** can also be stopped/started from Windows Services. **ATPNODE** logs events and errors into [*AUTOPILOT_HOME*]\logs directory. All logs have name of CEP server as part of log name.

Table C-5. ATPNODE Options

{Option} Option_Value	Description
-w [work_dir]	Identifies the work directory where property files are located (default=.\) (example: -w [*AUTOPILOT_HOME*]\localhost)
-p [prop_file]	Fully qualified server property filename. Identifies a user defined property file Default is [*AUTOPILOT_HOME*]\localhost\.\node.properties
-d [{domain} dsURL]	Designates the domain server URL (bbns://server:port) -domain bbns://Domain_Server:2325
-b [daemon]	Flag that the node will run in background (default is true) [*AUTOPILOT_HOME*]\localhost\ATPNODE -daemon
-c [console]	Flag that the node will run in foreground (default=false) [*AUTOPILOT_HOME*]\localhost\ATPNODE -console
-s [safe]	Starts node in safe mode; all services are disabled (default is false) [*AUTOPILOT_HOME*]\localhost\ATPNODE -safe
-x [debug]	Prints out all java properties at startup.
-r [{registry} regx_file]	Fully qualified service registry XML file name. Allows switching active service directory. Default is [*AUTOPILOT_HOME*]\localhost\.\registry.xml
-l [logfacts]	Records all facts for all services that have the *Recording* option enabled. The file remains open until the server stops. However, will not record if property server.facts.jdbc.driver = com.mysql.jdbc.driver is specified. [*AUTOPILOT_HOME*]/localhost/[server_name]_Facts.fct
-f [profile]	Enables sensor profiler globally for computation of sensor profiling metrics (default=disabled)
-k [stop]	Stops services on CEP server gracefully.
-x [debug]	Enables debug mode and printing of stack traces/properties (default=false)
-a [strong]	Enables strong authentication; per request (default=false)
-e [export]	For the domain server, exports account information into a text file.
-z [import]	For the domain server, imports accounts from an XML export file.
-n [noupdate]	Upon start of program, will not obtain and install software updates.
-j [{join} grid_name]	Joins server to the specified grid.
-i [{priority} number]	Specifies grid priority. Used when selecting primary servers (default=0)
-o [{out.log} log_dir]	Overrides the default log directory (default=[*install_dir*]/logs)
-h [help] | -?| -usage	Displays all command line options.

ATPNODE -console:
Use the –console option to run the node in the foreground.
(Example: *[AUTOPILOT_HOME]*\localhost\ATPNODE -console)

ATPNODE -safe:
Use the –safe option to start a node in safe mode, where all services are in stop mode. Used when troubleshooting. (Example: *[AUTOPILOT_HOME]*\localhost\ATPNODE –console -safe)

ATPNODE -debug:
Use the –debug option to print all Java properties when starting the node. Used when debugging or troubleshooting. (Example: [*AUTOPILOT_HOME*]\localhost\ATPNODE –console -debug)

ATPNODE –r regx_file:
Use the –r regx_file option to switch CEP server active service directory. (Example:
[AUTOPILOT_HOME]\localhost\ATPNODE -console –r registery.xml)

C.6 ATPNAMES – Domain Server

ATPNAMES starts the M6 domain server component. **ATPNAMES** normally runs in the background on all M6 nodes. The CEP server can be started and stopped from Windows Services. **ATPNAMES** logs events and errors into *[AUTOPILOT_HOME]*\logs directory. All logs have the name of the domain server as part of the log name. (Refer to Chapter 3 for additional information.)

