

System Management Hub

Configuration

Version 9.5 SP1

November 2013

This document applies to System Management Hub Version 9.5 SP1.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1999-2013 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://documentation.softwareag.com/legal/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Document ID: SMH-ADMIN-95SP1-20130929

Table of Contents

| | |
|---|----|
| Preface | v |
| I Before You Start Using System Management Hub | 1 |
| 1 Post-Installation Environment | 3 |
| Windows | 4 |
| UNIX | 4 |
| 2 Registry Settings | 5 |
| Registry Master Key | 6 |
| Section and Key Names | 6 |
| Syslog Parameters (UNIX only) | 17 |
| 3 Installing the Readme File | 19 |
| 4 Defining New Administrators | 21 |
| 5 Starting and Stopping Services | 23 |
| Windows | 24 |
| UNIX | 25 |
| II Security | 27 |
| 6 Overview of Security in System Management Hub | 29 |
| MIL Security | 30 |
| Host Security | 31 |
| Product Security | 31 |
| 7 Secure Sockets Layer (SSL) in System Management Hub | 33 |
| Overview of SSL | 34 |
| Registry Entries for Windows | 36 |
| Registry Entries for UNIX | 36 |
| SSL Workflow | 36 |
| Troubleshooting Guidelines | 37 |
| Error Messages | 38 |
| 8 Authentication in System Management Hub | 41 |
| Software AG Security eXtensions (SSX) | 42 |
| Functionality of the SSX module | 42 |
| SSX in System Management Hub | 43 |
| Authentication Scenarios | 44 |
| Configuring LDAP Authentication with Technical User Credentials | 45 |
| Configuring SSX Authentication with Internal User Repository | 46 |
| Verifying Local Security Configuration | 48 |
| 9 Single Sign-On (SSO) in System Management Hub | 49 |
| Authentication Workflow | 50 |
| IAF Service | 51 |
| IAF Configuration Parameters | 51 |
| IAF Configuration in SSX | 52 |
| III Logging Facilities | 55 |
| 10 System Logging Facilities | 57 |
| Windows Logging Facility | 58 |
| UNIX Logging Facility | 59 |

| | |
|--|----|
| 11 System Management Hub Logging Service | 71 |
| Event Layer Logging | 72 |
| Changing the Database Path | 73 |

Preface

This document describes how to configure and administer the System Management Hub environment.

The information is organized under the following headings:

Before You Start Using System Management Hub

Security

Logging Facilities

I Before You Start Using System Management Hub

This chapter provides information on steps to follow after the installation.

The information is organized under the following headings:

Post-Installation Environment

Registry Settings

Installing the Readme File

Defining New Administrators

Starting and Stopping Services

1 Post-Installation Environment

- Windows 4
- UNIX 4

Windows

On Windows, System Management Hub does not modify any environment variables.

UNIX

The installation procedure generates a customized environment script, `argenv` and updates the global `sagenv` environment. For more information see the *Readme.txt* in the UNIX directory of System Management Hub distribution media.

The following variables define the System Management Hub environment:

| Variable | Description |
|----------|--|
| SAG | The base directory for all Software AG products. |
| REGFILE | The registry storage file. |

2 Registry Settings

- Registry Master Key 6
- Section and Key Names 6
- Syslog Parameters (UNIX only) 17

This chapter describes the registry settings you can customize according to the requirements of your installation.

System Management Hub provides the user with a web interface to add, delete, or modify registry keys. For details on this feature, see *Working with the Registry* under *Web Interface*.

Registry Master Key

| | |
|------------|---|
| Master Key | HKEY_LOCAL_MACHINE \ SOFTWARE \ Software AG \ System Management Hub |
|------------|---|

Section and Key Names

The section names and key names are identical on every platform. Only the following keys are customizable:

- Global
- Client/Server Layer Server
- Management Independent Layer
- HTTP Layer
- Batch Layer Client
- Event Layer Server
- Cleanup Intervals
- SNMP Layer
- Java Layer
- Java Parameters
- Java Agents
- Net Drives
- Session Termination Callbacks

Global

The section name is *Global*.

| Key | Description |
|-------------|--|
| ArgLogDir | as String "\Software AG_directory\InstanceManager\log" as default |
| Config_Edit | as String "1" as default |

| Key | Description |
|-------------------|---|
| Def_Agents_Path | as String the default agent path |
| Install_Path | as String the base directory for the current installation |
| Language | as String predefined I18N support; currently only the English version, "EN", is supported. |
| PKCS12File | as String " <i>\Software AG_directory\InstanceManager\files\server.p12</i> " as default |
| PKCS12Pass | as String PKCS12 password field |
| Rgs_Edit | as String enable/disable registry editing on the local machine 0 disable (default) 1 enable |
| Rot_View | as String enable/disable 0 disable (default) 1 enable |
| Sag_User_Name | as String (UNIX only) pre-defined user ID for the execution of all managed products; set by the installation procedure |
| SSL | as String enable/disable 0 disable (default) 1 enable |
| SSL_Server_Cnf | as String " <i>\Software AG_directory\InstanceManager\files\ssl\server.cnf</i> " as default |
| Support_Data_Path | as String defines the default directory location for EPOST support files. " <i>\Software AG_directory\InstanceManager\DiagnosticFiles</i> " as default |
| Use_ICU | as String enable/disable 0 disable (default) 1 enable |

| Key | Description |
|-----------------------|--|
| Version | as String the current main version |
| Windows_Network_Logon | as String enable/disable 0 disable (default) 1 enable |
| Xds_Edit | as String enable/disable 0 disable (default) 1 enable |

Client/Server Layer Server

The section name is *CSLayerServer*.

| Key | Description |
|------------------------|--|
| Agent_Maximum_XML | as String "3000000" as default |
| Agent_stderr | as String controls diagnostics trace facilities within the agents "0" - disabled (default) "1" - enable trace, write trace data fo file (defined by user) "2" - enable trace, write trace data to STDERR "3" - enable trace, write trace data to BOTH file and STDERR |
| Agent_stderr_directory | as String defines the directory where agent trace (if enabled) is written. " <i>Software AG_directory\InstanceManager\log</i> " as default |
| Agent_Timeout | as String "1800000" as default |
| Agent_tmp_directory | as String defines the directory used by agents when they need to create temporary files or data during processing. " <i>Software AG_directory\InstanceManager\tmp</i> " as default |
| AutomatedScheduler | as String enable/disable 0 disable (default) 1 enable |

| Key | Description |
|----------------|---|
| AutomatedTasks | as String enable/disable 0 disable (default) 1 enable |
| Config_Edit | as String enable/disable 0 disable (default) 1 enable |
| Http | as String built-in HTTP mode "0" - disabled (default) "1" - enabled |
| Http_Port | as String 10015 (default) System Management Hub HTTP port |
| JRE_Path | as String the full path of the used native JVM library (jvm.dll, libjvm.so) |
| Java | as String Java bridge mode "0" - disabled "1" - enabled (default) |
| Java_Agents | as String the full path of the Java plug-in |
| Java_Image | as String " <i>Software AG_directory</i> \jvm\jvm160_32\bin" as default |
| Java_Switch | as String enable/disable 0 disable (default) 1 enable |
| Java_Wrapper | as String " <i>Software AG_directory</i> \InstanceManager\bin\argjwrapper.exe" as default |
| New_StartAgent | as String "0" - to use old agent invocation "1" - to use new agent invocation (default) |

| Key | Description |
|------------------------|--|
| SIN_Config_File | as String "\Software AG_directory\InstanceManager\conf\cim_jaas.properties" as default |
| Security | as String "0" (disabled) "1" - logging through the local FTP daemon; an FTP daemon must be available on the same machine "2" - use the native user authentication provided by the operating system (default) |
| Snmp | as String Snmp interface control "0" - disabled "1" - enabled (default) |
| Snmp_Interface | as String the full path of the Snmp plug-in |
| Snmp_Port | as String "10016" (default) the SNMP port |
| Tcp_Ip_Port | as String 10012 (default) System Management Hub server port |
| Tcp_Ip_Queue_Size | as String pre-defined queue size, default "100" |
| Tcp_Ip_Recv_Timeout | as String pre-defined timeout in seconds for the TCP/IP recv call, default "180" |
| Template_Agent_Enabled | as String enable/disable false - disable true - enable (default) |
| Trace | as String controls trace file "0" - disabled (default) "1" - enable trace, write trace data to file (defined by user) "2" - enable trace, write trace data to STDERR "3" - enable trace, write trace data to BOTH file and STDERR |

| Key | Description |
|------------------|---|
| Trace_Byte_Level | as String controls trace of TCP/IP and IPC communication, active only if Trace is enabled "0" - disabled (default) "1" - enabled |
| User_Cmd | as String user command control "0" - disabled "1" - enabled (default) |

Management Independent Layer

The section name is *MILayer*.

| Key | Description |
|----------------|--|
| Batch | as String built-in batch processing mode "0" - disabled "1" - enabled (default) |
| Config_Edit | as String enable/disable "0" - disabled "1" - enabled (default) |
| Hostlist_Path | as String the directory of the host list configuration file (<i>hostlist.xml</i>) |
| Http_Port | as String "10010" (default) The Runtime HTTP port |
| Snmp | as String SNMP interface control "0" - disabled "1" - enabled (default) |
| Snmp_Interface | as String the full path of the SNMP plug-in |
| Snmp_Port | as String "10018" (default) the SNMP port |

| Key | Description |
|-------------------------|--|
| Snmp_Proxy_Port | as String "10019" (default) the SNMP proxy port |
| SupportPlugIn | as String support plug-in control "0" - disabled "1" - enabled (default) |
| Support_session_timeout | as String Timeout for an inactive support session in seconds, for example, 300 (default is 900) |
| Tcp_Ip_Port | as String "10013" (default) The Runtime server port |
| Tcp_Ip_Recv_Timeout | as String pre-defined timeout in seconds for the TCP/IP recv call, default 180 |
| Url | as String "localhost " (default) the URL or IP address of the Runtime external HTTP server, for example, <i>http.softwareag.com</i> , if the Runtime uses an external HTTP server, this key must point to the HTTP host machine. |

