M-Guard Administration Guide

Table of Contents

List of Figures

List of Tables

Notice

The M-Guard product uses protocols and algorithms to encrypt network traffic and hence is subject to UK Export Controls. You must ensure you comply with these controls. All Isode software is subject to a license agreement and your attention is also called to the export terms of your Isode license.

Preface

Readership

This guide is intended for administrators of M-Guard product components, namely the M-Guard Appliance and its Guard software component and M-Guard Console, a desktop tool for managing the appliances and Guard instances.

Versions

This document is for M-Guard 1.3 covering the latest release (1.3.3) of M-Guard Console and the current release (1.3.3) of M-Guard Appliance, both at the time of publication. This document is generally pertinent to subsequent update releases of M-Guard Console and M-Guard Appliance.

Note that M-Guard Console and M-Guard Appliance have are independent version identifiers as they are not necessarily updated at the same time. They share the same form, M.N.U where M is the major version number, N is the minor version number, and U is the update number. The product version number is of the form M.N where M and N are aligned with the major and minor versions numbers of M-Guard Console.

Glossary of Product Terms

This section provides defines a few terms used across the M-Guard product family and its documentation..

Glossary of Product Terms

M-Guard
: The term M-Guard names the Isode *border guard* family of products as well as the *XML guard* product itself.

Appliance
: A physical or virtual appliance providing the Guard and/or other software components. The term may be qualified with a term naming the software components available on the Appliance, as in "A Guard Appliance can run multiple instances of the Guard software."

Guard
: refers to the *XML guard* software component of an Appliance.

instance
: refers to an instantiation of a particular software component and is normally qualified the term naming that component, as in "a Guard instance"

Console
: refers to the desktop tool used to manage appliances and the software components available on them, typically written M-Guard Console.

GCXP
: stands for the *Guard Content Exchange Protocol*. GCXP is used to transfer content, typically in XML format, from one application server to another through a series of guard or similar devices.

GCXP Producer
: refers to the desktop tool when running as a GCXP content producer

GCXP Consumer
: refers to the desktop tool when running as a GCXP content consumer

Typographical Conventions

The text of this manual uses different typefaces to identify different types of objects, such as file names and input to the system. The typeface conventions are shown in the table below.

Object	**Example**
File and folder names	`/etc`
User input	**`example.com`**
User menu item selection	File | Open...
Cross references	see the section called “Typographical Conventions”

Documentation Availability

Current M-Guard documentation is available from the Isode web site at http://www.isode.com/support/docs.html.

M-Guard Console includes the *M-Guard Administration Guide* and has an export to PDF facility.

Support Queries and Bug Reporting

A number of email addresses are available for contacting Isode. Please use the address relevant to the content of your message.

- For all account-related inquiries and issues: `<customer-service@isode.com>`. If customers are unsure of which list to use then they should send to this list. The list is monitored daily, and all messages will be responded to.

- For all licensing related issues: `<license@isode.com>`.

- For all technical inquiries and problem reports, including documentation issues from customers with support contracts: `<support@isode.com>`. Customers should include relevant contact details in initial calls to speed processing. Messages which are continuations of an existing call should include the call ID in the subject line. Customers without support contracts should not use this address.

- For all sales inquiries and similar communication: `<sales@isode.com>`.

Bug reports on software releases are welcomed. These may be sent by any means, but electronic mail to the support address listed above is preferred. Please send proposed fixes with the reports if possible. Any reports will be acknowledged, but further action is not guaranteed. Any changes resulting from bug reports may be included in future releases.

Chapter 1. Overview

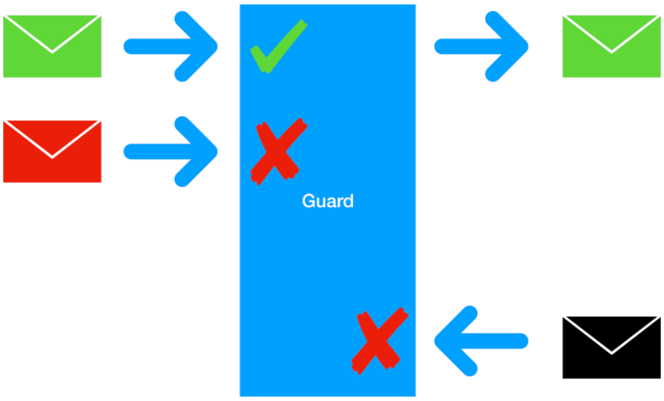
Abstract

The M-Guard product provides an *XML guard* for checking XML content exchanged across network boundaries. This chapter provides an overview to various components of the M-Guard product, which includes the M-Guard Appliance which runs Guard instances and the M-Guard Console management application.

Guard Overview

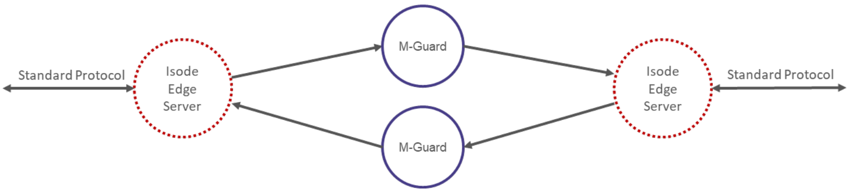
Each Guard instance functions as an *application-level data-diode* that validates XML content passing through it. Only XML content that conforms to a set of rules is passed through; other content is rejected. The following figure provides a functional overview of how the Guard treats valid and invalid content.

Figure 1.1. Guard Functional Overview



In an operational deployment in normal conditions, it is expected that all content providers will ensure that all content they provide adheres to the content rules and hence the Guard will accept and forward all content. When the Guard rejects content in such deployments, it indicates improper configuration or function of either the content provider, the Guard, or security failures which require immediate attention of the system/network administrators.

Figure 1.2. Example Edge Configuration



The Edge configuration shown in the above figure shows bidirectional content flow with each flow being checked by a Guard instance. The Guard may be used in other configurations and may interface to non-Isode servers. In the above Edge configuration, the Isode servers will convert a standard application protocol, such as SMTP or XMPP, for exchange through the Guard instances.

For bidirectional communication, as shown above, a pair of Guard instances will be used, one for each direction of data flow.

What is an XML Guard?

The term *XML guard* is used in a few places but is relatively specialized. The term *guard* by itself is used to describe a device which allows systems on otherwise separate networks to communicate subject to some set of restrictions. In some ways, a guard is much like a *network firewall* but instead of operating at the lower layers, such as the Internet layer of the protocol stack, guards tend to operate at higher levels, namely the Application layer. A guard can be thought of as an *application-level firewall*.

Table 1.1. Internet 5-Layer Model

Internet Layers	Examples
Application	XMPP, SMTP, LDAP
Transport	TCP, UDP
Internet	IP
Link	Ethernet, X.25, IPoAC
Physical	coaxial cable, optical fiber, pigeons

The term *XML guard* hence refers to specialized guards which act upon XML content being exchanged between two otherwise separate systems.

A related term is *XML Appliance* which Wikipedia [https://en.wikipedia.org] describes as “a special-purpose network device used to secure, manage and mediate XML traffic”. However, such devices perform a number of functions considered beyond the scope of guarding or application-level firewalling, including transformation of XML content and routing of XML content based upon the XML content. This is especially the case in securing communications between networks operating at different security levels or autonomous networks operating at equivalent security levels.

Isode describes the Guard as an *XML guard* which is intended to be used to ensure XML content exchanged between two separate networks adheres to the the set of restrictions imposed by policy upon the content exchange. The Guard does not provide any XML content transformation functions or routing capabilities. It is a pure guard.

A guard, like a network firewall, often supports bidirectional content exchange between the two separate networks. When restricted (by configuration or implementation) to unidirectional operation, it is sometimes referred to as a *data diode*. Guard is designed so that each Guard instance only exchanges application data (the XML content) in one direction, so it can be thought of as an *application-level data diode*. But Guard is not a *physical-layer data diode* which the term *data diode* typically refers to. *Physical-layer data diodes* physically ensure raw data can be exchanged in only a single direction, such as by having two devices connect by a single fiber optic cable with only one device having a laser to send data and the other only having a light sensor to receive data.

Modern *physical-layer data-diode* products are sometimes described as *unidirectional gateways*. This is to highlight that these products support application integration using a wider range of application interfaces than simple *data diode* For instance, a simple *data diode* might only offer a UDP interface for integration, whereas it might support *application-level proxying* for multiple application protocols (e.g., SMTP, HTTPS).

The Guard is designed to support exchanging traffic through *physical-layer data diodes* and *unidirectional gateways*.

Guard Content Exchange Protocol (GCXP)

The Guard communicates using *Guard Content Exchange Protocol (GCXP)* as a secure protocol for communicating XML content between Guard instances and processes on either side.

Many *XML Appliances* use proprietary mechanisms or protocols such as SOAP to communicate. Isode was not able to identify any existing standard protocol for communicating with an *XML Guard*.

GCXP is offered by Isode as an open specification. GCXP utilizes *Concise Binary Object Representation* (CBOR), as specified in [RFC 7049], for framing of XML content. GCXP utilizes *Transport Layer Security* (TLS) for strong mutual authentication, data integrity, and confidentiality protection.

Isode provides *open source* C++ implementations of GCXP and CBOR, as well as example consumer and producer programs. These can be used to develop applications to communicate with Guard or to enable applications using GCXP to communicate with other guards which support GCXP. These implementations are available at https://bitbucket.org/isode/.

GCXP is simple and CBOR implementations are available for a wide range of programming languages, which allows for GCXP to be implemented for applications which are not written in C++.

For more information, see Appendix B, *Guard Content Exchange Protocol (GCXP)*

Content Restrictions

The Guard verifies that the application content to be exchanged adheres to a set of restrictions imposed by policy. The set of restrictions can be expressed using a variety of standard XML technologies. This choice of technology allows rules to be specified in a manner appropriate to the rules. The Guard may be used with:

- XML Schema Definition [https://www.w3.org/standards/xml/schema] (XSD),
- RELAX NG [https://relaxng.org] (REgular LAnguage for XML Next Generation),
- Schematron [http://schematron.com],
- XML Stylesheets [http://www.w3.org/TR/xml-stylesheet] (XSLT),
- XML Canonicalization [http://www.w3.org/TR/xml-c14n11/] (XC14N),
- Unicode Normalization [https://unicode.org/reports/tr15/]

This choice of technology allows rules to be specified in a manner appropriate to the application information to be exchanged.

Rate Limiting

The Guard can also limit the rate in which messages are forwarded. When rate limiting is enabled, the Guard will not commence processing of a message until a specified minimum interval of time has elasped since processing of the previous message commenced.

Applications

The Guard is a *generic* XML guard. It is not tied to any particular application.

Applications integrating with the Guard are expected to represent the particular information they wish to exchange into XML for transfer using GCXP. For instance, to exchange an email message between two systems, typically the content provider would convert the email message from a standard email format, such as Internet message format, to an XML format. The XML would then be transferred using GCXP to the other system, which in turn would convert the XML to the desired email format.

GCXP application profiles specify how GCXP is to be used to transfer information through guards and, in particular, how application information is represented and encoded for exchange. A GCXP application profile typically specifies the XML schema the information is to be represented in and any XML and Unicode transforms that expected to be applied normalize the XML. Typically the transformations include XML canonicalization and Unicode normalization as well as application-specific transformations. GCXP application profiles also specifies how application content can flow within a GCXP stream.

Isode has four Guard applications in development:

- *Instant Messaging*: using the M-Link Edge product to interface with the Guard using GCXP. This will enable boundary and cross-domain XMPP services.
- *Messaging*: M-Switch to enable cross-domain messaging checks for SMTP, [STANAG 4406] and [ACP 127] messaging. M-Switch will convert each of the messaging formats used in the messaging systems to, and from, an XML messaging format.
- *[STANAG 5066] Crypto Bypass*: Icon-5066 (red side) will pass modem and ALE monitoring and control through the Guard to provide securely red/black separation.
- *Red/Black control and management*: Red/Black will enable red side monitoring and control of black side devices, with separation provided by the Guard.

GCXP application profiles are intended to be policy neutral, leaving policy enforcement to additional content rules. M-Guard supports content rule catalogs which provide various rules a deployment can enable and customize. Each content rule catalog is typically designed to be used in conjunction with a particular GCXP profile. For instance, the DemoP Base Rules catalog is intended to be used in conjunction with the DemoP GCXP Application profile.