Table C-6. ATPNAMES Options

{Option} Option_Value	Description
-w [work_dir]	Identifies the work directory where property files are located (def=.\) (example: -w [*AUTOPILOT_HOME*]\naming)
-p [prop_file]	Identifies a user defined property file (default is node.properties) (example: -p node.properties)
-d [{domain} dsURL]	Designates the domain server URL (bbns://server:port) -domain bbns://Domain_Server:2325
-b [daemon]	Flag that the node will run in background (default=true) [*AUTOPILOT_HOME*]\naming\ATPNAMES -daemon
-c [console]	Flag that the node will run in foreground (default=false) [*AUTOPILOT_HOME*]\naming\ATPNAMES -console
-s [safe]	Starts node in safe mode; all services are disabled (default=is false) [*AUTOPILOT_HOME*]\naming\ATPNODE -safe
-x [debug]	Prints out all java properties at startup.
-r [{registry} regx_file]	Allows switching active service directory (default=registry.xml)
-l [logfacts]	Records all facts for all services that have the *Recording* option enabled. The file remains open until the server stops. However, will not record if property server.facts.jdbc.driver = com.mysql.jdbc.driver is specified. [*AUTOPILOT_HOME*]\naming\[server_name]_Facts.fct
-f [profile]	Enables sensor profiler globally for computation of sensor profiling metrics (default=disabled)
-k [stop]	Stops services on CEP server gracefully.
-x [debug]	Enables debug mode and printing of stack traces/properties (default=false)
-a [strong]	Enables strong authentication; per request (default=false)
-e [export]	For the domain server, exports account information into a text file
-z [import]	For the domain server, imports accounts from an XML export file
-n [noupdate]	Upon start of program, will not obtain and install software updates
-j [{join} grid_name]	Joins server to the specified grid
-i [{priority} number]	Specifies grid priority. Used when selecting primary servers (default=0)
-o [{out.log} log_dir]	Overrides the default log directory (default=[*install_dir*]/logs]
-h [help] | -?| -usage	Displays all command line options

ATPNAMES -console:

Use the –console option to run the domain server in the foreground.
(Example: [*AUTOPILOT_HOME*]\naming\ATPNAMES -console)

ATPNAMES -safe:

Use the –safe option to start a domain server in safe mode. Used when troubleshooting.
(Example: [*AUTOPILOT_HOME*]\naming\ATPNAMES -console -safe)

ATPNAMES -debug:

Use the –debug option to print all Java properties when starting the domain server.
(Example: [*AUTOPILOT_HOME*]\naming\ATPNAMES -console -debug)

Appendix D: M6 Best Practices

D.1 Naming Conventions

The following naming conventions are recommended by Nastel Technologies to give you optimum performance while using M6.

- Avoid using blanks in service, policy, and user definitions.
- Avoid using special characters such as **$ * # ; :** etc.
- Experts should be named as <*prefix*>_Monitor
- Managers should be named as <*prefix*>_Manager

  Under Manager *General* tab in *Properties*, select *Naming Convention* and enable *Enforce Naming Convention* so that all policies will have the same naming convention automatically applied by the manager.
- Policies should be named as "<*prefix*>_Policy"
- Use service *Context* property to categorize services by category and usage or purpose.

  Under Services *General* tab in *Properties*, enter category and usage or purpose in *Context* field.

D.2 Guidelines for Building Policies

D.2.1 Modeling

- Avoid hard-coded names and use Global environmental variables such as Java variables and CEP server variables (node.properties) and/or Policy environmental variables.
- Avoid duplication by promoting modularity and reuse.
  - Define and enforce naming conventions
  - Focus on creating modular, reusable business views
  - Create complex policies by combining simple policies.
- Avoid large and complex policies.
  - Build small, simple, self-contained policies
  - Link policies together – avoid using copy and paste
  - Reference policies by reusing existing logic.
- Perform complex event processing using Business Views
  - A Business View is an event processor and should be used as one.

D.2.2 Deployment

- Deploy policies close to their source by using
  - Same CEP server as the origin for most of the facts (performance)
  - Same server as the origin for most of the facts
- Always consider security
  - Who do you want the policy to be accessible to? Should it be available on the web? If so, consider the correct permission model
  - Edit policy security properties to set permissions and ownership.
- Each CEP server should:
  - Be self-sufficient and independent of other CEP servers
  - Contain all the monitors and policies for monitoring.
- Make use of multiple manager instances for:
  - Improving performance
  - Improving response time
  - Group by policy category or purpose

D.2.3 Usage - Reducing Rule and Processing Delays

- Avoid using blocking scripts within sensors because they will:
  - Cause rule execution delays
  - Decrease effective rule per second processing speed
- Avoid excessive dynamic sensor creation and deletion during policy execution.
  - Structure rules so that dynamic sensors are created only on error conditions and removed when condition is resolved.
- If possible, avoid sensors that use multiple facts for evaluation.
  - The more facts included in the sensor, the more processing required during rule execution.

D.3 Key Performance Indicators

D.3.1 System Calendar

The following metrics are published under <*NODE_NAME*>_Facts\Calendar.

Table D-1. System Calendar

Calendar	Type	Description
day_of_month	Integer	Current day count within the current month.
day_of_week	Integer	Current day count within the current week.
day_of_week_in_month	Integer	Current day of week count within the current month.
day_of_year	Integer	Current day count within the current year.
month	Integer	Month count within current year. For example: 6 corresponds to the month of June.
time_zone	String	Current time zone (EDT, CDT, GMT, etc.)
week_of_month	Integer	Current week count within the current month.
week_of_year	Integer	Current week count within the current year.
year	Integer	Current year.