HTTP Layer

The section name is *HTTPLayer*.

| Key | Description |
|-------|--|
| Trace | as String control trace file generation for the internal HTTP server "0" - disabled (default) "1" - enabled |

Batch Layer Client

The section name is *BatchLayerClient*.

| Key | Description |
|---------------------|---|
| Tcp_Ip_Recv_Timeout | as String pre-defined timeout in seconds for the TCP/IP recv call, default "60000" |

Event Layer Server

The section name is *EventLayer*.

| Key | Description |
|-----------------|--|
| Config_Edit | as String enable/disable "0" - disabled "1" - enabled (default) |
| Logging_Path | as String the logging path for event messages |
| RefreshRate | as String "5" (default) defines the default rate (in seconds) between page updates for running jobs when displayed by the Events and Job Monitor agent |
| ShowDebug | as String controls the display of additional diagnostics and trace information in the event log display messages "0" - disabled (default) "1" - enabled |
| ShutdownTimeout | as String delay for registered plug-ins to shut down in seconds (default is "900") |
| Snmp | as String SNMP interface control 0 disabled 1 enabled (default) |
| Snmp_Interface | as String the full path of the Snmp plug-in |

| Key | Description |
|---------------------|--|
| Snmp_Port | as String "10017" (default) the SNMP port |
| Tcp_Ip_Port | as String "10014 " (default) The Event Dispatcher server port |
| Tcp_Ip_Queue_Size | as String pre-defined queue size (default "100") |
| Tcp_Ip_Recv_Timeout | as String pre-defined timeout in seconds for the TCP/IP recv call, default "60" |
| Trace | as String controls trace file "0" - disabled (default) "1" - enabled |

Cleanup Intervals

The section name is *EventLayer\Cleanup*.

| Key | Description |
|---------|--|
| Info | as String, single value between 1 and 59 (see note and table below) "30" (default) The number of days to retain messages of type Info |
| Warning | as String, single value between 1 and 59 (see note and table below) "30" (default) The number of days to retain messages of type Warning |
| Error | as String, single value between 1 and 59 (see note and table below) "30" (default) The number of days to retain messages of type Error |
| Fatal | as String, single value between 1 and 59 (see note and table below) "30" (default) The number of days to retain messages of type Fatal |



Note: A single value of 60 or greater specifies a clean-up interval in seconds. A single value of 59 or less specifies a clean-up interval in days.

The clean-up interval can be specified in a DD HH:MM:SS format. The following formats are supported:

| Time Format | Example | Meaning |
|-------------|-----------|--|
| SS | 61 | 61 seconds. For backward compatibility if the single value is less than 60, the value is assumed to be days. |
| MM:SS | 10:20 | 10 minutes, 20 seconds |
| HH:MM:SS | 1:10:20 | 1 hour, 10 minutes and 20 seconds |
| DD HH:MM:SS | 5 1:10:20 | 5 days, 1 hour, 10 minutes and 20 seconds |

SNMP Layer

The section name is *SNMPLayer*.

| Key | Description |
|----------------------|--|
| Config_Edit | as String enable/disable "0" - disabled "1" - enabled (default) |
| Mib_Path | as String the path of the internal MIB mapping configuration |
| NotificationContext | as String "public" as default |
| NotificationProtocol | as String enable/disable "0" - disabled (default) "1" - enabled |
| ReadCommunity | as String "public" as default |
| ReadWriteCommunity | as String "private" as default |
| ThreadModel | as String enable/disable "0" - disabled (default) "1" - enabled |
| Trace | as String control trace file for the Snmp plug-ins |

| Key | Description |
|-----------------------------|--|
| | "0 "disabled (default) 1 enabled |
| TraceSnmp | as String control trace file for the Snmp engine "0" - disabled (default) "1" - enabled |
| TraceSnmpAgent | as String control trace file for the Snmp MIBs "0" - disabled (default) "1" - enabled |
| TraceSnmpAgentDebugFilter | as String "6" as default |
| TraceSnmpAgentErrorFilter | as String "5" as default |
| TraceSnmpAgentEventFilter | as String "5" as default |
| TraceSnmpAgentInfoFilter | as String "5" as default |
| TraceSnmpAgentWarningFilter | as String "5" as default |
| Update_Timeout | as String pre-defined timeout in seconds for the update process for the internal MIB tables, default 65 |

Java Layer

The section name is *CSLayerServer*.

Java Parameters

The section name is *CSLayerServer\JVM_Parameters*.

The section includes the list of the JVM parameters, name=value.

Java Agents

The section name is *CSLayerServer\Java_Agents*.

The section includes the list of the agent packages, package_descriptor=package_full_path

Net Drives

The section name is *CSLayerServer\NetDrives*.

Session Termination Callbacks

The section name is *CSLayerServer\Session_Callbacks*.

This section includes a list of plug-ins for use by the Client/Server Layer server.

Syslog Parameters (UNIX only)

System Management Hub provides the facility to use syslog on UNIX systems for event messages. You can configure the message severity that is written to syslog with the registry string value "Logmask" under:

HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\Syslog

This value consists of a combination of the following characters that generate the desired level of logging:

| Character | Severity | Syslog Severity | Default |
|-----------|-------------------------------------|--------------------------|---------|
| F | ARGUS_SV_SEVERE_ERROR | LOG_CRIT | X |
| E | ARGUS_SV_ERROR | LOG_ERR | X |
| W | ARGUS_SV_WARNING | LOG_WARNING | |
| I | ARGUS_SV_INFO | LOG_INFO | |
| J | Open Job Messages of all severities | Appropriate LOG_severity | |

Syslog Severity Table

If any of these characters are present in this registry value, the event (or job information) is written in the syslog. To disable, set this value to an empty string.

Example: `Logmask = EFW` logs only warning and error messages.

UNIX Syslog

The *syslogd* daemon writes the syslog. When *syslogd* starts up, it reads its configuration file (the file */etc/syslog.conf*) to determine what kind of events to log and where to log them.

For detailed information about your current configuration file (the parameters of the *syslog.conf* file), run `man syslog.conf`.

There is no standard for the message output file and no utility to automatically read the correct file.

▶ To read the syslog:

- 1 Open the file */etc/syslog.conf* and check if there are any messages.
- 2 As a root user (or a user with root permissions), open the plain text syslog output file in any editor or run "cat", "more", "less", etc.
- 3 Select "Software AG".

For more information on logging on UNIX, see *UNIX Logging Facility*.

3

Installing the Readme File

You can view and download the readme file with the Installer or from the company site.

▶ **To download the latest product readme with the Software AG Installer**

- 1 Check (tick) the box **Documentation** in the Installer product selection menu.
- 2 **Main Product Readmes** install documentation readmes for the Software AG Installer, CentraSite, Tamino XML Server, System Management Hub, and Web Services Stack.
- 3 Click **Next** to start the download.

▶ **To download the latest product readme from the company site**

- 1 Go to *<http://documentation.softwareag.com/webMethods>*.
- 2 Scroll down the screen to the **Documentation by Product** section.
- 3 Click on the **Readmes** folder and choose **System Management Hub Readme** to start the download.

4 Defining New Administrators

This chapter describes the rules for granting administrator rights to users.

The system administration feature on a host level is a global functionality for all products that are managed by System Management Hub. By default, the system administrator can define administrators for the various products.

Product security is specific to each managed host. The installation procedure of System Management Hub requires the definition of an initial System Management Hub administrator who is identified uniquely by a user ID. On Windows, you can also use a domain account; for example, domain/user or domain\user.

After installation, only the user identified by this ID can manage System Management Hub. However, this initially-defined administrator can define new administrators or grant administrator rights to other users.

► **To define new System Management Hub administrators:**

- 1 Log on to the target host (initial administrator).
- 2 Select System Management Hub as the product to be managed.
- 3 Assign the rights using the following commands:
 - **Add administrator**
 - **Modify administrator**
 - **Delete administrator**

For more information on how to display, add, or delete administrators for System Management Hub and for the other installed products, see *Managing Administrator Accounts in System Management Hub*.

If you want to define new System Management Hub administrators, you must consider the following:

- When you install a product, you become automatically an administrator for that product.
- Administrators who install System Management Hub are automatically added as System Management Hub administrators.
- If you are an administrator, you can add administrators only for the product for which you act as an administrator.
- The administrators who are added later by the administrator of the product have administrator's rights only for their product.
- When you add a new administrator, you must specify at least one product for which to act as an administrator.
- The administrators for a product can be deleted as administrators for that product only.
- Two Administrators are registered after installation - the Install user and the Administrator user with a default password "manage".

5 Starting and Stopping Services

- Windows 24
- UNIX 25

The following section provides details about the services (daemons) available under Windows and UNIX after you have completed successfully a full installation of System Management Hub.

Windows

On Windows, the System Management Hub services are registered to start up automatically at system start.

Following is a table with the System Management Hub services:

| | |
|---|--------------|
| Software AG Runtime Service | sagctp95_1.1 |
| Software AG System Management Hub Service | sagcim95_1 |

▶ **To modify/start/stop the services, from the Windows Desktop**

- 1 Click *Start > Settings > Control Panel > Administrative Tools > Services*.
- 2 Select the service.

Right-click **Startup Type**, point to **Properties** and click.

- 3 Select **Automatic**, **Manual**, or **Disabled** from the **Startup type** drop-down menu. The recommended type is **Automatic**.
- 4 Click **OK**.