Guard Appliance

Guard instances run on a Guard Appliance. A single appliance may run one or more Guard instances. Running multiple Guard instances on one appliance may be helpful to minimize the number of appliances. Running each Guard instance on a separate appliance may be desirable for security separation.

The Guard Appliance is a physical or virtual device. The device utilizes the NanoBSD [https://www.freebsd.org/doc/en_US.ISO8859-1/articles/nanobsd/] variant of the FreeBSD operating system. NanoBSD is specifically designed for appliances. A key feature of NanoBSD is that the operating system and included software is read-only at run-time. The Appliance supports FreeBSD network security features, including host-level IP firewalls. The Appliance and the Guard instances utilizes syslog [https://en.wikipedia.org/wiki/Syslog] for logging of events.

M-Guard Console is used to manage Appliance and the Guard instances running on it. The Appliance also offers command line management through Secure Shell [https://en.wikipedia.org/wiki/Secure_Shell] (SSH) access.

Guard Appliances have multiple network adapters allowing for network separation. Typically three network adapters are used: one for each of the two - otherwise separate - networks which content is being exchanged between, and a third network for management and monitoring. Additional network adapters may be used as might be appropriate in the operational environment.

The Guard Appliance supports Network Time Protocol (NTP). This allows the appliance to synchronize its clock with one or more NTP servers, typically operating on the management network. The appliance may also provide time references to one or more networks.

Physical Appliance Platforms

The physical Appliance is certified for the Netgate SG-5100 [https://docs.netgate.com/manuals/pfsense/en/latest/sg-5100-security-gateway-manual.pdf] hardware system, an Intel Atom-based device designed for network security devices. The device has six network interfaces, each with Auto-

MDIX support, allowing systems to be connected to it directly (no Ethernet hub/switch needed). It is expected that Isode will certify additional platforms in the future based upon customer inputs.

Virtual Appliance Platforms

The following *hypervisors* are supported:

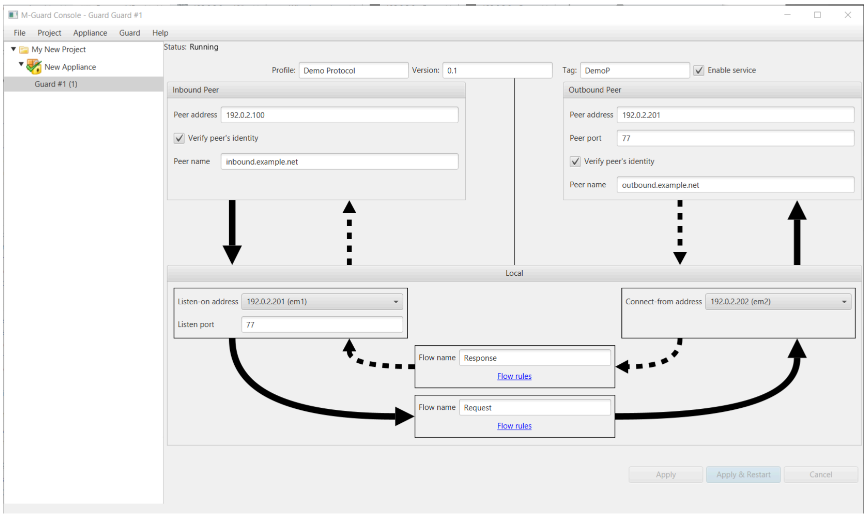
- Microsoft Hyper-V
- Oracle VirtualBox

The virtual appliances are configured with four network adapters, each of which can be associated with an appropriate virtual or physical network using the hypervisor's network configuration tools.

Future releases of Appliance may also support VMWare hypervisors.

M-Guard Console

Figure 1.3. Example of M-Guard Console



Status: Running

Profile: Demo Protocol
Version: 0.1
Tag: DemoP
☑ Enable service

Inbound Peer
Peer address 192.0.2.100
☑ Verify peer's identity
Peer name inbound.example.net

Outbound Peer
Peer address 192.0.2.201
Peer port 77
☑ Verify peer's identity
Peer name outbound.example.net

Local
Listen-on address 192.0.2.201 (em1)
Listen port 77
Connect-from address 192.0.2.202 (em2)
Flow name Response
Flow name Request

M-Guard Console is a pure Java desktop application for managing appliances and the Guard instances run on them. It will typically connect to an appliance using a management network, independent of the networks on either side of the appliance. The Console communicates securely with an appliance using Secure Shell (SSH).

Key features shown in the screen shot.

- At the top level, there are one or more Projects, each of which governs settings for a set of appliances, each containing a set of Guard instances.

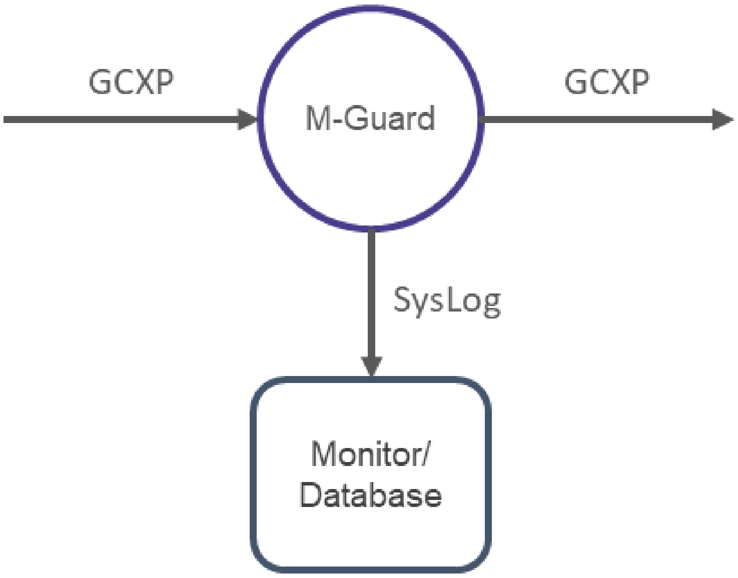
  In the screenshot above, there is one Project named `My New Project` with one appliance named `New Appliance`with one Guard instance named `Guard #1`.

- Settings for each Guard instance are configured by the Console.
- There is an application-data content flow of data from left to right.
- There is an *optional* flow of control data from right to left. Generally, this flow is either disallowed or only signals (no application data) to the producer that the content has been transferred to the consumer or has been rejected.
- Each flow must adhere to a specified set of rules which can be configured with M-Guard Console.

The Console has numerous other capabilities, including facilities which can be used to check a Guard instance's handling of user-provided XML content.

Syslog Monitoring

Figure 1.4. Syslog Monitoring

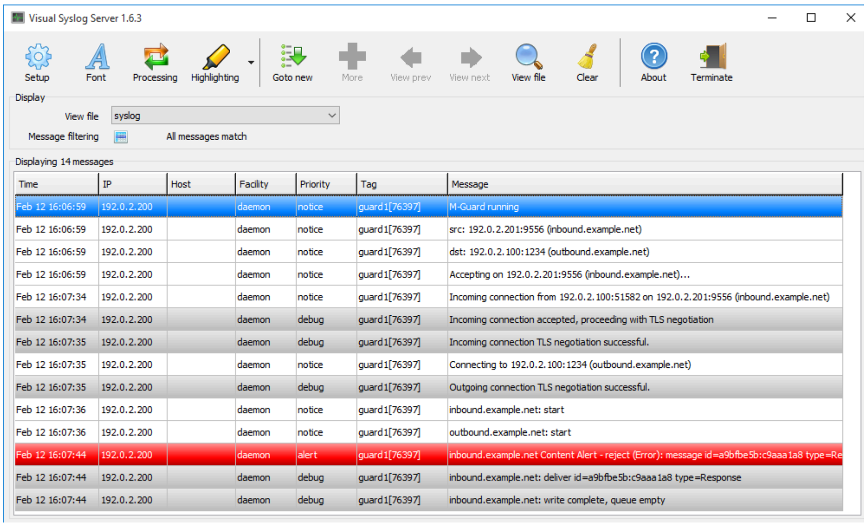


The Guard Appliance and Guard instances report events using syslog. The appliance can be configured to send events to one or more syslog servers on the management network. The appliance supports traditional syslog over UDP as well as syslog over TCP, syslog over TLS, Reliable Event Logging Protocol (RELP) over TCP, RELP over TLS and SNMP trap generation. Use of RELP over TLS is recommended for its reliability and security properties.

The Guard reports events for GCXP activity including when content is accepted or rejected. Guard events include an appropriate severity level.

When desirable, a syslog server can be installed on the same desktop as the Console. The figure below shows events as presented by the Visual Syslog tool. The Guard generates 'Content Alert - reject' events at the *alert* severity level, which this tool shows in red.

Figure 1.5. Visual Syslog Server



Display

View file: syslog

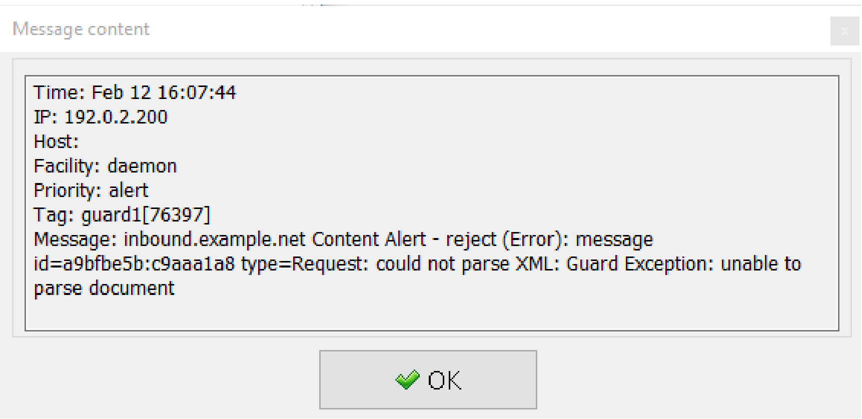
Message filtering: All messages match

Displaying 14 messages

Time	IP	Host	Facility	Priority	Tag	Message
Feb 12 16:06:59	192.0.2.200		daemon	notice	guard1[76397]	M-Guard running
Feb 12 16:06:59	192.0.2.200		daemon	notice	guard1[76397]	src: 192.0.2.201:9556 (inbound.example.net)
Feb 12 16:06:59	192.0.2.200		daemon	notice	guard1[76397]	dst: 192.0.2.100:1234 (outbound.example.net)
Feb 12 16:06:59	192.0.2.200		daemon	notice	guard1[76397]	Accepting on 192.0.2.201:9556 (inbound.example.net)...
Feb 12 16:07:34	192.0.2.200		daemon	notice	guard1[76397]	Incoming connection from 192.0.2.100:51582 on 192.0.2.201:9556 (inbound.example.net)
Feb 12 16:07:34	192.0.2.200		daemon	debug	guard1[76397]	Incoming connection accepted, proceeding with TLS negotiation
Feb 12 16:07:35	192.0.2.200		daemon	debug	guard1[76397]	Incoming connection TLS negotiation successful.
Feb 12 16:07:35	192.0.2.200		daemon	notice	guard1[76397]	Connecting to 192.0.2.100:1234 (outbound.example.net)
Feb 12 16:07:35	192.0.2.200		daemon	debug	guard1[76397]	Outgoing connection TLS negotiation successful.
Feb 12 16:07:36	192.0.2.200		daemon	notice	guard1[76397]	inbound.example.net: start
Feb 12 16:07:36	192.0.2.200		daemon	notice	guard1[76397]	outbound.example.net: start
Feb 12 16:07:44	192.0.2.200		daemon	alert	guard1[76397]	inbound.example.net Content Alert - reject (Error): message id=a9bfbe5b:c9aaa1a8 type=Re
Feb 12 16:07:44	192.0.2.200		daemon	debug	guard1[76397]	inbound.example.net: deliver id=a9bfbe5b:c9aaa1a8 type=Response
Feb 12 16:07:44	192.0.2.200		daemon	debug	guard1[76397]	inbound.example.net: write complete, queue empty

The details of the syslog event indicate why the XML content was rejected by the Guard.