D.3.2 Java Memory

The following metrics are published under <*NODE_NAME*>_Facts\Java.

Table D-2. Java Memory

Java	Type	Description
absolute_memory_utilization	Float	Ratio of memory used to maximum heap size. Multiply this number by 100 to get percentage of used memory.
free memory	Long	Number of bytes available memory based on the currently allocated heap size (**not** based on the maximum heap size).
heap_size_max	Long	Maximum memory that can be claimed by the server. If consumed 100%, then no more memory can be allocated, and CEP server will fail.
heap_utilization	Float	Current heap utilization. Multiply by 100 to percentage usage. If reaches 1.0 (100%), CEP server will no longer be able to claim more memory and will fail.
memory_growth_rate_sec	Float	Number of bytes claimed per second. If the number is positive, memory is growing. If the number is negative, memory is shrinking. The value is computed based on the previous sample (every 30 seconds).
memory_utilization	Float	Current heap utilization based on the currently allocated heap size which could be less than or equal to the maximum heap size.
overhead_usage_bytes	Float	Number of bytes used per functional unit which is defined as *fact + sensor.* This number can be used to estimate the required memory for a given number of facts + sensors. For example: if over_head_usage_bytes = 6000, then to support 10000 facts + 10000 sensors, the user will need 6000*(2000) bytes.
total_memory	Long	Total number of bytes allocated. This number will always be less than or equal to heap_size_max. This number will usually be close to the number of bytes claimed from the operating system.
total_thread_groups	Integer	Total number of allocated thread groups by the CEP server. Each group may hold 0 or more threads.
total_user_threads	Integer	Total number of threads running. Large number of threads may reduce CEP server performance due to thread management overhead.
used_memory	Long	Current number of bytes used by the CEP server within the Java Virtual Machine. This number does not indicate the amount of memory claimed from the operating system.

D.3.3 Database Logging

The following metrics are published under <*NODE_NAME*>_Facts\Logging:

Table D-3. Database Logging

Logging	Type	Description
db_arrival_rate_per_sec	Float	Number of SQL insert/update requests coming per second. Requests usually come from sensors which are enabled for logging.
db_delivery_rate_per_sec	Float	Number of SQL insert/update requests processed by the database.
db_flow	Boolean	Indicates whether database logging is enabled or disabled. The flow is disabled when db_queue_size reaches db_max_queue_limit to prevent memory exhaustion.
db_last_error	String	Last error encountered during SQL logging.
db_latency_ms	Long	Number of milliseconds that an SQL request sits on a queue before being processed by the database.
db_max_queue_limit	Integer	Maximum queue size for holding all outstanding SQL operations.
db_queue_size	Integer	Current number of outstanding SQL requests.
db_total_arrived	Long	Total number of issued SQL requests.
db_total_dropped	Long	Total number of dropped SQL requests. SQL requests are dropped when db_flow is false.
db_total_errors	Long	Total number of SQL related errors encountered during logging.
db_total_processed	Long	Total number of successfully processed SQL requests.

D.3.4 Sensor Performance

The following metrics are published under <*NODE_NAME*>_Facts\Sensor\Performance:

Table D-4. Sensor Performance

Sensor/Performance	Type	Description
absolute_rate_rules_per_sec	Float	Number of rules processed per second since start of the CEP server.
absolute_rate_sensors_per_sec	Float	Number of sensors processed per second since start of the CEP server.
last_sensor_exec_time_ms	Float	Time it took to execute last sensor (in milliseconds)
max_rule_exec_time_ms	Float	Longest time it took to execute a rule.
max_rule_hotspot	String	Name of rule associated with max_rule_exec_time_ms.
max_sensor_hotspot	String	Name of the sensor associated with max_rule_hotspot.
min_sensor_exec_time_ms	Float	Minimum time it took to execute a sensor.
rate_rules_per_sec	Float	Number of rules processed per second, based on the total time spent processing rules.
rate_sensors_per_sec	Float	Number of sensors processed per second, based on the total time spent processing sensors.
sensor_busy_percent	Float	Percentage of time used to process sensors.
sensor_idle_percent	Float	Percentage of sensors spent in idle state (not processing rules).
sensor_turn_around_time_ms	Long	Time it takes to process a given metric by one or more sensors. The larger the value the longer it takes from the point of metric discovery to it being processed by the sensor and showing up within business views. Growth of this value usually means that the rule engine is in overload and cannot keep up with the load.
total_processed_rules	Long	Total number of processed rules since last reset.
total_processed_sensors	Long	Total number of processed sensors since last reset.
total_sensor_time_ms	Long	Total time spent processing sensors since last reset (in milliseconds).