For the most recent information on the Windows installation procedure, please read the file *Install.txt* in the Windows directory of System Management Hub distribution media.

▶ **To start and stop System Management Hub manually**

- 1 Start System Management Hub script

`<SAGROOT>/InstanceManager/bin/startup.bat`

- 2 Stop System Management Hub script

`<SAGROOT>/InstanceManager/bin/shutdown.bat`

UNIX



Note: Before you start the daemon processes on UNIX, you need to set sufficient data user limits for the shell which starts the System Management Hub and Runtime daemons. Having an insufficient data user limit (`ulimit -d`) results in an `OutOfMemoryError` java exception at startup. For SMH, it is recommended that you have a data limit of at least 200000 (or higher). For more information, see the man page for `ulimit` or ask your system administrator.

The installation registers the daemons for System Management Hub and Runtime in the UNIX init structure so that it starts automatically at boot time. The following scripts are installed:

For Linux and Solaris

- `/etc/init.d/sag<n>cim95`
- `/etc/rc<R>.d/K20sag<n>cim95`
- `/etc/rc<R>.d/S60sag<n>cim95`

For AIX

- `/etc/sag<n>cim95`
- Entry in `/etc/inittab`:

```
sag<n>cim95:<R>: wait:/etc/sag<n>cim95 start > /dev/console 2>&1
```

For HP-UX

- `/sbin/init.d/sag<n>cim95`
- `/sbin/rc<R>.d/K20sag<n>cim95`
- `/sbin/rc<R>.d/S60sag<n>cim95`



Note: In the preceding script names, `<R>` refers to the system runlevel and `<n>` refers to a number which gets incremented by 1 for each installation on the local machine. The un-installation process will remove the scripts and links which correspond to the installation. To temporarily deactivate a service, you need to remove or rename those files manually. Native configuration tools like the Yast Run-Level-Editor on Linux do not work.

Following are System Management Hub daemons:

| Name | Description |
|--|---|
| <SAGROOT>/InstanceManager/conf/wrapper | Software AG System Management Hub Wrapper Service |

Following are Runtime daemons:

| Name | Description |
|------------------------------|---|
| <SAGROOT>/common/bin/wrapper | Software AG System Management Hub Runtime Wrapper Service |



Note: Each main daemon may have several child processes.

▶ To start and stop System Management Hub manually

- 1 Start System Management Hub daemons

```
<SAGROOT>/InstanceManager/bin/startup.sh
```

- 2 Stop System Management Hub daemons

```
<SAGROOT>/InstanceManager/bin/shutdown.sh
```


II Security

This chapter provides information on how System Management Hub handles security issues. It describes general and specific security facilities and how those facilities are implemented in System Management Hub.

System administrators will benefit most from the contents of the topics identified in this chapter.

The information is organized under the following headings:

Overview of Security in System Management Hub

Secure Sockets Layer (SSL) in System Management Hub

Authentication in System Management Hub

Single Sign-On (SSO) in System Management Hub

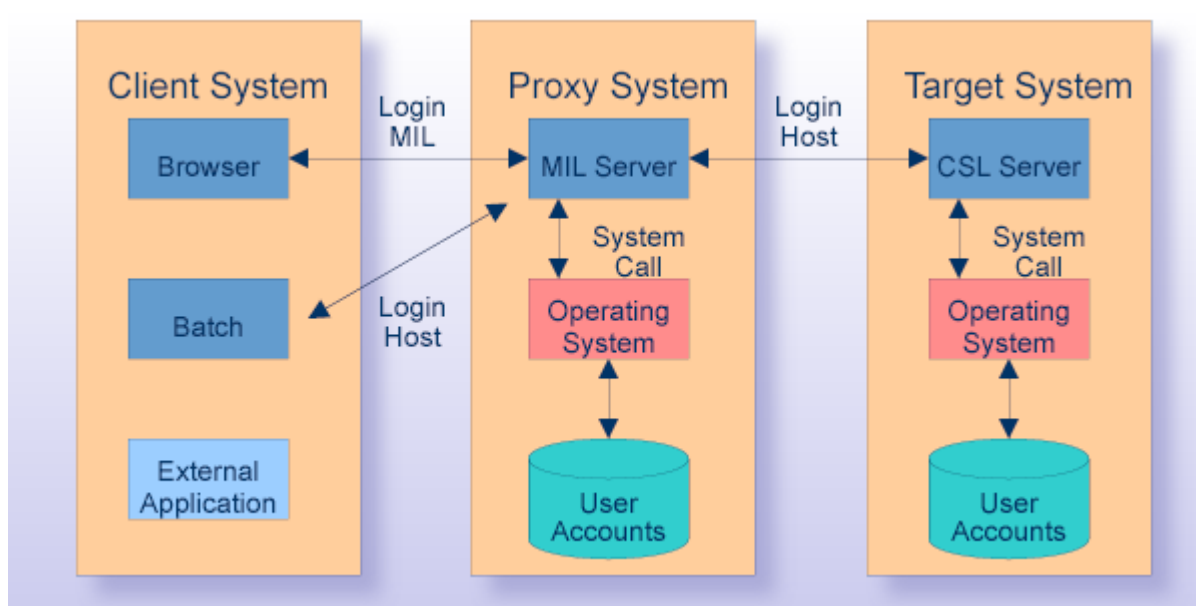
6 Overview of Security in System Management Hub

- MIL Security 30
- Host Security 31
- Product Security 31

The basic idea of security is to establish a secure connection between a server and a client so that reliable messages can be exchanged. According to the requirements of the system, this exchange of messages is designed for two types of authentication:

- Authentication of the client to the server
- Authentication of the server to the client

System Management Hub's functionality includes authentication for accessing the managed distributed systems. This authentication is necessary to start and execute the required services.



System Management Hub, with its n-tier architecture, distinguishes between several security levels.

This chapter covers the following topics:

MIL Security

This is the first level of user authentication to provide access to the generic System Management Hub's proxy system. When you work under web or batch interface, you must provide valid user information (usually a user name and a password) for an existing user account on the proxy machine (on which the appropriate MIL server is running).

The MIL server is responsible for verifying that the user information is registered on that machine. It uses the machine's operating system-specific (native) method. There are also other validation methods that can be configured by the system administrator. For details, see the sections [Product Security](#) and [Registry Settings](#).

Validation can be disabled at this level but this is not recommended except for testing purposes.

When you log on to the MIL, you establish a connection and create a session. Otherwise, no work with the system is possible.

Host Security

After a connection with the MIL server has been established, authentication is performed on the target machine on which you install the server-side components of System Management Hub.

System Management Hub's Client/Server Layer Server is responsible for verifying that the user information provided (usually a user name and a password) is actually registered on that machine, using that machine's operating system-specific (native) method. There are also other validation methods available that can be configured by the system administrator. See the sections [Product Security](#) and [Registry Settings](#).

If logging on to the target host is successful, a connection is established and it is possible to run the agents on it, to display products, and to manage System Management Hub. The product itself is then capable of setting up additional specific validations (see the next paragraph).

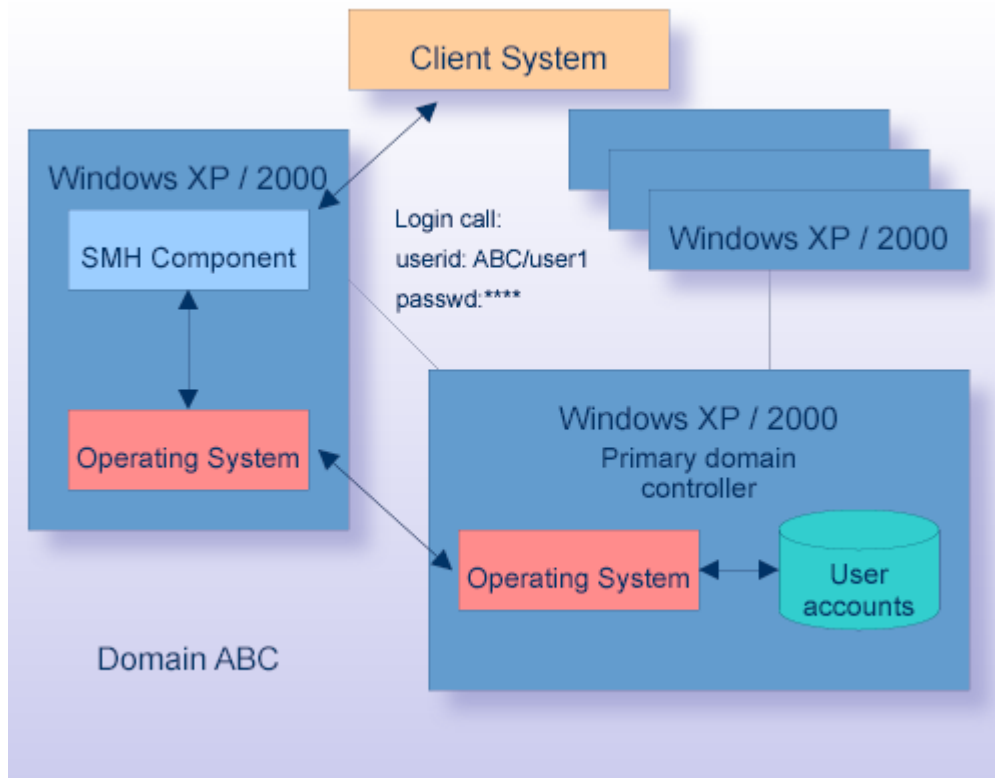
Product Security

Product security is specific to each managed host. By default, the installation procedure creates an initial System Management Hub administrator. To see how to create new administrators or assign administrator rights to other users, see [Defining New Administrators](#).

Security on Windows

Authentication on Windows is performed in the same way as on other platforms.