Figure 1.6. M-Guard Rejection Event



```
Time: Feb 12 16:07:44
IP: 192.0.2.200
Host:
Facility: daemon
Priority: alert
Tag: guard1[76397]
Message: inbound.example.net Content Alert - reject (Error): message
id=a9bfbe5b:c9aaa1a8 type=Request: could not parse XML: Guard Exception: unable to
parse document
```

The syslog events can also be sent to more sophisticated management/monitoring systems, which is expected to be the approach for most operational deployments.

Product Versioning and Distribution

The M-Guard product consists as two top-level components, the M-Guard Appliance and the M-Guard Console. These are separately packaged and independently versioned. Both however have version numbers of the form *M.N.U* where M is the *major* version number, N is a *minor* version number, and U is an *update* number.

Customers should generally use the latest version of the Appliance and the latest version of the M-Guard Console.

M-Guard Appliance is distributed as a set of platform-specific full system images and a cross-platform 'update' system image. Presently one physical machine is supported, the Netgate SG-5100 [https://docs.netgate.com/manuals/pfsense/en/latest/sg-5100-security-gateway-manual.pdf] appliance, and two virtual machine hypervisors are supported, Microsoft Hyper-V and Oracle VirtualBox hypervisors. The following table shows the naming convention of full and 'update' system images. Note that M, N, and U are placeholders for the *major*, *minor*, and *update* numbers as discussed above.

Table 1.2. Naming of M-Guard Appliance system images

Platform	Full Image	Update Image
Netgate SG-5100 appliance	*M-Guard-M.N.U-multi-full.img.xz*	Same for all platforms: *M-Guard-M.N.U-multi-update.img.xz*
Microsoft Hyper-V hypervisor	*M-Guard-M.N.U-hyperv-full.zip*	
Oracle VirtualBox hypervisor	*M-Guard-M.N.U-vbox-full.ova*	

A file containing SHA256 message digests (hashes) of each M-Guard Appliance release artifact is available and has a name of the form *M-Guard-M.N.U.digest*.

M-Guard Console is distributed as a signed Java Archive (.jar) file and has a name of the form *M-Guard-Console-M.N.U.jar*.

M-Guard Console Profiles, containing example profiles, is distributed as a signed Java Archive (.jar) file and has a name of the form *M-Guard-Console-M.N.U-Profiles.jar*.

A file containing SHA256 message digests (hashes) of each M-Guard Console release artifact is available and has a name of the form `M-Guard-Console-M.N.U.digest`.

Detached digital signatures are available for each M-Guard Appliance and M-Guard Console release artifact and each digest file. The signature for a particular release artifact has the name of the form `artifact-file-name.sig` where artifact-file-name is the file name of the release artifact (as described above).

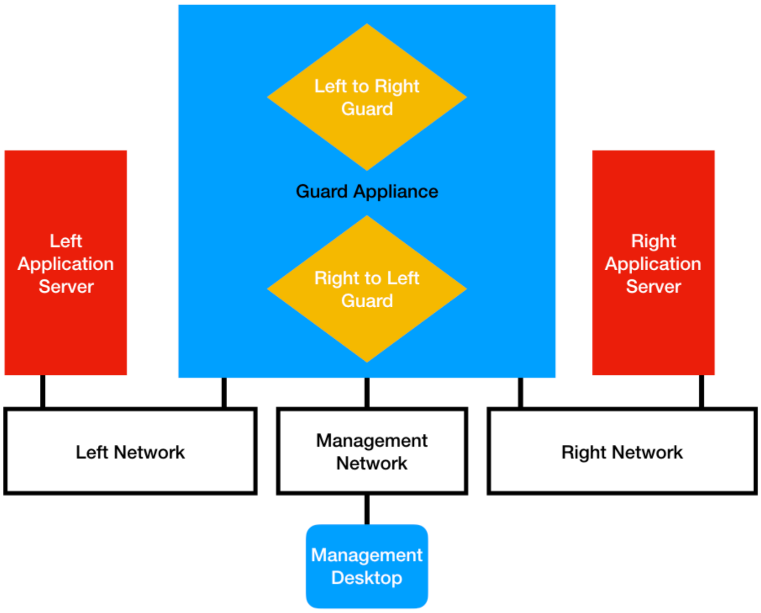
Chapter 2. Getting Started

Abstract

This chapter discusses the initial setup of a Guard Appliance using M-Guard Console and subsequent setup of the appliance to host two Guard instances. It describes the practical steps to accomplish this task.

When setting up a Guard Appliance for the first time, it may be more convenient to use the *M-Guard Evaluation Guide*, which provides a simple step-by-step guide to evaluating M-Guard using a virtual appliance. See the section called “Documentation Availability”.

This chapter discusses the initial setup of a Guard Appliance using M-Guard Console from installing both the Appliance and the Console, to using the Console to configure network settings. As discussed in Chapter 1, *Overview*, the appliance will host one or two Guard instances to exchange application data between two otherwise separate networks, depending on whether the application content was being exchanged unidirectionally or bidirectionally. A third network is typically used for management functions. The following figure provides a network diagram for this scenario.



In this scenario, the Guard Appliance will use three network interfaces, one attached to the left network, one to the right network and one to the *management network*. This chapter discusses the steps necessary to install and setup a Guard Appliance and M-Guard Console running on a *Management Desktop* in this scenario. Discussion of how to configure Guard instances is covered in the section called “Guard: Getting Started”.

You might want to plan your network setup. The following form can be used for this.

Table 2.1. Configuration Plan Form

	Left	**Management**	**Right**
Network Name			
Appliance Name			
Appliance Ethernet Port			
Appliance Interface (original)			

	Left	**Management**	**Right**
Appliance Interface (rename) optional			
Appliance IP Address / prefixlen			
Management Desktop Name	n/a		n/a
Management Desktop IP Address / prefixlen	n/a		n/a
Application Server Name		n/a	
Application Server IP Address / prefix		n/a	

For instance, one might have:

Table 2.2. Example Configuration Plan

	Left	**Management**	**Right**
Network Name	Red	Green	Black
Appliance Name	guard.example.net		
Appliance Ethernet Port	Network Adapter 2	Network Adapter 1	Network Adapter 3
Appliance Interface (original)	em1	em0	em2
Appliance Interface (rename) optional	red	green	black
Appliance IP Address / prefixlen	198.51.100.8/24	192.0.2.1/24	2001:db8::1/32
Management Desktop Name	n/a	desktop.example.net	n/a
Management Desktop IP Address / prefixlen	n/a	192.0.2.2/24	n/a
Application Server Name	red.example.net	n/a	black.example.net
Application Server IP Address / prefix	198.51.100.2/24	n/a	2001:db8::2/32

Note that during the initial setup of the appliance, link-local IP addressing is used until the appliance's management interface can be configured to use the desired IP address. Either IPv4 or IPv6 link-local addressing may be used during initial setup.

IPv6 link-local addresses are typically available whenever the network interface is configured for IPv6 use. IPv6 link-local addresses are derived from the adapter's media access control (MAC) address and hence are consistent across reboots.

IPv4 link-local addresses (sometimes referred to as *self-assigned* addresses) are typically only available when the network interface is configured to use DHCP and no DHCP server is available. IPv4 link-local addresses are randomly generated and hence change upon reboot. If you choose to use IPv4 link-local address, you must ensure that no DHCP server is operating on the management network and avoid unnecessary rebooting of the appliance until initial setup of the appliance, including IP address renumbering, has been completed.

In either case, it is recommended that you renumber the appliance to a static IP address.

Appliance Installation

The Guard Appliance is distributed as full appliance system images, one for each supported physical and virtual platform. The following platforms are supported:

- Netgate SG-5100 appliance
- Microsoft Hyper-V hypervisor
- Oracle VirtualBox hypervisor

You must have one of these platforms to install the Guard Appliance system image onto. You also must have access to the appliance console during its initial setup.

This section describes the steps to create a Guard Appliance using the appropriate *full* system image and readying it for initial setup as discussed below.

Note

Isode recommends evaluators use the Microsoft Hyper-V platform for ease of setup, ease of management, and compatibility reasons.

Microsoft Hyper-V

It is assumed that you have Hyper-V installed on an appropriate Windows system.

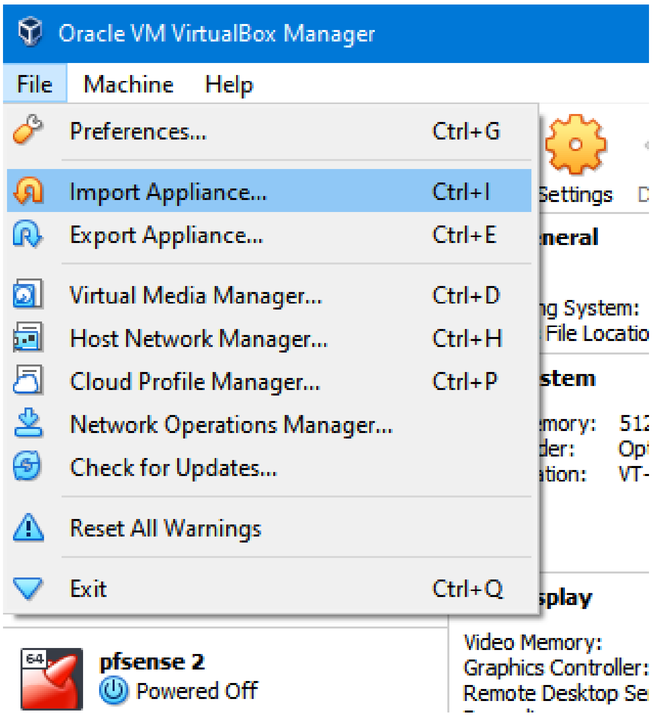
1. Extract the `M-Guard Appliance` folder from the `M-Guard-`*`M.N.U`*`-hyperv-full.zip` file. This folder contains an export of the M-Guard Appliance virtual machine.
2. The virtual machine is pre-configured to have its management interface, `hn0`, attached to a virtual switch named **`M-Guard Management`**. Create a virtual switch with this name. If your virtual machine host system also services as the management desktop, you can use an *Internal* virtual network. If the management desktop is a virtual machine guest on the same virtual machine host, you can use a *Private* network. Otherwise configure a management network which both the Appliance and M-Guard Console are connected to. Configure the management desktop for link-local addressing.
3. Import the virtual machine. Use the import type appropriate for your Hyper-V deployment.

Proceed to the the section called “First Boot of the Appliance” section below.

Oracle VirtualBox

It is assumed that you have the latest version of Oracle VirtualBox installed.

Figure 2.1. Import the Appliance full system image into VirtualBox



1. Create a Guard Appliance by using the VirtualBox File | Import Appliance... to import the latest *`M-Guard-M.N.U-vbox-full.ova`* full system image.

2. Use VirtualBox network configuration tools to connect the first network adapter to your management network. This adapter corresponds to the `em0` interface within the appliance and when using M-Guard Console. If your virtual machine host system also serves as the management desktop, you can use a host-only virtual network. If the management desktop is a virtual machine guest on the same virtual machine host, you can use an internal network. Otherwise configure a management network which both the Appliance and M-Guard Console are connected to. Configure the management desktop for link-local addressing.

You are now ready to turn on the Appliance. Proceed to the the section called “First Boot of the Appliance” section below.

Netgate SG-5100

If you wish to use a Netgate SG-5100 as an M-Guard Appliance, contact Isode for installation instructions. See the section called “Support Queries and Bug Reporting”.

The port labeled `IGB0` will be used for management. This port corresponds to the `igb0` interface within the appliance and when using M-Guard Console. Connect your management desktop directly to this port with an ethernet cable and configure the management desktop for link-local addressing.

Proceed to the the section called “First Boot of the Appliance” section below.

First Boot of the Appliance

With the appliance connected to your management network and only your management network, you are now ready to power-on the appliance. Power it on now.

On first boot, the appliance will attempt to configure itself sufficiently so that you can use M-Guard Console to complete the appliance setup. This includes:

- setting a random password for appliance administration,

- enabling IP with IPv4 and IPv6 *link-local* addressing on the management interface for remote administration, and
- generating SSH keys for remote administration.