D.3.5 Sensor Runtime

The following metrics are published under <*NODE_NAME*>_Facts\Sensor\Runtime:

Table D-5. Sensor Runtime

Sensor/Runtime	Type	Description
sensor_arrival_rate_per_sec	Float	Number of sensor events queued up for processing per second.
sensor_delivery_rate_per_sec	Float	Number of sensor events processed per second.
sensor_dropped_events	Long	Total number of dropped sensor events due to overload.
sensor_failed_actions	Long	Total number of failed user actions (launched by sensors)
sensor_failed_notifications	Long	Total number of failed notifications (SMTP)
sensor_table_size	Long	Total number of outstanding sensor requests.
sensor_total_action_threads	Integer	Total number of threads allocated to execute user actions.
sensor_total_expiring_threads	Integer	Total number of threads associated with expiring sensors (dynamic child sensors)
sensor_total_notifications	Long	Total number of successfully sent notifications (SMTP)
sensor_total_running	Long	Total number of actively running sensors, including user defined and dynamically created sensors.
sensor_worker_active_threads	Integer	Total number of threads allocated to process all sensors.
sensor_worker_thread_pool_size	Integer	Total number of threads within the thread pool designated for processing rules.

D.3.6 Session

The following metrics are published under <*NODE_NAME*>_Facts\Session:

Table D-6. Session

Session	Type	Description
avg_pipe_arrival_rate_per_sec	Float	Average rate at which objects are published for delivery to remote clients (subscribers)
avg_pipe_delivery_rate_per_sec	Float	Average rate at which objects are delivered to remote clients (subscribers)
avg_pipe_usage	Float	Average communication usage across all connections. Multiply by 100 to get percentage value.
max_resp_time	Long	Maximum response time for remotely executed actions.
max_send_queue_limit	Long	Maximum queue size of holding outbound requests for each communication session (between client/server).
total_bytes_sent	Long	Total number of bytes sent across all connections.
total_objects_sent	Long	Total number of objects sent across all connections.
total_objects_dropped	Long	Total number of objects dropped due to communication overload, meaning the remote end unable to keep up with consumption of data.
total_objects_recvd	Long	Total number of objects received across all connections.
total_pipes	Integer	Total number of active communications between client and server.
total_pipes_disabled	Integer	Total number of pipes disabled due to communication overload. Disabled pipes will drop objects.
total_pipes_full	Integer	Total number of connections in full state – 100% capacity utilization – their max queues are utilized to 100%.
total_pipes_idle	Integer	Total number of pipes idle.
total_requests	Long	Total number of issued requests (commands) to remote locations.
total_resp_timeouts	Long	Total number of timed out requests.
total_send_queue_depth	Long	Total number of objects queued up for outbound delivery.
total_user_timeouts	Long	Total number of timed out user-initiated requests.

D.3.7 Topic

The following metrics are published under <*NODE_NAME*>_Facts\Topic:

Table D-7. Topic

Topic	Type	Description
expiringFacts	Long	Total number of expiring facts; facts marked for expiration.
fact_delivery_backlog	Long	Total number of facts waiting to be published to corresponding subscribers. Growing number indicates that the publishing is at much higher rate than can be consumed by subscribers.
fact_delivery_turn_around_time_ms	Long	Time it takes to deliver a given fact to a subscriber. Growing number indicates longer latency and delayed processing by the rule engine.
fact_publish_backlog	Long	Number of outstanding facts waiting to be published.
fact_request_backlog	Long	Total number of outstanding requests for fact lookup.
sensor_delivery_backlog	Long	Total number of sensors waiting to be processed by the rule engine. Growing number indicates that the rule engine is overloaded and may not process information in a timely fashion.
sensor_delivery_limit	Long	Maximum number of allowed sensors that can be in waiting state before being dropped. Once the sensor_delivery_backlog exceeds this limit, some sensor activity will be dropped to prevent CEP server from consuming large amounts of memory and failing.
sensor_request_backlog	Long	Total number of sensors waiting for metric lookup.
totalFacts	Long	Total number of published facts.
totalFilters	Long	Total number of unique filters used by all created sensors.
totalSubscribers	Long	Total number of subscribers, which includes sensors as well as remote subscribers.
utilization	Float	Fact utilization totalFacts/maxFacts. Multiply by 100 to get percentage used.
maxFacts	Long	Maximum number of facts allowed within the CEP server. The limit is soft and does not prevent publishing more facts.