If both the MIL machine and the target machine or only one of them belongs to a Windows domain, you can log on to the domain instead of to the local machine. Use the domain name in the user name string in the standard format: `<domain_name>/<userid>`. Both, the forward slash "/" and the back slash "\" characters can be used as delimiters.



Secure Sockets Layer

In order to use Secure Sockets Layer (SSL), you must configure Runtime to use SSL.

When you configure SSL in Apache, you have a secure HTTPs connection between the client system and the proxy system. To have a secure connection from the proxy system to a product, you need to configure an SSL certificate.

For information on how to create or upload an SSL certificate, see [Secure Sockets Layer \(SSL\) in System Management Hub](#).

Software AG Security Extension (SSX)

SSX is Software AG' interface that contains functionality for authentication. The interfaces are written in C, and Java. Its components provide support for JAAS and the IAF service.

For details on the SSX module in System Management Hub, see [Authentication in System Management Hub](#).

7 Secure Sockets Layer (SSL) in System Management Hub

| | |
|--------------------------------------|----|
| ▪ Overview of SSL | 34 |
| ▪ Registry Entries for Windows | 36 |
| ▪ Registry Entries for UNIX | 36 |
| ▪ SSL Workflow | 36 |
| ▪ Troubleshooting Guidelines | 37 |
| ▪ Error Messages | 38 |

Overview of SSL

Originally developed by Netscape, the Secure Sockets Layer (SSL) protocol has been universally accepted on the Internet for authenticated and encrypted communication between clients and servers. It is included as part of both Microsoft and Netscape browsers and most web servers.

SSL protects data transferred over HTTP using encryption enabled by a server's SSL certificate. The SSL certificate contains unique authenticated information about the certificate's owner and verifies his or her identity.

An SSL certificate is a digital document that consists of a digital signature to bind together a public key with an identity - information such as the name of a person, an organization, an address, etc. A public key is used to encrypt information and a private key is used to decipher it. When a browser points to a secured domain, an SSL handshake authenticates the server and the client and establishes an encryption method, and a unique session key.

A trust store must be created to keep the certificates that the other party trusts to verify messages. These certificates (or trust store entries) contain the keystores that the client and server must have during the SSL handshake.

The keystore is like a container for the public and private keys that the application using SSL protocol utilizes along with additional information in order to sign the messages.

Server and Client Configurations

Usually, this is done by default. All web browsers use SSL over HTTP.

Encryption Algorithms (Ciphers)

The client and the server use the same algorithms to encrypt all the communication during the session. SSL protocol uses a combination of public keys and symmetric keys for data encryption. Symmetric key encryption is much faster than public-key encryption, but public-key encryption provides better authentication techniques.

SSL protocol supports the use of different cryptographic algorithms (or ciphers) for authenticating the server and the client to each other, transmitting certificates and establishing session keys. Clients and servers may support different sets of ciphers depending on the SSL version and the company policies for encryption strength.

SSL Handshake

The SSL handshake protocol determines how the server and the client negotiate the sets of ciphers that they use to authenticate each other. The result of this authentication is the transmitting of certificates and establishing session keys between them.

An SSL session always begins with an exchange of messages called the SSL handshake. The handshake tampers detection during the authentication session and does the following actions:

- The server authenticates itself to the client using public-key techniques
- The client and the server cooperate in the creation of symmetric keys used for rapid encryption (and decryption)
- (optional) Authentication of the client to the server

SSL Limitations

Following are several of the limitations of SSL:

- Latency

Latency is increased with the SSL connection since there are additional round trips added in order to establish authentication. This is only a fraction of a second for each connection, so it is not an issue on most small or medium businesses. For large business processing, however (with numerous transactions per minute) this can add up, especially if a client authentication is required.

- Bandwidth

Bandwidth considerations are not an issue for most small- or medium-sized businesses. SSL transactions can, however, increase by about 1 KB each. This is an issue with large numbers of transactions.

- Processor Usage

Processor usage is increased with SSL connections. This is an issue only on servers running numerous transactions per minute. Processor usage can also be minimized by using the most processor-efficient encryption methods.

Registry Entries for Windows

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\Global]
```

```
"PKCS12File"="{{ARGDIR}}\files\server.p12"  
"SSL"="0"  
"SSL_Server_Cnf"="{{ARGDIR}}\files\ssl\server.cnf"
```

`{{ARGDIR}}` is the System Management Hub installation directory. By default, it is the *Software AG_directory\InstanceManager* directory.

Registry Entries for UNIX

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\Global]
```

```
"PKCS12File"="{{ARGDIR}}/files/server.p12"  
"SSL"="0"  
"SSL_Server_Cnf"="{{ARGDIR}}/files/ssl/server.cnf"
```

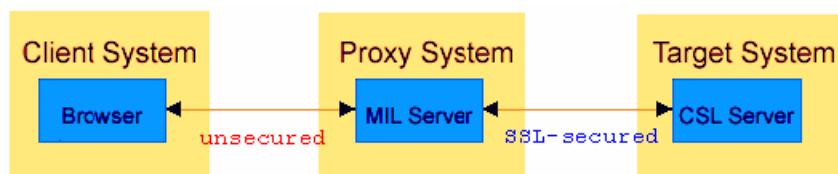
`{{ARGDIR}}` is the System Management Hub installation directory. By default, it is *Software AG_directory\InstanceManager*.

SSL Workflow

There are two separate connections that have to be secured:

- The connection between the client server and the proxy
- The connection between the proxy and the target machine

System Management Hub provides you with a secure SSL-encrypted connection only between the proxy and the target system. If the SSL configuration on the web server is not enabled, the connection between the client server (web or batch interface) and the proxy is not secured.



To have a secure connection from the proxy system to a managed host, you must enable the SSL functionality and configure System Management Hub to either create or upload an SSL certificate.

For more details on how to do that, see *Enabling SSL Support in System Management Hub*.



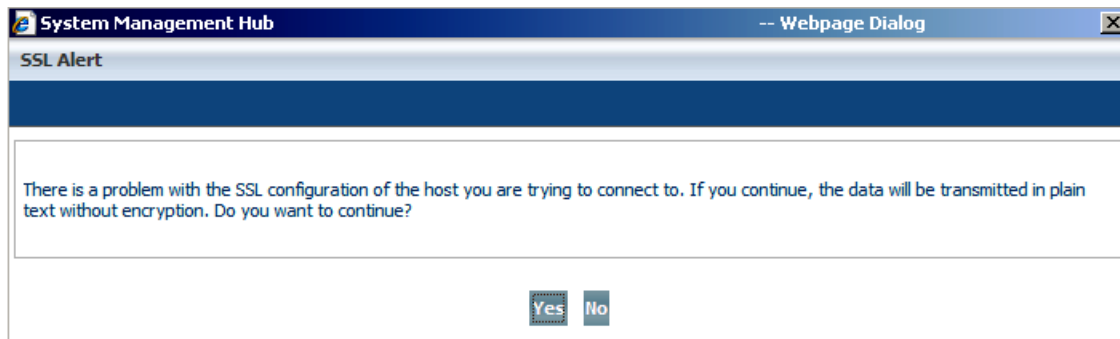
Note: In order to use Secure Sockets Layer (SSL) between the client server and the proxy, you must configure Runtime to use SSL.

Troubleshooting Guidelines

Following is a list with the possible configuration problems:

- Missing or wrong registry entry for PKCS12 keystore location
- Internal error in the SSL engine
- Missing private key file part of the keystore
- Wrong PKCS12 password
- Extraction of the certificate from the keystore is not possible
- Extraction of the private key from the keystore is not possible

The SSL message on the following screen capture registers problems with your host' SSL configuration:



If you select **Yes**, you continue the session without encryption of the information transmitted between MIL and the Client/Server Layer. The system remembers your choice, and it does not notify you again if the same problem occurs.

If you select **No**, you cannot run commands in batch interface. Ignore this error message by choosing **Yes**.

For information on how to ignore SSL error messages in batch interface, see *Batch Commands*.

To diagnose the exact cause of the problem, you must activate the trace log function. Access the registry master key (HKEY_LOCAL_MACHINE\\SOFTWARE\\Software AG\\System Management Hub) and change the settings of two of the customizable keys of the Client/Server Layer. They are disabled ("0") by default.

| Key | Description | Settings |
|------------------|---|--|
| Trace | <i>String</i> ; Controls trace file. | <ul style="list-style-type: none"> ■ "0" disabled (default); ■ "1" enable trace, write trace data to file (defined by user); ■ "2" enable trace, write trace data to STDERR; ■ "3" enable trace, write trace data to BOTH file and STDERR. |
| Trace_Byte_Level | <i>String</i> ; Controls trace of TCP/IP and IPC communication, active only if Trace is enabled. | <ul style="list-style-type: none"> ■ "0" disabled (default); ■ "1" enable. |

Switch Trace and Trace_Byte_Level keys to "1" (enabled) in your Client/Server Layer registry branch.

The trace log path parameter is *Agent_stderr* directory. This directory defines the location where agent trace (if enabled) is written.