The appliance will print information upon completion of the boot process, just before the `login:` prompt. The following is an example:

```
SSH host fingerprints:
    SHA256:vSa0IEGF5kLtTIi1fvOH4BMhQmlPcEmocAOuVg2tCxk (ECDSA)
    MD5:be:41:83:47:1e:f1:8e:f1:66:ab:8e:1a:29:78:c3:fb (ECDSA)
    SHA256:1ywFdPZ5gUEal1hWhOUEvbB/ikmznWCKhvYXK8J1Fqa (ED25519)
    MD5:dc:fa:33:5d:2d:be:f6:ef:b6:c6:0a:64:3b:17:82 (ED25519)
    SHA256:cSuJxwbQrUzew4U9X9E2b71o3tavadnBwFVDrZyrZ4I (RSA)
    MD5:9a:28:b0:3c:b2:a3:ea:06:8a:51:e0:99:21:85:d3:dd (RSA)
IPv4 link-local address: 169.254.70.77/16 (em0)
IPv6 link-local address: fe80::a00:27ff:fe27:898c%em0/64
Administration (root) password set to: Yy30ehCJAxEcq

M-Guard Appliance/m-guard (Amnesiac) (ttyu0)

login:
```

Keep the information printed by your appliance handy. You'll use one of the IP addresses, the password, and the SSH host fingerprints during the initial appliance setup using M-Guard Console.

Note

The settings described above are temporary. They have not been saved and hence will not persist across reboots. When you complete the initial appliance setup using M-Guard Console, appropriate configuration settings will be saved. If you reboot the appliance prior to completing the initial setup, the appliance will generate new temporary settings for configuration settings not yet saved.

Getting Started with M-Guard Console

M-Guard Console can now be set up to manage the Guard Appliance. This section discusses how to install and set up M-Guard Console so that you can finish the initial setup of the Guard Appliance .

Installing M-Guard Console Software

M-Guard Console is a pure Java tool and requires a suitable Java run-time environment. M-Guard Console has been certified to run properly under up-to-date versions of OpenJDK 11 as provided by Adoptium Termurin [https://adoptium.net] (formally AdoptOpenJDK). M-Guard Console should run properly on other distributions of Java 11.

M-Guard Console is distributed as a `.jar` file of the form `M-Guard-Console-M-N-U.jar`. To install, simply place the `.jar` file in a suitable location. Different versions of M-Guard Console can be installed alongside each other.

Running M-Guard Console

To run M-Guard Console, if your desktop has a file association for .jar files, you can simply double click on the `M-Guard-Console-`*`M-N-U`*`.jar`. Otherwise, you can run it from the command line:

java **-jar M-Guard-Console-*M-N-U*.jar**

Note that on Unix and Unix-like systems, you can append an **&** to this command to run it in the background.

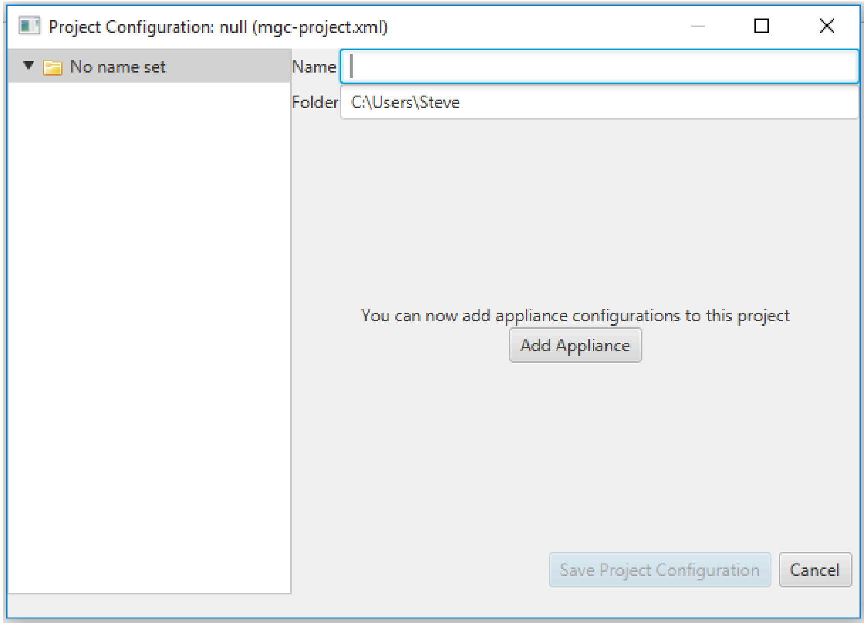
Note that the above command assumes that the **java** program is in the command search path. If not, replace **java** with the full path of the **java** executable.

Project Creation

M-Guard Console groups appliances into projects. For now a single project will do. To create a project, select the File | New Project... menu item. This will present a dialog for you to choose the folder for the project. Use the dialog to navigate to the location where you would like this folder to be, then use the New Folder button to create a new folder with the desired name. Then press Open to complete the initial project creation.

M-Guard Console will then present a Project Configuration dialog.

Figure 2.2. Name the new project



Project Configuration: null (mgc-project.xml)

No name set

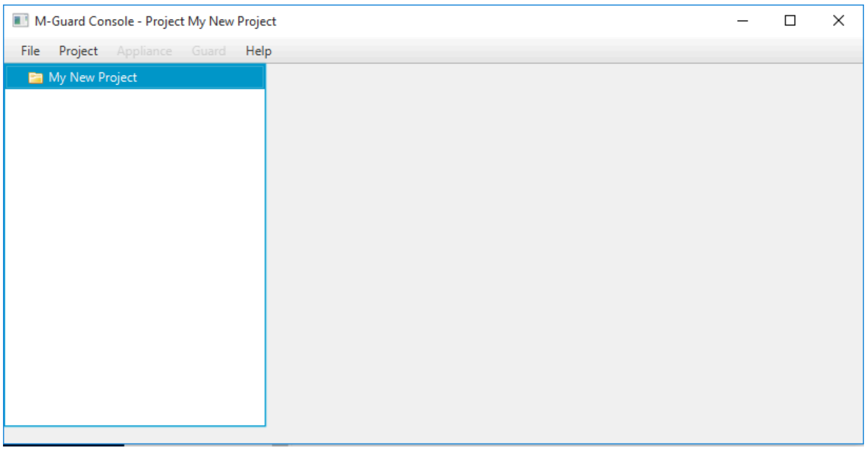
Name

Folder C:\Users\Steve

You can now add appliance configurations to this project

Name the Project and then press Save Project Configuration. This will close the dialog. The Console's main window will be updated as shown below.

Figure 2.3. Initial project created

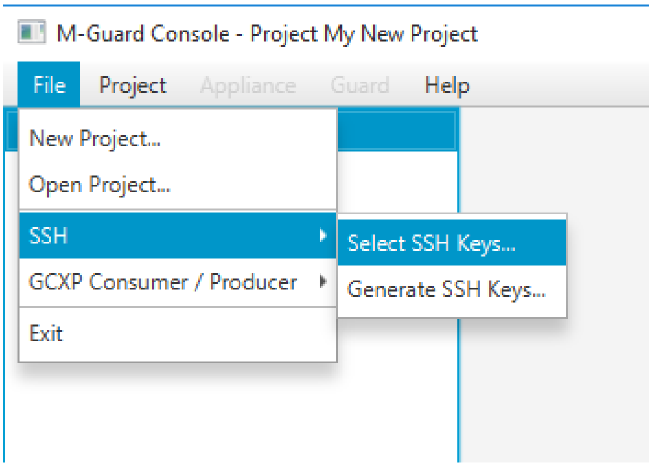


M-Guard Console - Project My New Project

Secure Shell (SSH) Setup

M-Guard Console interacts with the appliance using Secure Shell (SSH). SSH should be configured as discussed below.

Figure 2.4. Setting up SSH Keys

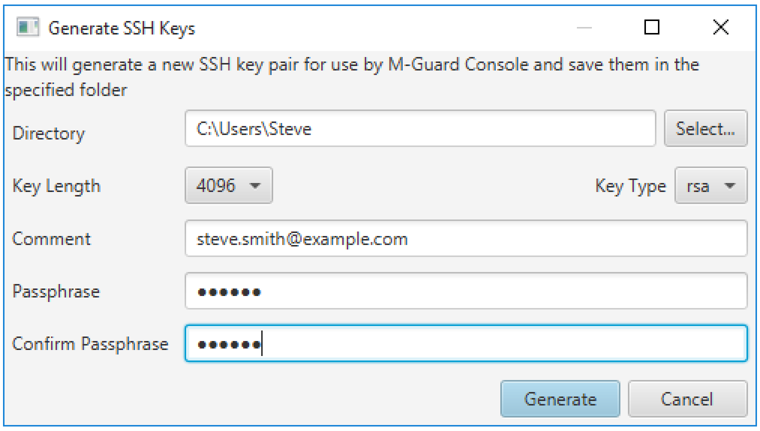


The File | SSH menu item offers two alternatives for configuring SSH keys: Select SSH Keys... for using existing keys and Generate SSH Keys... for generating new keys.

If you have existing SSH keys you would like the console to use, select the Select SSH Keys... menu item. You will then be prompted to specify the folder that contains these keys.

Otherwise, new keys should be generated using the Generate SSH Keys... option. This presents the following dialog:

Figure 2.5. Generate SSH Keys



Generate SSH Keys

This will generate a new SSH key pair for use by M-Guard Console and save them in the specified folder

Directory: C:\Users\Steve

Key Length: 4096

Key Type: rsa

Comment: steve.smith@example.com

Passphrase: ••••••

Confirm Passphrase: ••••••

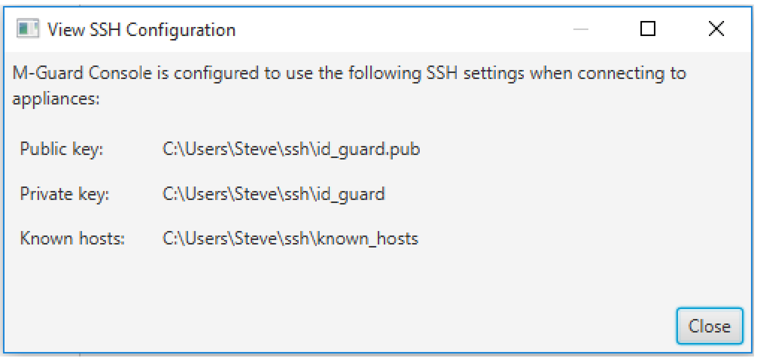
Fill in the form as follows:

- A Folder should be selected for storing the SSH information.
- Key parameters may be selected. The default of RSA 4096 is recommended.
- It is recommended to use the user's email address as the Comment.
- The Passphrase is used to encrypt the SSH private key. This passphrase will be prompted for when M-Guard Console starts. Note that the passphrase may be omitted but this is not generally recommended.

Select Generate to create your SSH setup.

Once SSH keys are configured, you can view the SSH configuration by selecting the File | SSH | View SSH Configuration menu item.

Figure 2.6. SSH Configuration



View SSH Configuration

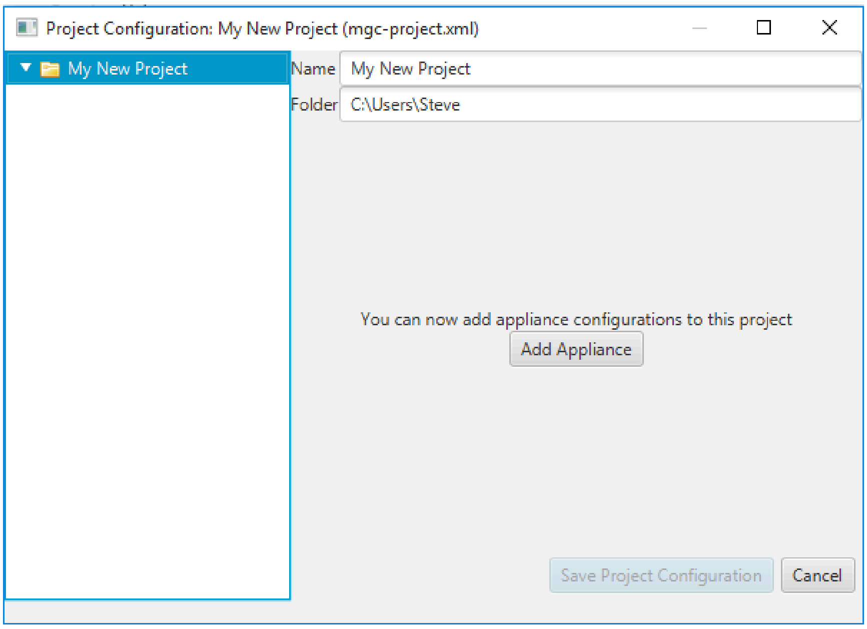
M-Guard Console is configured to use the following SSH settings when connecting to appliances:

Public key:	C:\Users\Steve\ssh\id_guard.pub
Private key:	C:\Users\Steve\ssh\id_guard
Known hosts:	C:\Users\Steve\ssh\known_hosts

Adding an Appliance

You are now ready to add your first appliance to your project. Use Project | Configure to return you to the Project Configuration dialog.