This Page Intentionally Left Blank

Appendix E: Required Linux Platform Configurations

The following configurations are required on Linux platforms to ensure M6 runs smoothly and efficiently.

Table E-1. Linux Configurations

Description	Configuration
Newer glibc-2.2.x libraries cannot handle initial thread stack sizes greater than 6MB and can cause a segmentation fault. (Red Hat 7.0, Mandrake 8.0, SUSE 7.2, Debian 2.2)	Use "u-limit –s 2048" in bash shell or "limit stacksize 2048" in tcsh to limit the initial thread stack to 2MB.
A thread that is waiting on an I/O operation will not wake up if a file involved in the I/O operation is closed.	Set the J2SE_PREEMPTCLOSE environment variable to 1. J2SE_PJREEMPTCLOSE=1 export J2SE_PREEMPTCLOSE

This Page Intentionally Left Blank

Appendix F: Dashboard Database Schema

Table F-1. Historical Sensor Names		
Sensor Name	**Data Type**	**Default**
SID	integer	NOT NULL
SensorName	char (255)	NOT NULL
Severity	char (32)	NOT NULL
SensorValue	char (255)	NULL
NumericValue	float	NULL
Health	float	NULL
LogTime	datetime	NULL
EventID	integer	NULL
ServerName	char (255)	NULL
OSName	char (128)	NULL
OSVersion	char (12)	NULL
OSArch	char (12)	NULL
OS User	char (48)	NULL
APUser	char (48)	NULL
ServiceCategory	integer	NULL
ServiceType	char (48)	NULL
ObjectType	char (48)	NULL
Location	char (128)	NULL
SensorIndex	char (64)	NULL

Table F-2. Real-Time Sensor Names		
Sensor Name	**Data Type**	**Default**
SID	integer	NOT NULL
SensorName	char (255)	NOT NULL
Severity	char (32)	NOT NULL
SensorValue	char (255)	NULL
NumericValue	float	NULL
Health	float	NULL
LogTime	datetime	NULL
EventID	integer	NULL
ServerName	char (255)	NULL
OSName	char (128)	NULL
OSVersion	char (12)	NULL
OSArch	char (12)	NULL
OSUser	char (48)	NULL
APUser	char (48)	NULL
ServiceCategory	integer	NULL
ServiceType	char (48)	NULL
ObjectType	char (48)	NULL
Location	char (128)	NULL
SensorIndex	char (64)	NULL

Table F-3. Policy Names		
Policy Name	**Data Type**	**Default**
SID	integer	NOT NULL
ManagerName	char (128)	NOT NULL
PolicyName	char (128)	NOT NULL
DomainName	char (128)	NOT NULL
TableName	char (128)	NOT NULL
RssUrl	char (128)	NOT NULL
RssItemUrl	char (128)	NOT NULL
TworksUrl	char (128)	NOT NULL

Appendix G: Web Service Interface

The Web Service interface allows users to control their M6 environment directly from AutoPilot M6 Web in the following ways:

- Publish facts to M6
- Lookup facts, directory, service status
- Start and stop services

It is deployable on J2EE 1.4 compliant application servers: WebSphere Application Server, WebLogic Application Server and JBoss Application Server.

The web application package, *apws.war*, is located in [*AUTOPILOT_HOME*]/webservice. Additional information is located at either [*AUTOPILOT_HOME*]/webservice/doc or [*AUTOPILOT_HOME*]/webservice/apws_doc.zip.

Refer to the *AutoPilot M6 Installation Guide* for the installation procedure for each application server.

This Page Intentionally Left Blank

Appendix H: Derived Metrics

Table H-1. Derived Metrics		
Derived Metric	**Description**	**Formula**
Value	Current value of fact	
Name	Complete fact name (all tree levels, from root to leaf node)	
Class	Class of fact (e.g., java lang long)	
Length	Length of fact, in bytes	
Update-Count	Number of times the fact was updated (changed or same)	
Change-Count	Number of times the fact was changed	
Reset-Count	Number of times the fact was reset	
Previous-Value	Previous value of the fact	
Time-Created	Time the fact was created	
Last-Updated	Time of most recent update (changed or same)	
Last-Changed	Time of most recent change	
Update-Age	Time since last update	
Change-Age	Time since last change	
Time-Difference	Time difference in ms between fact publisher (origin) and subscriber	
Min	Overall minimum value since reset	
MAvg	Moving average	
Counter	Last actual value for a counter type, versus the delta reported	
Ignore-Status	Ignore status indicator – if ignored, sensor alerting will not trigger	
Time-Since-Reset	Time since reset	
Change-Latency	Time between latest changes	
Update-Latency	Time between latest updates	
Update-Velocity	Rate of update	
History-Size	Number of facts in history store	
History-Locked	If the history is defined but locked, new numbers will not be put into history when they arrive	
History-Max-Size	Maximum number of history samples	
History-Time	Time of fact history	