Following is a sample of a log file with an error message for a faulty SSL configuration:

```
15:38:59.609 (2320)NET DRIVES: ARGUSUtlNetEnumerateDrives: ret TRUE
15:38:59.625 (2320)SSL error: Can't get a PKCS12 certificate
15:39:02.406 (2320)Flow : Thread started: 1
```

Error Messages

| Error message | Description |
|----------------------------------|--|
| Empty Public Certificate File | The certificate file part of the keystore is missing. |
| Empty Private Key File | The private key file part of the keystore is missing. |
| SSL_CTX_use_certificate | This prefix points to internal error in SSL engine. |
| SSL_CTX_use_PrivateKey | This prefix points to internal error in SSL engine. |
| PKCS12 File entry is missing | The registry entry for PKCS12 keystore location is wrong or missing. |
| Can't get the PKCS12 password | The PKCS12 password is wrong. |
| Can't get the PKCS12 certificate | It is not possible to extract the certificate from the keystore. |

| | |
|----------------------------------|--|
| Can't get the PKCS12 private key | It is not possible to extract the private key from the keystore. |
|----------------------------------|--|

8

Authentication in System Management Hub

- Software AG Security eXtensions (SSX) 42
- Functionality of the SSX module 42
- SSX in System Management Hub 43
- Authentication Scenarios 44
- Configuring LDAP Authentication with Technical User Credentials 45
- Configuring SSX Authentication with Internal User Repository 46
- Verifying Local Security Configuration 48

The information is organized under the following headings:

Software AG Security eXtensions (SSX)

Software AG Security eXtensions (SSX) is a user database interface that contains functions for user authentication and for the retrieval of repository objects. The main benefit of the interface is that it gives a client a uniform view to all the different user databases that implement it. The interface is written in C and Java and consists of a set of libraries that allow authentication against different systems (for example, LDAP, AD, and OS).

SSX authenticates a user by providing credentials. Its native functions retrieve repository data and administer functions to users and groups.

There are two SSX modules for authentication:

- In the Pluggable UI

The Pluggable UI is installed with the installation of System Management Hub. By default, it is set to authentication against the operating system.

- In the Client/Server Layer of System Management Hub

SSX is disabled by default in System Management Hub 9.0. For guidelines on how to enable SSX in System Management Hub 8.0, see [SSX in System Management Hub](#).

For guidelines on how to configure SSX to use IAF service for SSO authentication, see [IAF Configuration in SSX](#).

Functionality of the SSX module

Following is an overview of the basic functionality of Software AG Security eXtensions:

- Authentication of a user with a given password and, optionally, a domain name
- Handling users from all domains
- Distinguishing users from all domains when multiple user repositories are involved in a large scale application
- Enumerating all local groups that the user belongs to
- Enumerating groups and users of the specified user database
- Manipulation of entries of the specified user database
- Handling of repository requests through IAF
- Filtering support in the repository

- Specifying an output file for logging
- Specifying the user database that you want to work with:
 - "OS" for the native operating system
 - "LDAP" for an LDAPserver
- (LDAP only) Specifying an LDAP server type; this sets internally appropriate defaults. Available server types are:
 - ActiveDirectory
 - SunOneDirectory
 - OpenLDAP
- Graphical user interface to select the desired LDAP server type:
- Specifying how long an authenticated user entry must remain in the cache (the time in seconds)
- Specifying the number of invalid logon attempts before any further authentication attempts are blocked
- Specifying the maximum number of cached users authenticated successfully. When the cache overflows, the oldest entry is removed

SSX in System Management Hub

System Management Hub 9.0 comes with SSX disabled by default. It can be enabled via the web interface, the batch interface, or the registry.



Note: You must have a valid admin user to use the web interface or run batch commands.

For more details on how to set up the SSX configurations in System Management Hub via the web interface, see the *SSX Configuration* of the Client/Server Layer under *Web Interface*.

For more details on how to set up the SSX configurations in System Management Hub via the batch interface, see *Configuring SSX* under *Batch Commands* of the *Batch Interface*.

▶ To enable SSX via the registry

- 1 Open the Registry Editor.
- 2 Switch the registry key SSX_Enabled from "0" to "1":

```
HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management  
Hub\CSLayerServer\SSX_Enabled
```

- 3 Restart CSLayer service.

If you experience logon problems after enabling SSX authentication, change the SSX authentication mode of the target machine, or of the application from which you are trying to access System Management Hub, or both. For more information about the different settings and their affects, see *SSX Authentication Scenarios*.

SSX has a separate log file that gives additional information. If you still cannot solve the problem, change the SSX logging level and send the logging file to Software AG Support.

▶ To change SSX Logging Level registry

- 1 Open the Registry Editor.
- 2 Switch the registry key SSX_Log_Level from "1" to "6":

```
HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management  
Hub\CSLayerServer\SSX_Log_Level
```

- 3 Restart CSLayer service.



Note: The log file is located at `<SAGROOT>/common/arg/log/SSX.log`

Authentication Scenarios

There are two SSX modules for authentication - one in the Pluggable UI, and another one in the Client/ Server Layer of System Management Hub.


The Pluggable UI is installed with the installation of System Management Hub, but its SSX authentication module is different from System Management Hub's SSX module. The two SSX modules authenticate users against a different authentication type (LDAP, Active Directory, or operating system). By default, the SSX authentication module in the Pluggable UI is set to authentication against the operating system, while the one in System Management Hub 9.0 is disabled.



Important: If you change the authentication mode on the target machine (for example, from "default" to "LDAP"), you affect the authentication for accessing other products (for example, CentraSite Registry/ Repository, the Application Server Tier components, CentraSite Control).

To change the authentication mode on the target machine, you must have administrator's rights. Following are some of the possible authentication scenarios to illustrate the fact that System Management Hub and the Pluggable UI have different SSX authentication modules:

| Pluggable UI SSX Authentication set to... | Target Machine SSX Authentication set to... | Provides this logon scenario... |
|---|---|---|
| OS | LDAP | You can log on to the Pluggable UI on the target machine using the operating system's user name and password. To use System Management Hub, you must authenticate again with your domain credentials. |
| OS | OS | You can log on to the Pluggable UI on the target machine using the operating system's user name and password. To use System Management Hub, you do not have to authenticate again. |
| LDAP | OS | You can log on to the Pluggable UI on the target machine with your domain credentials. To use System Management Hub, you must authenticate again using the operating system's user name and password. |
| LDAP | LDAP | You can log on to the Pluggable UI on the target machine with your domain credentials. To use System Management Hub, you do not have to authenticate again. |

 **Important:** With System Management Hub, using the Pluggable UI with SSX authentication set to "OS" is sufficient for most cases. However, on UNIX systems, only default encryption of user passwords is possible for System Management Hub authentication. SSX supports all of them, so you can enable SSX in those cases. When using the Pluggable UI with SSX authentication set to a value other than "OS", use the same authentication type for System Management Hub. Enable SSX authentication and set it to the same authentication value as is the Pluggable UI SSX.

Configuring LDAP Authentication with Technical User Credentials

You configure authentication via technical user to access and search for users on LDAP servers that do not support anonymous queries. The following task allows you to provide and configure technical user settings in your System Management Hub.

▶ To authenticate against an LDAP server using technical user credentials and SSX

- 1 Create a technical user credential file.

For more information about creating technical user credential files, see the *Software AG Security Infrastructure* documentation.

- 2 Start System Management Hub web interface in a web browser.
- 3 Click **Local Security configuration**
- 4 On the **Context** dropdown menu, select one of the login contexts that are available in the *jaas.config* file. The following list outlines the default login context that are available in the *jaas.config* file. However, depending on the use case, the file can contain other login contexts.

- **SSXLoginOS.**

Use this default login context to define the default login modules that you want to use for authentication on the platform.

- **PluggableUI.**

Use this login context to define the login modules that you want to use to authenticate against the Pluggable UI of System Management Hub.

- 5 On the **Authentication Login Module options** area, select **Available Login Module** radio button.
 - 6 On the dropdown menu, select **LDAP (SSXLoginModule)**.
 - 7 Click **Add**.
 - 8 On the **Effective Login Modules** area, select **LDAP (SSXLoginModule)** option.
 - 9 Click **Configure**.
 - 10 On the **Options for LDAP configuration of SSXLoginModule** dialog, configure the following properties:
 - **useLdapTechUser**
A Boolean parameter which default value is false. The parameter is optional and allows you to enable the usage of a technical user.
 - **techLdapUserCredFile**
The parameter is mandatory if you enable the usage of a technical user. It specifies the path of the technical user credentials file.
 - **techLdapUserKeyFile**
The parameter is optional and specifies the path of the alternative key file.
- For more information about configuring `SSXLoginModule` settings, see the *Software AG Security Infrastructure* documentation.
- 11 Click **OK**.
 - 12 Click **Apply Changes**.

Configuring SSX Authentication with Internal User Repository

You can configure SSX authentication via internal user repository. The following task allows you to provide and configure internal user repository to a login context that is used in System Management Hub. The internal repository text file is an alternative to the OS and LDAP repositories. It is recommended to use an internal repository only during the initial setup of all required components or until you configure a real repository.

► **To authenticate using internal user repository in SSX**

- 1 Create an internal user credential file.

For more information about creating technical user credential files, see the *Software AG Security Infrastructure* documentation.

- 2 Start System Management Hub web interface in a web browser.

- 3 Click **Local Security configuration**

- 4 On the **Context** dropdown menu, select one of the login contexts that are available in the *jaas.config* file. The following list outlines the default login context that are available in the *jaas.config* file. However, depending on the use case, the file can contain other login contexts.

- **SSXLoginOS.**

Use this default login context to define the default login modules that you want to use for authentication on the platform.

- **PluggableUI.**

Use this login context to define the login modules that you want to use to authenticate against the Pluggable UI of System Management Hub.

- 5 On the **Authentication Login Module options** area, select **Available Login Module** radio button.

- 6 On the dropdown menu, select **Internal Repository (SSX)**.

- 7 Click **Add**.

- 8 On the **Effective Login Modules** area, select **Internal Repository (SSX)** option.

- 9 Click **Configure**.

- 10 On the **Control flag** dropdown menu, set the flag of the login module. Valid values are:

- **required**

- **requisite**

- **sufficient**

- **optional**

For more information about the control flag of login modules, see the *Software AG Security Infrastructure* documentation.

- 11 On the **Internal Repository** dialog, click **Manage**.

- 12 On the **Manage Local Repository** area, proceed as follows.

- To add a user, click **Add**.

On the dialog that opens, provide new user name and password, and click **OK**.