Figure 2.7. Adding a new Appliance



Project Configuration: My New Project (mgc-project.xml)

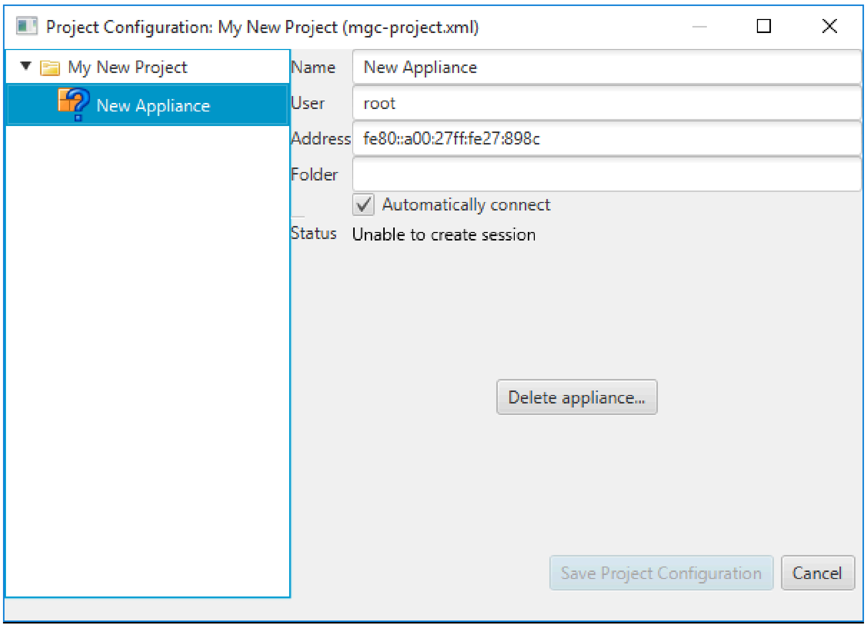
Name: My New Project

Folder: C:\Users\Steve

You can now add appliance configurations to this project

Right-click on your project in the left pane and select Add Appliance from the context menu. This will add an unnamed appliance under your project, which you can then edit, as shown in next screen.

Figure 2.8. Edit the appliance



Project Configuration: My New Project (mgc-project.xml)
My New Project
New Appliance
Delete appliance...
Save Project Configuration
Cancel

Name New Appliance

User root

Address fe80::a00:27ff:fe27:898c

Folder

☑ Automatically connect

Status Unable to create session

Fill in the parameters as follows:

1. Enter a Name for the appliance.

2. Leave User as **`root`**.

3. Set the Address of the appliance. This should be either the IPv4 or IPv6 link-local addresses.

   Note that the IPv6 link-local address may need to be qualified for the management desktop's network interface, such as when the management desktop does not support IPv6 *neighbor discovery*. If the appliance is not first reachable using the unqualified IPv6 link-local address, try again with the address qualified for the management interface. For instance, where the appliance says its IPv6 link local address is `fe80::290:bff:fe7d:144f`, on Windows, assuming the first network interface is connected to the management network, enter **fe80::290:bff:fe7d:144f%*1***. On Unix systems, the interface name is used, as in **fe80::290:bff:fe7d:144f%*eth0***.

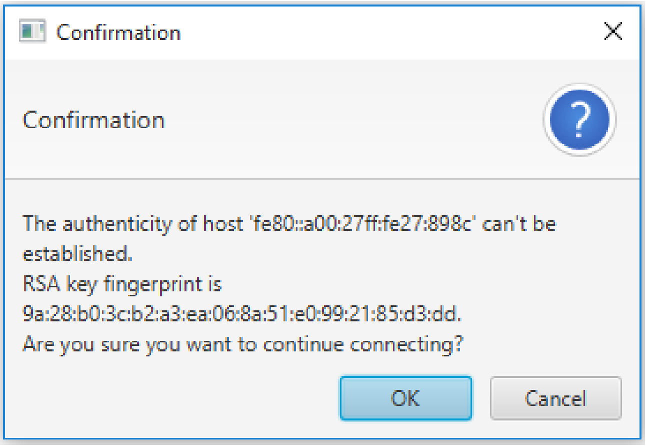
   Note that the IPv4 link-local address (sometimes referred to as a *self-assigned* IP address) may only be usable when the management interface is configured to use DHCP and no DHCP server is available on the management network.

Then choose Save Project Configuration

Select your appliance and choose Appliance | Connect from the menu.

You will be prompted to confirm the appliance's SSH public key as shown below.

Figure 2.9. Confirm the SSH Key



Confirmation

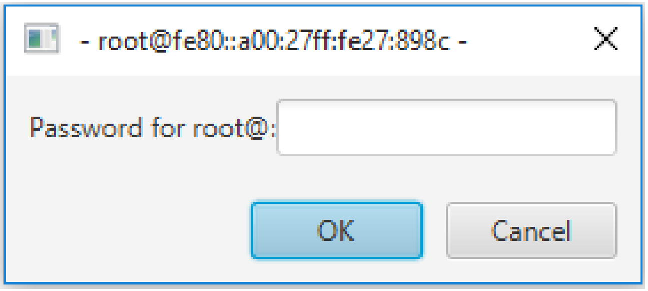
Confirmation

The authenticity of host 'fe80::a00:27ff:fe27:898c' can't be established.
RSA key fingerprint is
9a:28:b0:3c:b2:a3:ea:06:8a:51:e0:99:21:85:d3:dd.
Are you sure you want to continue connecting?

Check that the fingerprint shown here is one of the fingerprints shown on the appliance's console just after it completed booting. If the key matches one of the ones listed, press OK. If it is not, and assuming there is no error, this indicates a *man-in-the-middle* attack. You should not proceed without determining why the fingerprint shown here is not listed by the appliance as one of its fingerprints.

As the appliance does not yet recognize the SSH key you generated above, M-Guard Console will prompt you for the appliance's `root` password.

Figure 2.10. Enter Appliance's root password



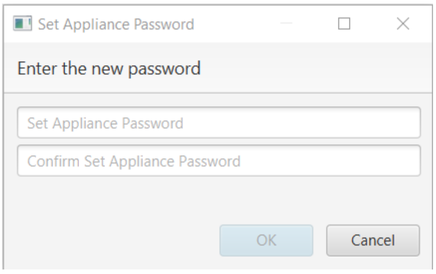
- root@fe80::a00:27ff:fe27:898c -

Password for root@:

Change the Appliance's Administration Password

You may change the appliance's administration (`root`) password to one of your choosing. To change the password, select Appliance | Maintenance | Set Password... to open the Set Appliance Password dialog.

Figure 2.11. Set Appliance Password



Set Appliance Password

Enter the new password

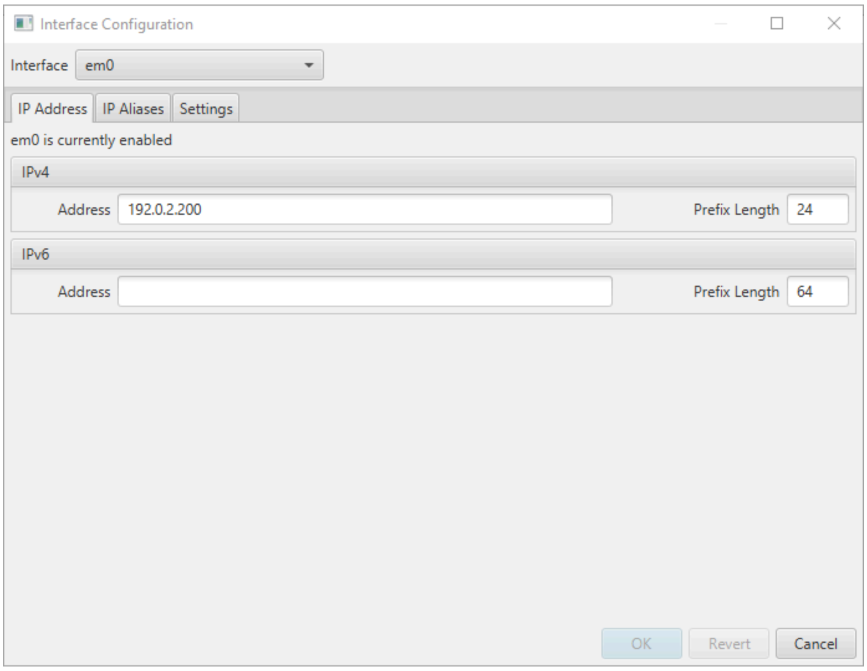
Choose an appropriate password, enter it into both the Set Appliance Password and Confirm Set Appliance Password fields, and then press the OK button. The administration password of the live system will be set to the specified password. Additionally, all settings including the new password will be saved so it persists across reboots.

Renumbering the Appliance and Management Desktop

The appliance and the management desktop network interfaces on the management network should be changed to use appropriate static IP addresses. You need to change the appliance first and then the management desktop.

To change the appliance's management interface, use the Appliance | Setup | Configure Interfaces... menu item. This leads to the following dialog:

Figure 2.12. Interface Configuration Screen



Interface: em0

em0 is currently enabled

IPv4 Address: 192.0.2.200 Prefix Length: 24

IPv6 Address: Prefix Length: 64

Select the management interface. Specify an IPv4 address and prefix length or an IPv6 address and prefix length or, if you wish to configure both IPv4 and IPv6, provide both. Upon pressing the OK button, the appliance's network settings will be immediately changed to use the specified address, although the new address will not be saved.

You should change the project's IP address for the appliance to the appliance's new IP address. You should then exit the M-Guard Console and change the management desktop's network interface to its IP address and prefix length on the management network. You should then restart M-Guard Console.

Communications between the appliance and the M-Guard Console can now be restored. Right-click on the appliance's name and select the Connect item. You will be prompted to enter the appliance's administration (root) password. Be sure to provide the value you set in the previous step.

With communications restored, you can continue with the steps below.

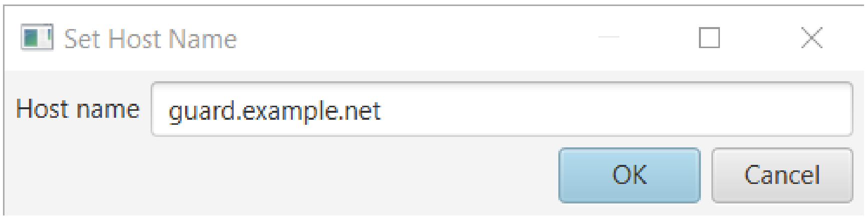
If you are unable to restore communication, first check that the management desktop networking is appropriately configured and then verify that the project's appliance configuration has the correct address for the appliance.

If you want to return the appliance management interface to link-local addresses, you can reboot the appliance by doing a power reset. Upon boot, it should automatically re-assign link local IP addresses. The addresses assigned may differ from the previous boot. You can then reset the management desktop network setting to use link local addressing and change the project's appliance IP address to the appropriate link local addressing. Then you can redo this step.

We'll discuss configuration of the additional interfaces later.

Appliance Host Name Setup

Figure 2.13. Set Host Name



Set Host Name

Host name: guard.example.net

The host name of the appliance should be set using Appliance | Setup | Set Host Name...

This gives the dialog shown above. It is recommended that the name be in the form of domain, e.g. *`guard.example.net`*. You may use a private, local domain, e.g., `.test`. You should avoid `.local` and other special domain names.

Upon pressing OK, the host name of the appliance will be changed in the live system.