Table H-1. Derived Metrics		
Derived Metric	**Description**	**Formula**
History-Avg	Average of values in history facts	
History-EMAvg	Exponential Moving Average of values in fact history	=ema(last 10 samples)
History-Max	Maximum value in fact history	=max(all history samples)
History-Min	Minimum value in fact history	=min(all history samples)
History-Variance	Variance of values in fact history	=var(all history samples)
History-Deviation	Standard Deviation of values in fact history	=stdev(all history samples)
History-Dev-Mean	Number of standard deviations from the mean	current - mean(history))/stdev(history)
History-Bound	Upper bound based on Chebyshev in-equality	http://en.wikipedia.org/wiki/Chebyshev's_inequality
History-Band-High	High band based on Bollinger bands	http://en.wikipedia.org/wiki/Bollinger_bands
History-Band-Low	Low band based on Bollinger bands	http://en.wikipedia.org/wiki/Bollinger_bands
History-RSI	Relative Strength Indicator	http://en.wikipedia.org/wiki/Relative_strength_index
History-SO-K	Stochastic oscillator	http://en.wikipedia.org/wiki/Stochastic_oscillator
History-CAvg	Average percent change in history (based on % change)	
History-CVariance	Variance of values in fact history(based on % change)	
History-CDeviation	Standard Deviation of values in fact history (based on % change)	
History-CBound	Upper bound based on Chebyshev in-equality (based on % change)	
History-CDev-Mean	number of standard deviations from the mean (based on % change)	
History-CBand-High	High band based on Bollinger bands (based on % change)	
History-CBand-Low	Low band based on Bollinger bands (based on % change)	
History-CAvg-Gain	Average Percent Gain	
History-CAvg-Loss	Average Percent Loss	
History-CAD-Ratio	Ratio of Advances to Declines	# advances (increases) / # declines (decreases)
History-HROC	Historical rate of change percent	(current – old_history) / old_history
History-IROC	Instantaneous rate of change percent	(current – last) / last
Expiry-Time	Defined life of the fact in milliseconds. If the fact is not updated within this time, it will be deleted	

Table H-1. Derived Metrics		
Derived Metric	**Description**	**Formula**
Expiry-Timer	The amount of time left before it expires (when 0, it will be deleted)	

Glossary

Application Programming Interface (API): a source code interface that an operating system, library or service provides to support requests made by computer programs.

API: *see* Application Programming Interface.

Asynchronous: Communication between computer and devices that can occur at any time and at irregular intervals.

AutoPilot M6: Nastel Technologies' Enterprise Application Management Platform. AutoPilot M6 is designed to monitor and control distributed IT services such as application servers, middleware, user applications, workflow engines, brokers, Service Oriented Architecture (SOA) and Enterprise Service Bus (ESB) based applications and their impact on business services.

AutoPilot M6 for WMQ: Nastel Technologies' WebSphere MQ management solution. Re-designated as M6 for WMQ with release 6.0, prior releases retain the AP-WMQ or MQControl trademark.

AutoPilot M6 Web: AutoPilot M6 Web is a browser-based interface that provides monitoring and operational control over managed resources and applications. It allows users to monitor health, recover from a failure, view historical performance graphs, and visualize impacts of a failure.

BSV: *see* Business View

Business View (BSV): A collection of rules that define a desired state of an eBusiness environment. Business Views can be tailored to present information in the form most suited to a given user, as defined by the user.

CEP: *see* Complex Event Processing.

CEP Server: A container that can host any number of AutoPilot M6 services such as experts, managers, policies, etc.

Client: Any programming component that uses the AutoPilot M6 infrastructure; for example, the AutoPilot M6 User Console.

Complex Event Processing (CEP): A technology for building and managing event-driven information systems. CEP is primarily an event processing concept that deals with the task of processing multiple events from an event cloud with the goal of identifying the meaningful events within the event cloud. CEP employs techniques such as detection of complex patterns of many events, event correlation and abstraction, event hierarchies, and relationships between events such as causality, membership, and timing, and event-driven processes.