Note: When you enter a user name or a password, you can use only digits, Latin letters, and the following characters: ! () - . ? [] _ ~.

- To edit existing user credentials, select a user entry and click **Change Password**.

On the dialog that opens, provide a new password and click **OK**.

- To delete a user entry, select the entry and click **Delete**.

For more information about configuring `SSXLoginModule` settings, see the *Software AG Security Infrastructure* documentation.

- 13  **Important:** Once you confirm the changes and click the **Store** button, the changes are saved in an external file on the file system. At a later stage, you cannot revert the changes that are stored to the file by choosing the **Reset** button.

Click **Store**.

- 14 Click **Apply Changes**.

Verifying Local Security Configuration

When you configure a login context, you can verify that logging context by executing it using real user credentials against a real Pluggable UI or `SSXLoginOS`.

▶ To verify the configuration you provide

- 1 On the **Context** dropdown menu, select one of the login contexts that are available in the `jaas.config` file. The following list outlines the default login context that are available in the `jaas.config` file. However, depending on the use case, the file can contain other login contexts.

- **SSXLoginOS.**

Use this default login context to define the default login modules that you want to use for authentication on the platform.

- **PluggableUI.**

Use this login context to define the login modules that you want to use to authenticate against the Pluggable UI of System Management Hub.

- 2 Enter user credentials that you want to use with the configured login context.
- 3 Click **Verify Configuration**

9 Single Sign-On (SSO) in System Management Hub

- Authentication Workflow 50
- IAF Service 51
- IAF Configuration Parameters 51
- IAF Configuration in SSX 52

This chapter provides details on the installation and configuration of the SSO feature in System Management Hub.

SSO manages centrally user authentication. Once the user is authenticated, the application passes a token and / or an artifact that is used by the IAF server.

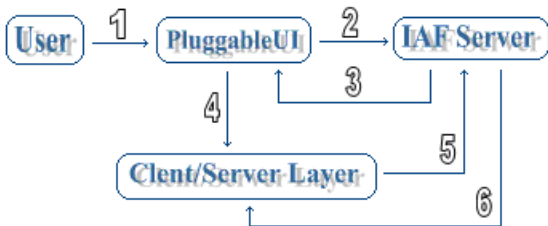
The architecture of IAF defines a central service (the IAF service). It creates the IAF token that contains all the information about the authenticated user and the IAF artifact that is a short index to the information about the user. In System Management Hub, the SSX configuration provides a GUI for setting the IAF service.

The information is organized under the following headings:

Authentication Workflow

After successful authentication in CTP on the IAF server, each plug-in of the Pluggable UI is provided with a user name and an IAF artifact instead of a password. These credentials are used for authentication on the Client/Server Layer.

The diagram below illustrates the SSO workflow in System Management Hub:



Following is an explanation of the different steps of the SSO authentication.

► To authenticate on the system using SSO

- 1 A user tries to authenticate on the system using the authentication interface provided by the Pluggable UI. (--1-->)
- 2 The Pluggable UI sends a request for a token to the IAF server. (--2-->)
- 3 The IAF server sends back the token to the Pluggable UI. (--3-->)
- 4 The token is sent to the Client/Server Layer of System Management Hub. (--4-->)
- 5 The Client/Server Layer sends a request for token validation to the IAF server. (--5-->)
- 6 The IAF server validates the token and returns it to the Client/Server Layer. (--6-->)

IAF Service

The IAF service is contacted by multiple clients in order to do the following:

- Authenticate a user
- Create a unique and fraud-resistant token
- Create an artifact for the existing token
- Validate tokens or artifacts
- Pass information about the token and the authenticated user to the owner of the token

The installation of the IAF service comes with the product installation of a Software AG application. The service is disabled by default on Windows and UNIX.

▶ To Enable the IAF Service on Windows and UNIX

- 1 On Windows, right-click **My Computer** and select **Manage -> Services and Application -> Software AG Integrated Authentication Framework Service**.
- 2 On UNIX, run `/etc/init.d/sag<n>iafd start/stop`.

▶ To use the IAF service

- Pass the URL of the IAF server to the application.

IAF Configuration Parameters

Following are the custom parameters for IAF configuration in the SSX login module:

| Parameter | Description |
|-----------------|--|
| IAFserverHost | <p>Host of the IAF server.</p> <p>Specify the host name (plus SSL port) of the IAF server.</p> <p>To do this, either use <code>IAFserverHost</code> and <code>IAFcertLocation</code> in combination or <code>serverHost</code> only.</p> <p>For example, <code>IAFserverHost="vmsec02:11958:SSL?TRUST_STORE=\$IAFCertLocation&VERIFY_SERVER=N"</code></p> |
| IAFCertLocation | <p>Location of the IAF certificate.</p> <p>For example, <code>IAFCertLocation="C:/Software AG/common/runtime/security/certs/IAFCaCert.pem"</code>.</p> |

| Parameter | Description | Mandatory |
|---------------|---|-----------|
| serverHost | <p>The host of the server.</p> <p>The combination of <code>IAFserverHost</code> and <code>IAFcertLocation</code> provide the same configuration as the <code>serverHost</code> parameter.</p> <p>Specify the <code>serverHost</code> parameter if the location of the IAF certificate (that is, the value of the <code>IAFcertLocation</code> parameter) is used directly in the declaration of the <code>IAFserverHost</code> parameter.</p> <p>For example, <code>serverHost="vmsec02:11958:SSL?TRUST_STORE=C:/Software AG/common/runtime/security/certs/IAFCaCert.pem&VERIFY_SERVER=N"</code>.</p> | Yes |
| localCodePage | <p>Local code page for IAF communication.</p> <p>This is required because the transport protocol encoding is UTF-8.</p> <p>On IBM mainframes, the default code page is "IBM_037", elsewhere it is "ISO8859-1".</p> | No |
| homeDir | <p>Locates the broker stub module and the crypto library.</p> <p>SSX looks for specific libraries. The <code>homeDir</code> parameter specifies the path to those libraries and loads them dynamically.</p> <p>Following is a list of the libraries for the different operating systems:</p> <ul style="list-style-type: none"> ■ Windows broker32.dll / sagssxtomcrypt.dll ■ UNIX broker.so/sl / libsagssxtomcrypt.so/sl ■ z/OS BROKER31 / SSXCTC <p>For example (Windows):</p> <pre>homeDir="C:\<Software AG_directory>\webMethods\<DIR_DLL_Libraries>"</pre> | No |

IAF Configuration in SSX

To use SSO in System Management Hub, you must configure SSX to use the IAF service for authentication.

The configuration panel for the IAF service is under the **SSX Configuration** menu of the **Client/Server and Agent Layer**:

If the SSX configuration is disabled, you must use the **Enable SSX authentication** option (that appears in the place of **Disable SSX authentication**) to activate it.

Following is a screen capture of the default IAF configuration:

▶ **To set the authentication type to IAF**

- 1 Right-click the **SSX Configuration** menu.
- 2 Select **Modify SSX authentication**.
- 3 In the right panel of the screen, set the authentication type to **IAF**.

Click **Next** to go to the text field for entering the IAF host. System Management Hub offers a tooltip for entering the correct format for the IAF host. Point to the text field to see it:

Click **Finish** to complete the configuration.

III Logging Facilities

This chapter describes the logging facility in System Management Hub.

Advantages of the Logging Facility

Following are the major advantages of the logging facility for the optimum performance of a system:

- Maintaining system integrity (if the services are aborted due to an error)
- Identifying who has performed a certain action and at what time
- Tracking the changed data elements

▶ To manage log files

- 1 Ensure that all your log files are copied to your backup media on a regular basis.

The timing of the backups must be such that any file that is periodically reset is copied to the backups before the reset is performed.

- 2 Review the logs regularly to note when a problem occurred.

This can help discover problems with your hardware, with your network configuration, and with your security.

- 3 Filter the messages you look at and reduce them to a more manageable collection, otherwise you can easily miss an important message.

- 4 Do not trust your log files completely!

Log files can be altered or deleted by an intruder who obtains super user rights.

The information is organized under the following topics:

System Logging Facilities

System Management Hub Logging Service

10 System Logging Facilities

- Windows Logging Facility 58
- UNIX Logging Facility 59

The information is organized under the following headings:

Windows Logging Facility

The logging facility on Windows operating systems is the Event Viewer.

▶ To trace log messages of the installed products with the Event Viewer

- 1 Enable the logging registry key of the Client/Server Layer by setting *Enable* to "1"

The default registry key setting of the Client/Server Layer after installation of System Management Hub is [HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\CSLayerServer\Syslog] "Enable"="0".

- 2 Set the *logmask* values to use the Windows logging service by setting *Logmask*.

Logmask specifies which messages go to the system log file.

For example: *Logmask* = WEF stands for a warning with error and fatal messages logged.

Logmask can be a combination of the following:

```
I: info message
  W: warning message
  E: error message
  F: fatal message
  J: job message
```

The default registry key setting of the Event Layer after installation of System Management Hub is [HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\Syslog] "Logmask"="FE".

Following is the location of the log files:

■ Location of Client/Server Layer log files

```
"Agent_Stderr_Directory"="Software AG_directory\InstanceManager\log"
```

■ Location of Event Layer log files


```
"Logging_Path"="Software AG_directory\InstanceManager\log\events"
```

UNIX Logging Facility

The logging facility on UNIX operating systems is called syslog.

The information is organized under the following headings:

- [Overview of Syslog](#)
- [Syslog Fields](#)
- [Interface Configuration](#)
- [Samples of Configuration Files](#)

Overview of Syslog

UNIX has a centralized system logging process that runs the program */etc/syslogd* or */etc/syslog*. Individual programs that need to have information logged send the information to syslog. The big advantage of syslog is that the message file holds all reported messages. By using scripts, a system administrator can detect complex error scenarios and act accordingly.