Appliance Secure Shell (SSH) Setup

In order for M-Guard Console to connect to the appliance without prompting for the root password, the SSH key generated above needs to be authorized on the appliance to allow this M-Guard Console instance to connect. Do this by selecting Appliance | Setup | Authorize SSH Key...; see Figure A.34, “Authorise SSH Key”

You may require use of SSH keys for remote access authentication as discussed in Figure A.35, “Restrict SSH”

Save the changes and reboot the appliance

Use Appliance | Save Configuration... to save the changes. This is necessary as the M-Guard Console dialogs generally only change the live system. Saving the configuration causes the changes to persist across system reboots.

With the changes saved, you should reboot the appliance to ensure everything works as expected. Use the Appliance | Maintenance | Reboot... menu item to reboot the appliance.

Guard: Getting Started

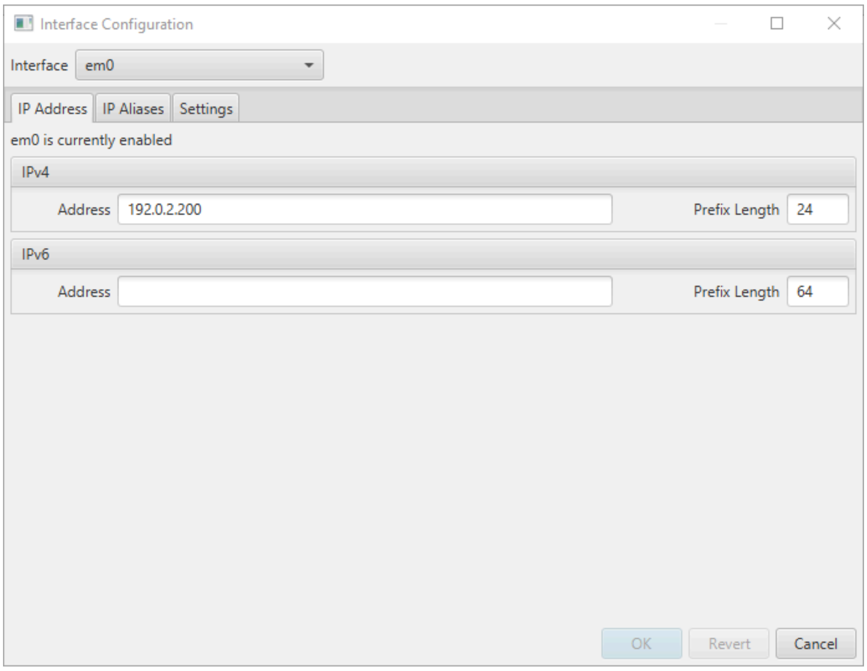
Abstract

This section discusses configuring an appliance to have two additional network interfaces for guarding content between two otherwise separate networks. The sections covers configuring two Guard instances on an appliance. This section assumes an appliance has been initially setup as discussed above.

Configuring Additional Network Interfaces

The appliance needs to be attached to the two networks that content is to be exchanged between. As was done for the management interface in the previous chapter, use M-Guard Console's Appliance | Setup | Configure Interfaces... menu item to configure two additional interfaces.

Figure 2.14. Interface Configuration Screen



Interface: em0

em0 is currently enabled

IPv4

Address: 192.0.2.200 Prefix Length: 24

IPv6

Address: Prefix Length: 64

Select the first additional interface. Specify an IPv4 address and prefix length or an IPv6 address and prefix length or, if you wish to configure both IPv4 and IPv6, provide both.

The IP Aliases tab can be used to assign additional IP addresses to the management interface. In this scenario, no IP aliases are needed.

The Name tab allows you to change the name of the interface.

Repeat for the second additional interface.

Verify connectivity between the Appliance and each of application servers. You can use Appliance | Execute Command... to run **ping -c 3** ***`198.51.100.2`***. where *`198.51.100.2`* is the IP address of the appliance. If you using IPv6, replace **ping** with **ping6**.

With connectivity verified, save the settings using Appliance | Save Configuration.

Appliance TLS Configuration

An appliance must be configured with a private key and associated certificate. These are used by Guard instances when establishing TLS connections with GCXP peers. The appliance will have trust anchors that Guard instances use in authenticating GCXP peers. The TLS configuration is also used to support secure remote system logging when enabled.

Use Appliance | Setup | Generate TLS CSR... to create a certificate signing request (CSR) for an appliance certificate. This will cause the appliance to generate a private key and a CSR. M-Guard Console will prompt you to save the CSR to a file on the management desktop. The private key is not transferred to the management desktop.

The CSR file contains a PEM-encoded certificate signing request. This should be provided to a suitable certification authority (CA) for signing.

The appliance requires that the signed certificate, as well as the complete chain of CA certificates, be saved in a PEM-encoded file starting with the *end-entity* certificate and ending with the certificate for *root CA*. Your CA may or may not provide such a file when it returns the requested certificate. If it does not, you must create the file using appropriate tools.

The certificate (with complete chain) file can be installed on the appliance by selecting Appliance | Setup | Load TLS Certificate....

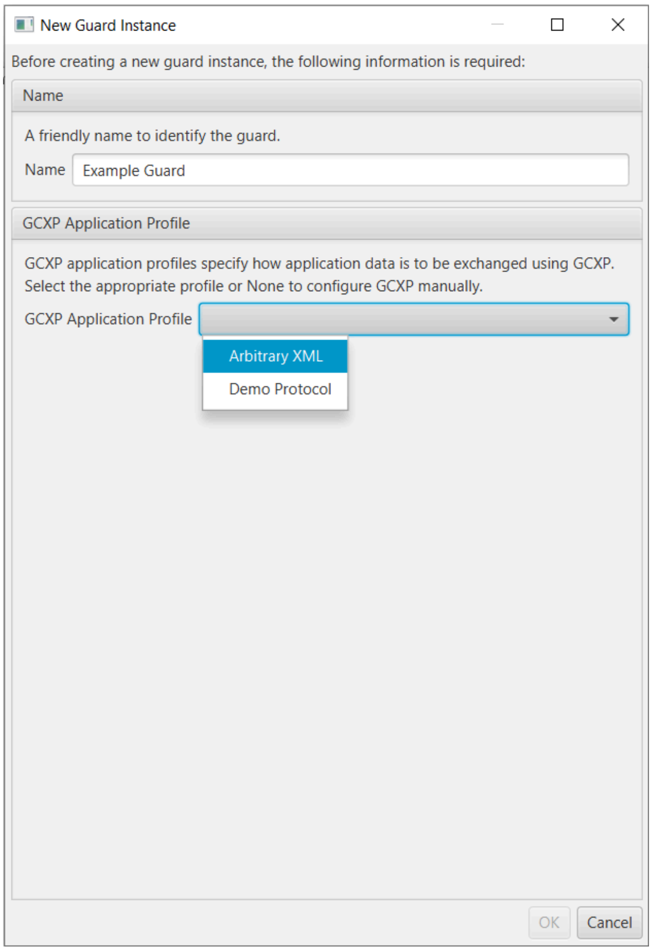
To install a trust anchor, use Appliance | Setup | Load TLS Trust Anchor... This will prompt you to specify a file containing one or more PEM-encoded CA certificates to trust. This must include an appropriate trust anchor for authentication of GCXP peers, such as the certificate(s) of the root CA(s) for their certificates.

Isode's Sodium CA product can be used to provide the necessary certificate authority services.

Creating New Guard Instances

To create a new Guard instance on a Guard Appliance select Appliance | New Guard…. This will open up the New Guard Instance dialog.

Figure 2.15. New Guard Instance dialog



Before creating a new guard instance, the following information is required:

A friendly name to identify the guard.

Name: Example Guard

GCXP application profiles specify how application data is to be exchanged using GCXP. Select the appropriate profile or None to configure GCXP manually.

Enter a name for the Guard instance and select an appropriate GCXP application profile.

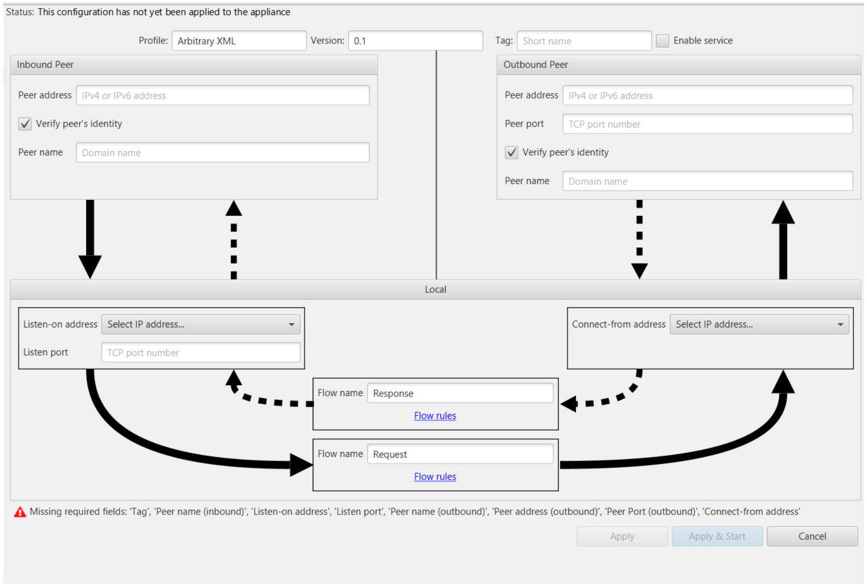
GCXP application profiles specify how application data is to be exchanged using GCXP. Two profile are built into the Console:

Arbitrary XML	A profile to allow any arbitrary XML
Demo Protocol	A profile for the Demo Protocol

Additional profiles may be imported.

After providing a name and selecting an appropriate GCXP application profile, click OK to bring up the Configure Guard Instance dialog for the new Guard instance. Here the initial configuration of the Guard instance should be specified.

Figure 2.16. Guard Configuration Form



Status: This configuration has not yet been applied to the appliance

Profile: Arbitrary XML

Version: 0.1

Tag: Short name

□ Enable service

Inbound Peer

Peer address: IPv4 or IPv6 address

☑ Verify peer's identity

Peer name: Domain name

Outbound Peer

Peer address: IPv4 or IPv6 address

Peer port: TCP port number

☑ Verify peer's identity

Peer name: Domain name

Local

Listen-on address: Select IP address...

Listen port: TCP port number

Connect-from address: Select IP address...

Flow name: Response

Flow name: Request

Missing required fields: 'Tag', 'Peer name (inbound)', 'Listen-on address', 'Listen port', 'Peer name (outbound)', 'Peer address (outbound)', 'Peer Port (outbound)', 'Connect-from address'

Each Guard instance transfers application data in a single direction, from a GCXP peer on the input side to a GCXP peer on the output side. In the form above, the solid arrows represent the flow of application data while the dashed arrows represents the optional reverse flow. The Inbound Peer pane specifies information about the GCXP peer that is expected to connect to the Guard on the input side and below this are configuration items which specify how the Guard is to listen for connections on the input side. The Outbound Peer pane specifies information about the GCXP peer that the Guard will connect to on the output side and below this are configuration items which specify further details on how the Guard will make connections on the output side. At the bottom are two panes for configuring the two flows.

Below are some notes on filling in the fields:

- Profile and Version give details of the GCXP application profile used to create the Guard. These fields can not be edited.
- Tag is a (typically short) string used in syslog to identify this Guard.
- Enable Service toggles whether the service created for this Guard will be enabled.
- The Inbound Peer gives information about the peer connecting to the Guard:
  - Peer Address is the IP address that the peer will connect from. When specified, the Guard will only accept connections from this address. When empty the Guard will accept connections from any IP address.
  - Verify Peer's Identity tells the Guard to verify the peer's identity using TLS authentication. It is recommended that be selected.
  - Peer name is the domain name of the inbound peer. This is used in verification of the peer's identity when enabled. It is also used in event logging.
- The Outbound Peer is the outbound peer that the Guard will connect to:
  - Peer Address is the address the Guard will use to connect.

  - Peer Port is the number of the TCP port that the Guard will connect to. No port is reserved specifically for GCXP. You may use any appropriate port of your choosing (see note below for suggestions).

  - Verify Peer's Identity tells the Guard to verify the peer's identity using TLS authentication. It is recommended that be selected.