Console: The console acts as the graphical interface for AutoPilot M6.

Contacts: A subordinate to a given Manager or Expert.

Data Source Name (DSN): Logical name that is used by Open Database Connectivity (ODBC) to refer to the drive and other information that is required to access data. The name is use by Internet Information Services (IIS) for a connection to an ODBC data source, (Example: Microsoft SQL Server database). The ODBC tool in Control Panel is used to set the DSN. When ODBC DSN entries are used to store the connection string values externally, you simplify the information that is needed in the connection string. This makes changes to the data source completely transparent to the code itself.

Decision Support System (DSS): An AutoPilot M6-based service designed to monitor, store, and display any event information generated by AutoPilot M6-enabled middleware and applications.

Deploy: To put to use, to position for use or action.

Domain Server: A specialized CEP server that maintains the directory of CEP servers, experts etc. The domain server is also capable of hosting experts, managers etc.

DSN: *see* Data Source Name

DSS: *see* Decision Support System

Enterprise Service Bus (ESB): A software architecture construct, implemented by technologies found in a category of middleware infrastructure products usually based on standards that provides foundational services for more complex architectures via an event-driven and standards-based messaging engine.

ESB: *see* Enterprise Service Bus

EVT: Event Log file extension (e.g.: sample.evt)

Event: An Event is something that happens to an object. Events are logged by AutoPilot M6 and are available for use by AutoPilot M6 Policies or the user.

Expert: Services that monitor specific applications such as an applications server, web-server or specific components within the applications (example - channels in WebSphere MQ). Experts generate facts.

Facts: Facts are single pieces of data that have a unique name and value. One or more facts are used to determine the health of the object, application, or server.

File Monitor: Built-in expert that monitors application error logs and publishes logs to AutoPilot M6 as facts and events.

Graphical User Interface (GUI): A type of environment that represents programs, files, and options by means of icons, menus, and dialog boxes on the screen. The user can select and activate these options by pointing and clicking with a mouse or, often, with the keyboard. Because the graphical user interface provides standard software routines to handle these elements and report the user's actions (such as a mouse click a particular icon or at a particular location in text, or a key press); applications call these routines with specific parameters rather than attempting to reproduce them from scratch.

Grids: In AutoPilot M6, a collection of clusters that allows users to define and automate CEP server failover. Defined under domain server\naming\grid folder.

GUI: *see* Graphical User Interface.

IIS: *see* Internet Information Services

Internet Information Services (IIS): Microsoft's brand of Web server software, utilizing HTTP to deliver World Wide Web documents. It incorporates various functions for security, allows CGI programs, and also provides for Gopher and FTP services.

J2EE: *see* Java Platform, Enterprise Edition.

Java: A platform-independent, object-oriented programming language developed and made available by Sun Microsystems.

Java Database Connectivity (JDBC): The JDBC API provides universal data access from the Java programming language. Using the JDBC 2.0 API, you can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built. The JDBC *Test Tool* that was developed by Merant and Sun Microsystems may be used to test drivers, to demonstrate executing queries and getting results, and to teach programmers about the JDBC API.

Java Developer's Kit (JDK): A set of software tools developed by Sun Microsystems, Inc., for writing Java applets or applications. The kit, which is distributed free, includes a Java compiler, interpreter, debugger, viewer for applets, and documentation.

Java Management Extensions (JMX): Java technology that supplies tools for managing and monitoring applications, system objects, devices (e.g. printers) and service-oriented networks. Those resources are represented by objects called MBeans (for *Managed Bean*).

Java Message Service (JMS): Java message-oriented middleware API for sending messages between two or more clients.

Java Platform, Enterprise Edition (J2EE): A widely used platform for server programming in the Java programming language. The Java EE Platform differs from the Standard Edition (SE) of Java in that it adds additional libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

Java Server Pages (JSP): JSP technology enables rapid development of web-based applications that are platform independent. Java Server Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content. Java Server Pages technology is an extension of the Java™ Servlet technology.

Java Virtual Machine (JVM): The "virtual" operating system that JAVA-written programs run. The JVM is a hardware- and operating system-independent abstract computing machine and execution environment. Java programs execute in the JVM where they are protected from malicious programs and have a small compiled footprint.

JDBC: *see* Java Database Connectivity

JDK: *see* Java Developer's Kit.

JMS: *see* Java Message Service.

JMX: *see* Java Management Extensions.

JRE: JAVA Run-time Environment. The minimum core JAVA required to run JAVA Programs.

JSP: *see* Java Server Pages

JVM: *see* JAVA Virtual Machine.