The syslog daemon is native to all distributions of UNIX and Linux, but is not available under Windows. It can be configured to write messages depending on their facility and severity to different log files. Additionally, it can discard them totally or forward them to a chained syslog daemon.

Following are several limitations to the system log facility:

- The message size is limited to 1K
- No assumption is made upon the formatting or contents of the messages. The protocol is simply designed to transport these event messages. Since the format of the message is not standardized, there is a large variety of message contents.
- Traditional syslog works well over a non-congested network.
- Syslog protocol is UDP-based, so it is unreliable and does not guarantee you the delivery of the messages. Expect some packet loss when the network is congested, the sender or receiver is busy, the WAN network is slow, or there is heavy syslog traffic.
- A machine that is not configured correctly sends syslog messages to a syslog daemon representing itself as another machine. This means the status of the supposed sender of the messages is not accurately reflected in the received messages.

Syslog Fields

The full format of a syslog message has three distinct parts - priority, header, and message.

- [The Priority Field](#)
- [The Header Field](#)
- [The Message Field](#)

The Priority Field

This is a number that represents the facility and the severity of the message.

With the facility and the severity values, you can apply certain filters on the events in the syslog daemon.



Note: The syslog daemon does not generate the severity and the facility values. They are generated by the applications on which the event is generated. If the message received by the syslog daemon has a generated code by the application that is different from the recommended code for the specified situations, then the syslog daemon receives the message without changing the code.

Facility of Syslog

The facility is the application or operating system component that generates a log message. It is defined by the syslog protocol and provides a basic clue from what part of a system the message originated.

Traditionally, under UNIX, there are facilities like KERN (the OS kernel itself), LPD (the line printer daemon) and so on. There are also the LOCAL_0 to LOCAL_7 facilities (see code number 16 from the facility codes below) that are traditionally reserved for administrator and application use.

A device on which syslog is enabled allows you to configure any value as the facility to distinguish between different classes of syslog messages. That is why the facility field is very helpful to define rules that split messages (for example, to different log files based on the facility level).

Refer to the following table for the facility codes and their description:

| Code | Description |
|------|---|
| 0 | kernel messages |
| 1 | user-level messages |
| 2 | mail system |
| 3 | system daemons |
| 4 | security/ authorization messages |
| 5 | messages generated internally by <i>syslogd</i> |

| Code | Description |
|------|--|
| 6 | line printer subsystem |
| 7 | network news subsystem |
| 8 | UUCP subsystem |
| 9 | clock daemon |
| 10 | security/ authorization messages |
| 11 | FTP daemon |
| 12 | NTP subsystem |
| 13 | log audit |
| 14 | log alert |
| 15 | clock daemon |
| 16 | local use 0 reserved for site-specific use |

Severity of Syslog

Codes

The severity is the emergency of the generated messages.

Refer to the following table for the severity codes and their description.

| Code | Description |
|------|---|
| 0 | Emergency: system is unusable. |
| 1 | Alert: action must be taken immediately. |
| 2 | Critical: critical conditions. |
| 3 | Error: error conditions. |
| 4 | Warning: warning conditions. |
| 5 | Notice: normal but significant condition. |
| 6 | Informational: informational messages. |
| 7 | Debug: debug-level messages. |

The Header Field

This field contains the following data:

- A timestamp value for the date and time at which the message was generated
- A value for the hostname (the IP address of the device)

The Message Field

The message field is the third part of the syslog packet. It contains some additional information of the process that generated the message and the text of the message. The message field consists of two parts:

- *tag*

The tag value is the name of the program or process that generated the message.

- *content*

The content value contains the details of the message.

Interface Configuration

The registry string value `HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\Syslog Logmask` decides which Joblog messages are written to syslog.

| Character | Argus Severity | Syslog severity | default |
|-----------|---------------------------------------|---------------------------|---------|
| F | ARGUS_SV_SEVERE_ERROR | LOG_CRIT | X |
| E | ARGUS_SV_ERROR | LOG_ERR | X |
| W | ARGUS_SV_WARNING | LOG_WARNING | |
| I | ARGUS_SV_INFO | LOG_INFO | |
| J | Open Job Messages form all severities | Appropriate LOG_ severity | |

If F, E, W, I or J are part of the `Logmask` value, the Joblog is also written to syslog. All messages are written under `LOG_USER`.

The `syslogd` daemon writes the syslog. When `syslogd` starts up, it reads its configuration file (usually `/etc/syslog.conf`) to determine what kinds of events to log and to what location.

For detailed information about your current configuration file call `man syslog.conf`. It describes the parameter of your `syslog.conf` file.

There is no standard for the message output file and no utility known that reads automatically the right file.

Follow these steps:

1. Open the `/etc/syslog.conf` and find out there messages are written.
2. Open the plain text syslog output file in any editor or run `cat`, `more`, `less`, etc., as user root or with root permissions.
3. Browse to **Software AG**.

Samples of Configuration Files

- [Linux](#)
- [Linux \(Suse 9.1\) Syslog.conf Man Pages](#)
- [Solaris Example](#)

Linux

Suse 9.1 /etc/syslog.conf example:

```
# /etc/syslog.conf - Configuration file for syslogd(8)
#
# For info about the format of this file, see "man syslog.conf".
#
#
# print most on tty10 and on the xconsole pipe
#
kern.warning;*.err;authpriv.none /dev/tty10
kern.warning;*.err;authpriv.none|/dev/xconsole
*.emerg *
# enable this, if you want that root is informed
# immediately, for example, of logins
#*.alert root
#
# all email-messages in one file
#
mail.*-/var/log/mail
mail.info-/var/log/mail.info
mail.warning-/var/log/mail.warn
mail.err /var/log/mail.err
#
# all news-messages
#
# these files are rotated and examined by "news.daily"
news.crit-/var/log/news/news.crit
news.err-/var/log/news/news.err
news.notice-/var/log/news/news.notice
# enable this, if you want to keep all news messages
# in one file
#news.*-/var/log/news.all
#
# Warnings in one file
#
*.=warning;*.=err                -/var/log/warn
```

```
*.crit                                /var/log/warn
#
# save the rest in one file
#
*.*;mail.none;news.none              -/var/log/messages
#
# enable this, if you want to keep all messages
# in one file
#*.*                                  -/var/log/allmessages
#
# Some foreign boot scripts require local7
#
local0,local1.*                       -/var/log/localmessages
local2,local3.*                       -/var/log/localmessages
local4,local5.*                       -/var/log/localmessages
local6,local7.*                       -/var/log/localmessages
```

Critical errors, errors and warnings here are written to */var/log/warn*. The rest are written to */var/log/messages*.

Example:

```
cat /var/log/warn
...
Sep 28 10:24:43 pclinarg1 Software AG - argevsrv[9997]: Test - Warning
Sep 28 10:24:43 pclinarg1 Software AG - argevsrv[9997]: Test - Error
Sep 28 10:24:43 pclinarg1 Software AG - argevsrv[9997]: Test - Crit Error

cat /var/log/messages
...
Sep 28 10:24:43 pclinarg1 Software AG - argevsrv[9997]: Test - Info
Sep 28 10:24:43 pclinarg1 Software AG - argevsrv[9997]: Test - Warning
Sep 28 10:24:43 pclinarg1 Software AG - argevsrv[9997]: Test - Error
Sep 28 10:24:43 pclinarg1 Software AG - argevsrv[9997]: Test - Crit Error
Sep 28 10:54:27 pclinarg1 Software AG - argevsrv[10449]: start argevsrv demon
```

Linux (Suse 9.1) Syslog.conf Man Pages

■ NAME

syslog.conf - *syslogd(8)* configuration file

■ Description

The *syslog.conf* file is the main configuration file for the *syslogd(8)* that logs system messages on UNIX and Linux systems. This file specifies rules for logging. For special features see the *syslogd(8)* manpage.

Every rule consists of two fields, a selector field and an action field. These two fields are separated by one or more spaces or tabs. The selector field specifies a pattern of facilities and priorities belonging to the specified action.

Lines starting with a hash mark (#) and empty lines are ignored.

This release of *syslogd* is able to understand an extended syntax. One rule can be divided into several lines if the leading line is terminated with an backslash (\).

■ Selectors

The selector field itself again consists of two parts, a facility and a priority, separated by a period (.). Both parts are case insensitive and can also be specified as decimal numbers, but don't do that, you have been warned. Both facilities and priorities are described in *syslog(3)*. The names mentioned below correspond to the similar *LOG_-values* in */usr/include/syslog.h*.

The facility is one of the following keywords: 'auth', 'authpriv', 'cron', 'daemon', 'kern', 'lpr', 'mail', 'mark', 'news', 'security' (same as 'auth'), 'syslog', 'user', 'ucp' and 'local0' through 'local7'. The keyword security is not used anymore and mark is only for internal use. Therefore, it is not used in applications. Anyway, if you want to specify and redirect these messages here, you can so. The facility specifies the subsystem that produced the message, that is, all mail programs log with the mail facility (LOG_MAIL) if they log using syslog.

The priority is one of the following keywords, in ascending order: 'debug', 'info', 'notice', 'warning', 'warn' (same as 'warning'), err, error (same as 'err'), 'crit', 'alert', 'emerg', 'panic' (same as 'emerg'). The keywords 'error', 'warn' and 'panic' are deprecated and must not be used anymore. The priority defines the severity of the message

The behavior of the original BSD *syslogd* is that all messages of the specified priority and higher are logged according to the given action. This *syslogd(8)* behaves the same, but has some extensions.

In addition to the above mentioned names the *syslogd(8)* understands the following extensions: An asterisk (*) stands for all facilities or all priorities, depending on where it is used (before or after the period). The keyword none stands for no priority of the given facility.