  - Peer name is the domain name of the outbound peer, This is used in verification of the peer's identity when enabled. It is also used in event logging.

- The Guard networking options are as follows:

  - Listen-on address specifies the particular IP address the Guard is to listen on for inbound connections. It is recommended that this be set to the Guard's input side IP address. If not set, the Guard will listen for connections on all interfaces.

  - Listen Port is the TCP port the Guard is to listen on for input side connections to it. No port is reserved specifically for GCXP. You may use any appropriate port of your choosing (see note below for suggestions).

  - Connect-from address specifies the particular IP address the Guard is to use to make its outbound connections from. It is recommended that this be set to the Guard's output side IP address. If not set, the Guard will leave the choice of which IP address to use to the Appliance.

- The two Flow Name fields name the respective flows. The flows must be named and must have different names. As these names are used in event logging, typically short names are used.

  The flow from left to right (bottom flow) is shown with solid arrows. This indicates that the flow carries application content.

  The flow from right to left (top flow) is, by default, shown with dotted arrows. This indicates that the flow is only used to signal to the content provider that the content has been passed to the content consumer or has been rejected. This signalling may be disabled.

  Each flow can be configured to meet application-specific and deployment-specific needs as discussed in subsequent chapters.

Note: when choosing a TCP port for GCXP use, pick a port unlikely to conflict with any port in use on the Guard, the management desktop, or the application servers. Ports 24, 35, 57, 59, 75, 77, and 87 are reserved for private use and are unlikely to be in use.

Once the required settings have been entered on the Configure Guard Instance dialog select Apply or Apply & Start to create the Guard instance on the appliance. To disregard changes and to cancel creating the new instance select Cancel.

Chapter 3. Maintaining M-Guard

Abstract

This chapter describes how to perform various maintenance tasks.

Backing up your Appliance Configuration

You can use M-Guard Console to make a backup of your *last-saved* appliance configuration at any time. See the section called “Back Up”.

It is recommended you make regular backups of your configuration.

A number of options are available for automating scheduled backups to a backup server. Contact Isode Support to discuss options. See the section called “Support Queries and Bug Reporting”.

Restoring your Appliance Configuration from a Backup

You can use M-Guard Console to restore the Guard Appliance configuration from a backup. See the section called “Restore”.

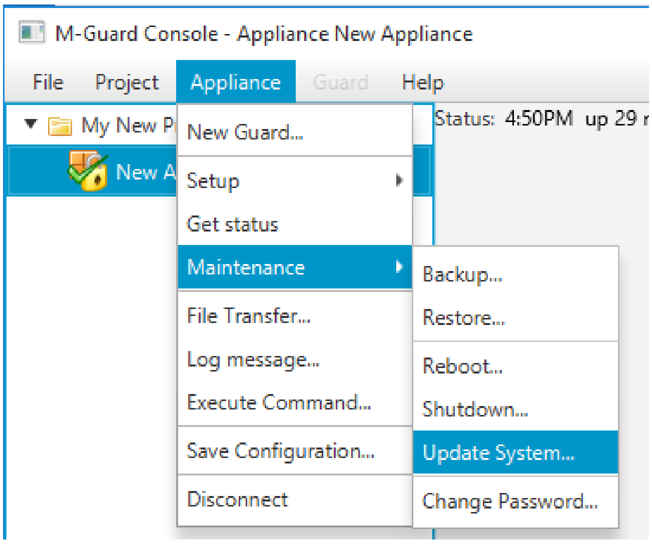
Updating M-Guard Console

As each M-Guard Console package is uniquely named, you can install the new version alongside any previously installed versions and put it to use. See the section called “Installing M-Guard Console Software” for instructions.

When you no longer need an old version, you can simply delete its `.jar` file.

Updating M-Guard Appliance

Figure 3.1. Update System in M-Guard Console



An appliance system can be updated using the *latest* M-Guard Console. Use: Appliance | Maintenance | Update System.... See the section called “Updating M-Guard Console”.

It is recommended you backup your appliance configuration before upgrading your appliance. See the section called “Backing up your Appliance Configuration”.

You will be asked to select the new 'update' system image to update to. Once selected, you will be asked to confirm the install before it is undertaken. After the install finishes, reboot the appliance to run the new system image.

The previous system remains available until a further upgrade is performed, just in case you need to revert to it.

Reverting M-Guard Appliance to the previous system

Run the command **revert && reboot** using Appliance | Execute Command...

This command will revert the appliance to the previous system and immediately reboot the system without saving the current configuration.

It is recommended you backup your appliance configuration before reverting M-Guard Appliance to a previous system. See the section called “Backing up your Appliance Configuration”.

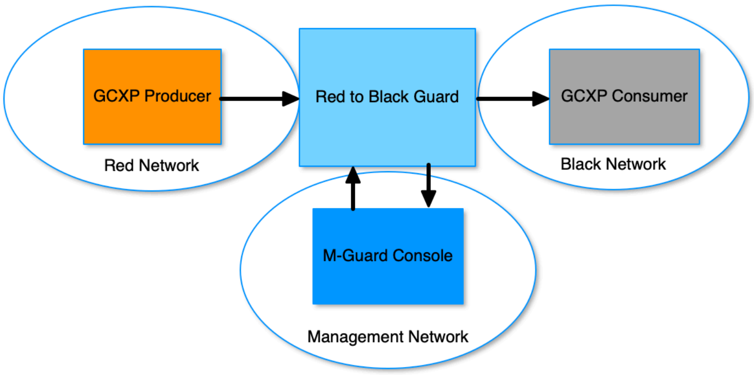
Chapter 4. Testing Guard instances

Abstract

This chapter gives instruction in the testing of Guard instances.

GCXP Producer & GCXP Consumer

Figure 4.1. GCXP Producer and GCXP Consumer



In order to test a Guard instance in a realistic manner, Isode provides two desktop tools that can communicate with Guard instances using GCXP in the manner that all peers will connect to the Guard:

- GCXP Consumer accepts a GCXP connection from a Guard instance as an XML content consumer. The Consumer displays the XML content carried in all GCXP messages received. It also shows activity at the GCXP layer in a second window.

- GCXP Producer connects using GCXP to a Guard instance as an XML content producer. Content can be entered by various means including direct user input and cut-n-paste. Content may also be loaded from files or from built-in set of examples. At the user's direction, the content is sent to the Guard instance. It also shows activity at the GCXP layer in a second window.

To start the consumer and producer, use the following commands:

```
java -jar M-Guard-Console-M-N-U.jar --gcxp-consumer
```

```
java -jar M-Guard-Console-M-N-U.jar --gcxp-producer
```

Note that on Unix/Linux/MacOS, you can append an **&** to these commands to run the programs in the background.

They enable configuration of the following parameters:

- GCXP Consumer: the TCP port to listen on.

- GCXP Producer: the IP address and TCP port to connect to.

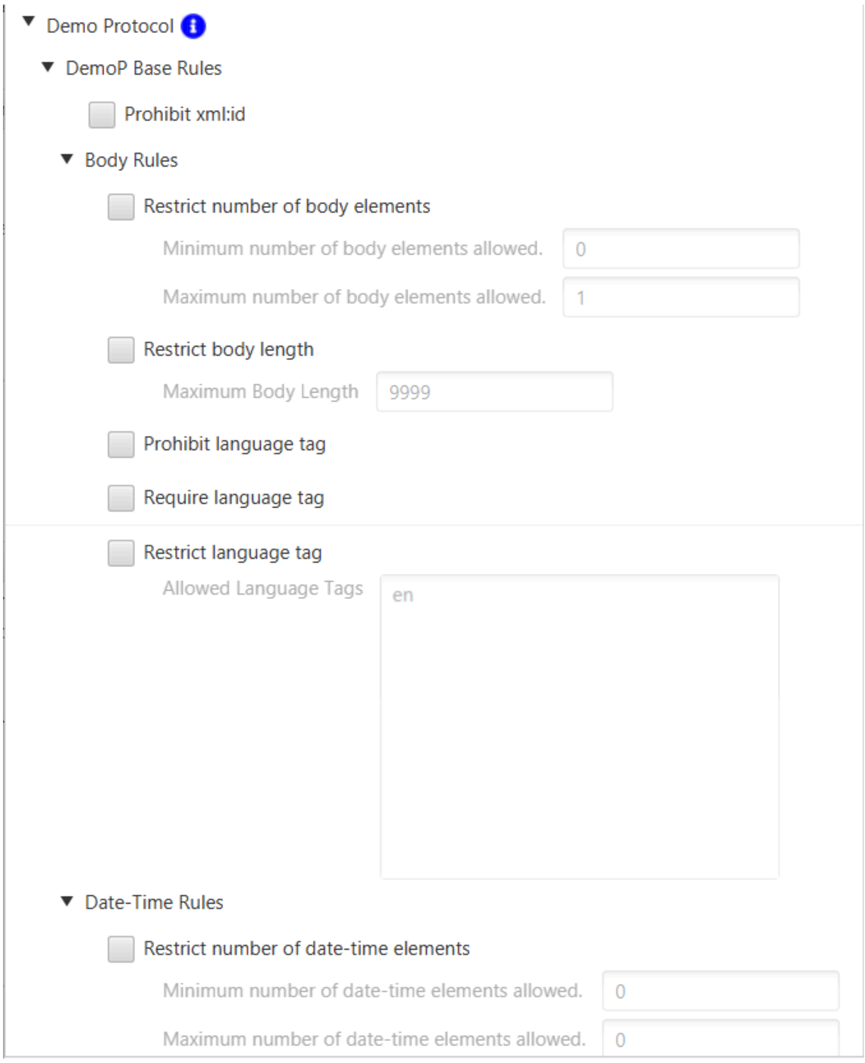
GCXP Consumer and GCXP Producer can be used to test a Guard instance by attempting to send traffic through it; arbitrary content may be sent so that any Guard configuration can be tested.

TLS Authentication Configuration

The GCXP Consumer and GCXP Producer must be configured with a private key, associated certificate and trust anchor. There is a dialog to generate a private key and a certificate signing request (CSR). The inbound and outbound service names, which must be different, are the names the Guard instance will use in authenticating the inbound and outbound GCXP connections.

Demo Protocol

Figure 4.2. Demo Protocol Configuration



▼ Demo Protocol
▼ DemoP Base Rules
□ Prohibit xml:id
▼ Body Rules
□ Restrict number of body elements
Minimum number of body elements allowed. 0
Maximum number of body elements allowed. 1
□ Restrict body length
Maximum Body Length 9999
□ Prohibit language tag
□ Require language tag
□ Restrict language tag
Allowed Language Tags en
▼ Date-Time Rules
□ Restrict number of date-time elements
Minimum number of date-time elements allowed. 0
Maximum number of date-time elements allowed. 0

M-Guard Console has built in rules for checking a Demo Protocol. Configuration of these rules is shown above.

The GCXP Producer allows selection of Demo Protocol content, as shown above. This content can be used as is, or edited before sending. A range of Demo Protocol content is supplied so that Guard validation of each of the rules can be easily shown.

Chapter 5. Troubleshooting

Abstract

This chapter notes various approaches to troubleshooting.

- Look at syslog output to view Guard and other activity.
- Use diagnostics and options in M-Guard Console
- Run commands on the Guard Appliance to get additional information.
- Access files stored on the appliance.
- Contact Isode support for assistance. See the section called “Support Queries and Bug Reporting”.

Monitoring with Syslog

The appliance logs events using syslog [https://en.wikipedia.org/wiki/Syslog]. By default, events are recorded in log files on the appliance. The appliance can be configured to send syslog events to a remote syslog server. Guard events are logged using the daemon facility.

To configure the appliance to send Guard events (specifically daemon.* events) to a remote syslog server, select the M-Guard Console's Appliance | Setup | Remote Logging... menu item to raise a Remote Logging Configuration dialog. For demonstration purposes, Isode recommends Visual Syslog Server for Windows [https://github.com/MaxBelkov/visualsyslog/blob/master/README.md] running on the management desktop.

It is anticipated that the operational M-Guard configurations will have preferred management tools, which can consume all M-Guard Appliance events. In absence of an existing compatible logging server, RSYSLOG [https://rsyslog.com] using RELP over TLS for event logging is recommended.