LDAP: *see* Lightweight Directory Access Protocol.

Lightweight Directory Access Protocol (LDAP): a networking protocol for querying and modifying directory services running over TCP/IP.

Manager: Managers are the home or container for policies. All business views must reside on managers, and managers must be deployed prior to deploying a business view or policy.

Message-Oriented Middleware (MOM): Message-oriented middleware (MOM) is a client/server http://www.sei.cmu.edu/str/descriptions/clientserver.htmlinfrastructure that increases the interoperability, portability, and flexibility of an application by allowing the application to be distributed over multiple heterogeneous platforms.

Message Queue Interface (MQI): The Message Queue Interface (MQI) is part of IBM's Networking Blueprint. It is a method of program-to-program communication suitable for connecting independent and potentially non-concurrent distributed applications.

MOM: *see* Message-Oriented Middleware.

MQI: *see* Message Queue Interface

Naming Service: A common server records "names" of objects and associates them with references, locations, and properties.

Object Request Broker (ORB): a piece of middleware software that allows programmers to make program calls from one computer to another via a network.

ODBC: *see* Open Database Connectivity.

Open Database Connectivity (ODBC): Provides a standard software API method for using database management systems.

ORB: *see* Object Request Broker.

Package Manager: The command line utility that allows users to list, install, uninstall, verify, and update AutoPilot M6 installation on any CEP server.

PKGMAN: *see* Package Manager

Policy/Business Views: Business views are a collection of one or more sensors. Business views are used to visually present the health and status of the different systems as well as automatically issue remedial actions.

Process Wrapper: Built-in wrapper that monitors a process or script started by AutoPilot M6.

Sensor: A rule that is used to determine the health of an object or application based on one or more facts. Actions can then be issued, based on the health.

Service-Oriented Architecture (SOA): An evolution of distributed computing and modular programming. SOAs build applications out of software services. Services are relatively large, intrinsically unassociated units of functionality, which have no calls to each other embedded in them. They typically implement functionalities most humans would recognize as a service, such as filling out an online application for an account, viewing an online bank statement, or placing an online book or airline ticket order. Instead of services embedding calls to each other in their source code, protocols are defined which describe how one or more services can talk to each other. This architecture then relies on a business process expert to link and sequence services, in a process known as orchestration, to meet a new or existing business system requirement.

Simple Mail Transfer Protocol (SMTP): A TCP/IP protocol for sending messages from one computer to another on a network. This protocol is used on the Internet to route e-mail. *See also* communications protocol, TCP/IP. *Compare* CCITT X series, Post Office Protocol.

SMTP: *see* Simple Mail Transfer Protocol

SOA: *see* Service-Oriented Architecture

Synchronous: communication within a computer that occurs at regular intervals. Usually governed by the microprocessor clock.

TCP/IP: *see* Transmission Control Protocol/Internet Protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP): A protocol developed by the Department of Defense for communications between computers. It is built into the UNIX system and has become the de facto standard for data transmission over networks, including the Internet.

UDP: *see* User Datagram Protocol.

User Datagram Protocol (UDP): A connectionless protocol that runs on top of IP networks. Provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. It's used primarily for broadcasting messages over a network.

Virtual Machine (VM): Software that mimics the performance of a hardware device, such as a program that allows applications written for an Intel processor to be run on a Motorola chip. *Also See* Java Virtual Machine.

Visual Source Safe (VSS): Microsoft VSS helps you manage your projects by saving them to a database. When you need to share files between two or more projects, you can share them quickly and efficiently. When you add a file to VSS, the file is backed up on the database, made available to other people, and changes that have been made to the file are saved so you can recover an old version at any time. Members of your team can see the latest version of any file, make changes, and save a new version in the database.

VM: *see* Virtual Machine

VSS: *see* Visual Source Safe.

WebSphere MQ: IBM's message queuing product.

Websphere_MQ_Manager: A specialized manager capable of hosting one or more MQSeries specific policies, apart from the regular policies.

Wireless Application Protocol (WAP): An open global specification that is used by most mobile telephone manufacturers. WAP determines how wireless devices utilize Internet content and other services. WAP enables devices to link diverse systems contents and controls.

XEN: A free software virtual machine monitor for IA-32, x86-64, IA-64, and PowerPC architectures. It is software that runs on a host operating system and allows several guest operating systems to be run on top of the host on the same computer hardware at the same time. Modified versions of Linux and NetBSD can be used as hosts

.

Index

K

L

M

N

O

P

R

S

T

U

V