You can specify multiple facilities with the same priority pattern in one statement using the comma (,) operator. You can specify as much facilities as you want. Remember that only the facility part from such a statement is taken. A priority part is skipped.

Multiple selectors can be specified for a single action using the semicolon (;) separator. Remember that each selector in the selector field is capable to overwrite the preceding ones. Using this behavior you can exclude some priorities from the pattern.

This *syslogd(8)* has a syntax extension to the original BSD source, that makes its use more intuitively. You can precede every priority with an equation sign (=) to specify only this single priority and not any of the above. You can also precede the priority with an exclamation mark (!) to ignore all that priorities, either exact this one or this and any higher priority. If you use both extensions than the exclamation mark must occur before the equation sign, just use it intuitively.

■ Actions

The action field of a rule describes the abstract term *logfile''*. A *logfile''* need not to be a real file. The *syslogd(8)* provides the following actions.

■ Regular File

Typically messages are logged to real files. The file has to be specified with full pathname, beginning with a slash */*.

You can prefix each entry with the minus *-* sign to omit syncing the file after every logging. Note that you may lose information if the system crashes right behind a write attempt. Nevertheless, this may give you back some performance, especially if you run programs that use logging in a very verbose manner.

■ Named Pipes

This version of *syslogd(8)* has support for logging output to named pipes (fifos). A fifo or named pipe can be used as a destination for log messages by prepending a pipe symbol (*|*) to the name of the file. This is handy for debugging. Note that the fifo must be created with the *mkfifo(1)* command before *syslogd(8)* is started.

■ Terminal and Console

If the file you specified is a *tty*, special *tty*-handling is done, same with */dev/console*.

■ Remote Machine

This *syslogd(8)* provides full remote logging, that is, it is able to send messages to a remote host running *syslogd(8)* and to receive messages from remote hosts. The remote host does not forward the message again, but just logs them locally. To forward messages to another host, use the hostname with the at sign (*@*).

With this feature you can control all syslog messages on one host, if all other machines log remotely to that host.

■ List of Users

Usually, critical messages are also directed to *root''* on that machine. You can specify a list of users that shall get the message by simply writing the login. You can specify more than one user by separating them with commas (*,*). If they are logged in they get the message.

■ Everyone logged on

Emergency messages often go to all users currently online to notify them that something strange is happening with the system. To specify this *wall(1)*-feature use an asterisk (***).

■ Examples

Here are some example, partially taken from a real existing site and configuration.


```
# Store critical stuff in critical
#
*.=crit;kern.none          /var/adm/critical

That stores all messages with the priority crit in the file /var/adm/critical,
except for any kernel message.

# Kernel messages are first, stored in the kernel
# file, critical messages and higher ones also go
# to another host and to the console
#
kern.*                    /var/adm/kernel
kern.crit                 @finlandia
kern.crit                 /dev/console
kern.info;kern.!err      /var/adm/kernel-info
```

The first rule direct any message that has the kernel facility to the file */var/adm/kernel*.

The second statement directs all kernel messages of the priority crit and higher to the remote host finlandia. This is useful, because if the host crashes and the disks get irreparable errors, you may not be able to read the stored messages. If they are on a remote host, too, you can still try to find out the reason for the crash.

The third rule directs these messages to the actual console, so the person who works on the machine can get them, too.

The fourth line tells the *syslogd* to save all kernel messages that come with priorities from info up to warning in the file */var/adm/kernel-info*. Everything from *err* and higher is excluded.

```
# The tcp wrapper loggs with mail.info, we display
# all the connections on tty12
#
mail.=info                /dev/tty12
```

This directs all messages that uses *mail.info* (in source *LOG_MAIL* | *LOG_INFO*) to */dev/tty12*, the 12th console. For example the TCP wrapper *tcpd(8)* uses this as it is default.

```
# Store all mail concerning stuff in a file
#
mail.*;mail.!=info       /var/adm/mail
```

This pattern matches all messages that come with the mail facility, except for the info priority. These are stored in the file */var/adm/mail*.

```
# Log all mail.info and news.info messages to info
#
mail,news.=info          /var/adm/info
```

This command extracts all messages that come (either with *mail.info* or with *news.info*) and stores them in the file */var/adm/info*.

```
# Log info and notice messages to messages file
#
*.=info;*.=notice;\
    mail.none /var/log/messages
```

This lets the *syslogd* log all messages that come with either the info or the notice facility into the file */var/log/messages*, except for all messages that use the mail facility.

```
# Log info messages to messages file
#
*.=info;\
    mail,news.none      /var/log/messages
```

This statement causes the *syslogd* to log all messages that come with the info priority to the file */var/log/messages*. In this case, messages coming with the mail or the news facility are not stored.

```
# Emergency messages are displayed using wall
#
*.=emerg                  *
```

This rule tells the *syslogd* to write all emergency messages to all currently logged on users. This is the wall action.

```
# Messages of the priority alert is directed
# to the operator
#
*.alert                   root,joe
```

This rule directs all messages with a priority of alert or higher to the terminals of the operator, that is, of the users root" and joe" if they're logged on.

```
*.*                       @finlandia
```

This rule redirects all messages to a remote host called finlandia. This is useful especially in a cluster of machines where all syslog messages are stored on one machine only.

■ Configuration File Syntax Differences

Syslogd uses a slightly different syntax for its configuration file than the original BSD sources. Originally all messages of a specific priority and above were forwarded to the log file. The modifiers '=', '!', and '-' were added to make the *syslogd* more flexible and to use it in a more intuitive manner.

The original BSD *syslogd* does not understand spaces as separators between the selector and the action field.

```
FILES
/etc/syslog.conf
    Configuration file for syslogd
```

■ BUGS

The effects of multiple selectors are sometimes not intuitive. For example *mail.crit, *.err* ' selects mail" facility messages at the level of err" or higher, not at the level of crit" or higher.

Solaris Example

Sun Solaris 8 64 */etc/syslog.conf* example:

```
#ident  "@(#)syslog.conf      1.5      98/12/14 SMI"      /* SunOS 5.0 */
#
# Copyright (c) 1991-1998 by Sun Microsystems, Inc.
# All rights reserved.
#
# syslog configuration file.
#
# This file is processed by m4 so be careful to quote (`') names
# that match m4 reserved words. Also, within ifdef's, arguments
# containing commas must be quoted.
#
*.err;kern.notice;auth.notice      /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit      /var/adm/messages

*.alert;kern.err;daemon.err      operator
*.alert      root

*.emerg      *

# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
#auth.notice      ifdef(`LOGHOST', /var/log/authlog, @loghost)

mail.debug      ifdef(`LOGHOST', /var/log/syslog, @loghost)

#
# non-loghost machines uses the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(`LOGHOST', ,
user.err      /dev/sysmsg
user.err      /var/adm/messages
user.alert      `root, operator'
user.emerg      *
)
```


11 System Management Hub Logging Service

- Event Layer Logging 72
- Changing the Database Path 73

The information is organized under the following headings:

Event Layer Logging

System Management Hub offers a product logging facility. The Event Layer service (daemon) is designed to uniform the manner in which events are logged for all products.

The Event Layer service uses the TCP protocol and a database to store events. The location of the Event Layer service database (*jobs.db*) on your local machine is:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\EvtSql]
"SqlDb_Name"="jobs.db"
"SqlDb_Path"="C:\<Software AG_directory>\InstanceManager\log\events"
```

The location of the Event Layer logging folder is shown on the following screen capture.

All the files outside the events folder (for example, *argsrv.log*) are internal system trace files. They keep a record of actions taken within System Management Hub.

This internal system trace is disabled by default. When you enable it (see below the *trace* key from the registry), you are able to trace those changes with the system logging, too.

The registry keys for the Event Layer are in the System Management Hub node.

Following are the default System Management Hub settings of the Event Layer registry keys:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer]
"Tcp_Ip_Port"="10014"
"Tcp_Ip_Queue_Size"="100"
"Tcp_Ip_Recv_Timeout"="60"
"Logging_Path"="C:\<Software AG_directory>\InstanceManager\log\events"
"Trace"="0"
"ShutdownTimeout"="900"
"Config_Edit"="1"
"ShowDebug"="0"
"RefreshRate"="5"
"Snmp"="1"
"Snmp_Interface"="C:\<Software AG_directory>\InstanceManager\bin\argevsnp.dll"
"Snmp_Port"="10017"

[HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\Cleanup]
"Info"="30"
"Warning"="30"
"Error"="30"
"Fatal"="30"

[HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\EvtSql]
```

```

"SqlDb_Name"="jobs.db"
"SqlDb_Path"="C:\<Software AG_directory>\InstanceManager\log\events"
"Max_Jobs_In_Memory"="200"
"Logging_On"="0"
"Sql_On"="1"
"Restore"="0"
"SqlDb_Save_Old"="0"
"SqlDb_Cache_size"="2000"

[HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\Syslog]
"Logmask"="FE"

```

Refer to the following table for a list of the most important registry keys and their values.

| Registry Key | Values |
|--------------|---|
| SNMP | enable ("1") disable ("0") |
| Trace | enable ("1") disable ("0") |
| Logging_Path | The path to the SQL database storage of all log messages for products |

Changing the Database Path

There are user scenarios in which you may want to change the database path of the Event Layer service.

► To change the event layer database path

- 1 Stop System Management Event Layer service.
- 2 Remember the value of registry `HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\EvtSql SqlDb_Path` and the location of the folder on your local machine (for example, `C:\<Software AG_directory>\InstanceManager\log\events`).
- 3 Change the value for database path in registry: `HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\EventLayer\EvtSql SqlDb_Path` to the new location (for example, `C:\<basedir>`).
- 4 Copy or move the file from the directory in step 2 to the new location directory that is the value of `SqlDb_Path`.
- 5 Start System Management Event Layer service.