Executing Commands on the Appliance

There are three ways to execute commands on the appliance.

1. Run commands on the device from M-Guard Console using Appliance | Execute Command. This is only suitable for running simple, non-interactive commands which complete immediately, such as **tail /var/log/messages**.
2. Use the Guard Appliance command line using SSH. It is highly recommended that customers install a SSH client and use it for appliance command line access, such as OpenSSH for Windows [https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_overview].
3. Use the Guard Appliance console for command line access. This requires access to the appliance's console. It is recommended the appliance's console only be used for command line access with SSH is not available.

Accessing Files Stored on the Appliance

While in many cases, the best way to access appliance files is through the command line, short files or the last few lines of a file can be accessed simply by running **cat** or **tail** command. To watch changes to a file, such as a log file, use of **tail** with the **-F** flag is a good option. For instance, **tail -F /var/log/messages**. Note that as this command will not immediately complete and hence it should not be invoked using Execute Command.

You can copy files from the appliance to the management desktop using a Secure Copy client such as scp(1) or Secure File Transfer client such as sftp(1). M-Guard Console also provides a Secure File Transfer client.

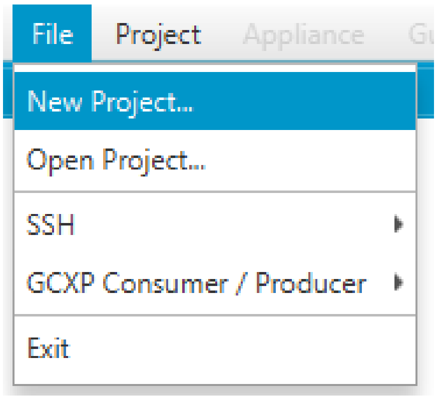
Appendix A. M-Guard Console Reference

Abstract

This chapter gives a reference for all M-Guard Console functions. It is structured according to the M-Guard Console menu structure.

File Menu

Figure A.1. File Menu



New Project...

Creates a new project in a chosen folder

Open Project...

Opens an existing project

NOTE: M-Guard Console allows only one open project at a time.

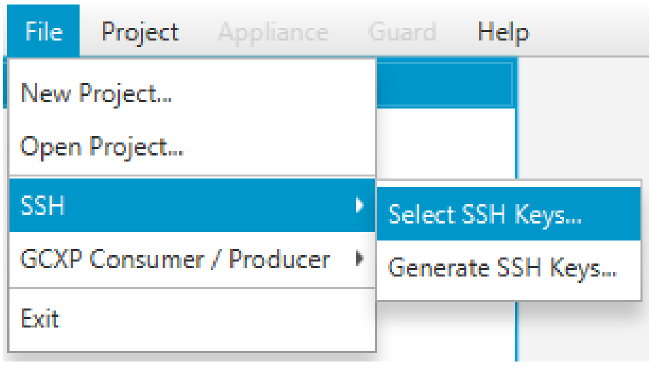
Secure Shell (SSH) Management

This menu allows the setting up and management of the SSH keys M-Guard Console used to connect to the appliance.

Setting up SSH Keys...

If no SSH keys are configured for M-Guard Console the menu will look like this

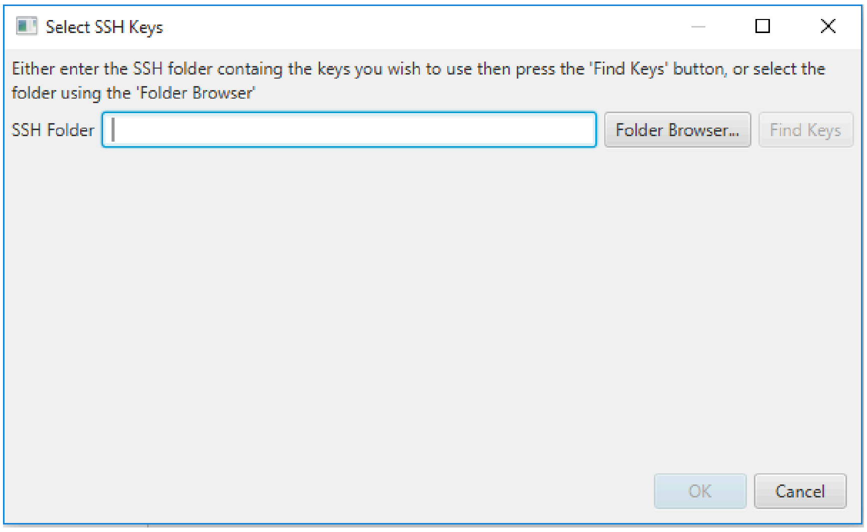
Figure A.2. SSH Menu when SSH not configured



Select SSH Keys...

Allows the selection of an existing set of SSH keys to be used with M-Guard Console.

Figure A.3. Select SSH Keys



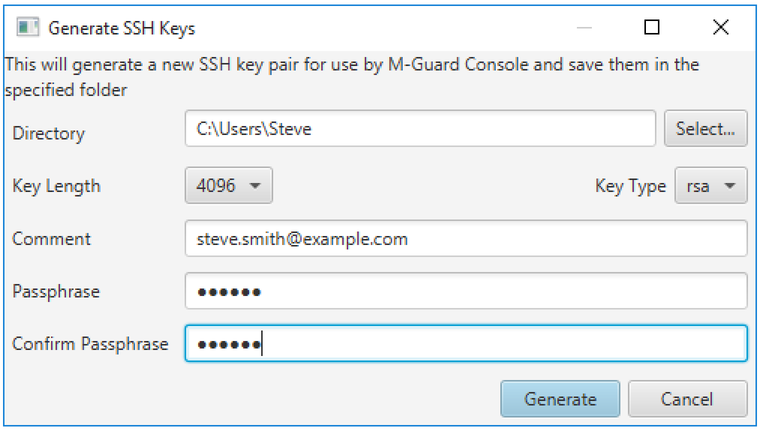
Either enter the SSH folder containg the keys you wish to use then press the 'Find Keys' button, or select the folder using the 'Folder Browser'

SSH Folder

Generate SSH Keys...

Open a dialog to generate a new set of SSH keys for used with M-Guard Console.

Figure A.4. Generate SSH Keys



This will generate a new SSH key pair for use by M-Guard Console and save them in the specified folder

Directory: C:\Users\Steve

Key Length: 4096

Key Type: rsa

Comment: steve.smith@example.com

Passphrase: ••••••

Confirm Passphrase: ••••••

The fields are as follows:

Directory
: The folder in which the SSH keys are to be stored.

Key Length & Key Type
: Length and type of SSH key to generate.

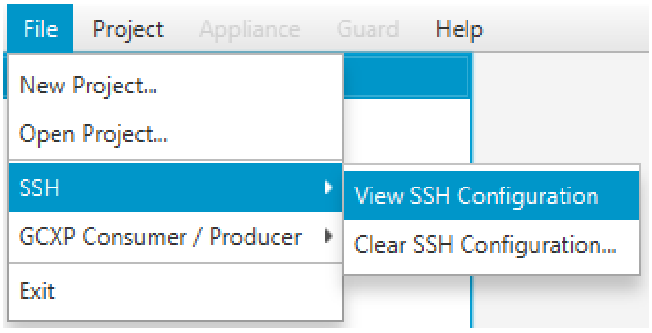
Comment
: A comment for the SSH key. An email address is recommended.

Passphrase & Confirm Passphrase
: Passphrase to encrypt (and needed to decrypt) the SSH key.

Managing existing SSH configuration

If SSH is currently configured for M-Guard Console the menu will look as follows.

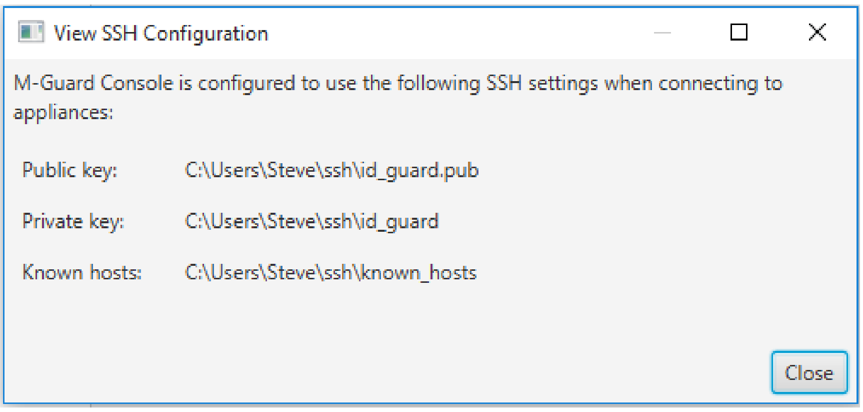
Figure A.5. SSH Menu when SSH configured



View SSH Configuration...

Shows the SSH configuration that M-Guard Console is configured to use.

Figure A.6. Viewing the SSH Configuration



View SSH Configuration

M-Guard Console is configured to use the following SSH settings when connecting to appliances:

Public key: C:\Users\Steve\ssh\id_guard.pub

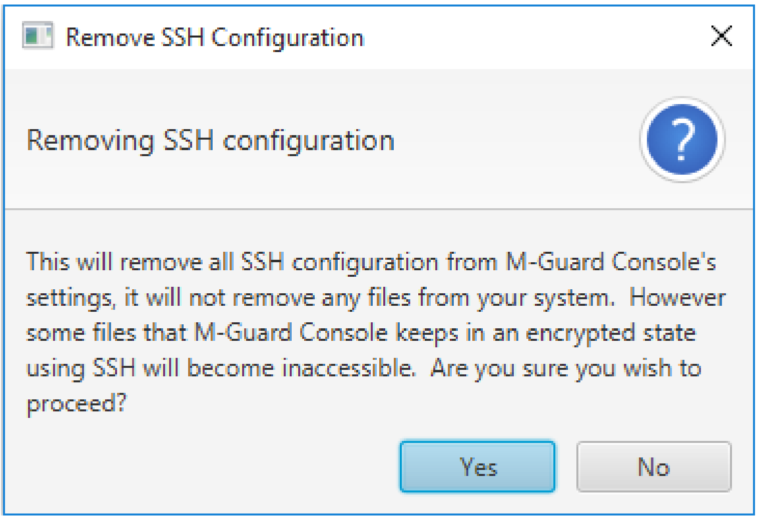
Private key: C:\Users\Steve\ssh\id_guard

Known hosts: C:\Users\Steve\ssh\known_hosts

Clear SSH Configuration...

Removes the SSH configuration after the following warning.

Figure A.7. Removing the SSH Configuration



Remove SSH Configuration

Removing SSH configuration

This will remove all SSH configuration from M-Guard Console's settings, it will not remove any files from your system. However some files that M-Guard Console keeps in an encrypted state using SSH will become inaccessible. Are you sure you wish to proceed?

GCXP Consumer/Producer

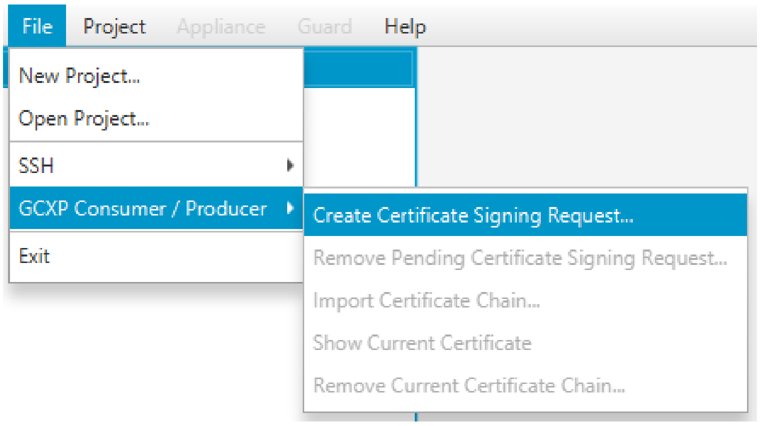
This menu allows the setting up and management of the TLS certificate used for the GCXP Producer and GCXP Consumer applications.

Note: it will not be possibly to fully configure the GCXP TLS certificate unless an SSH key is configured.

No TLS certificate configured

When no TLS certificates is configured the menu will look like the following:

Figure A.8. GCXP Consumer/Producer when no TLS certificate is configured

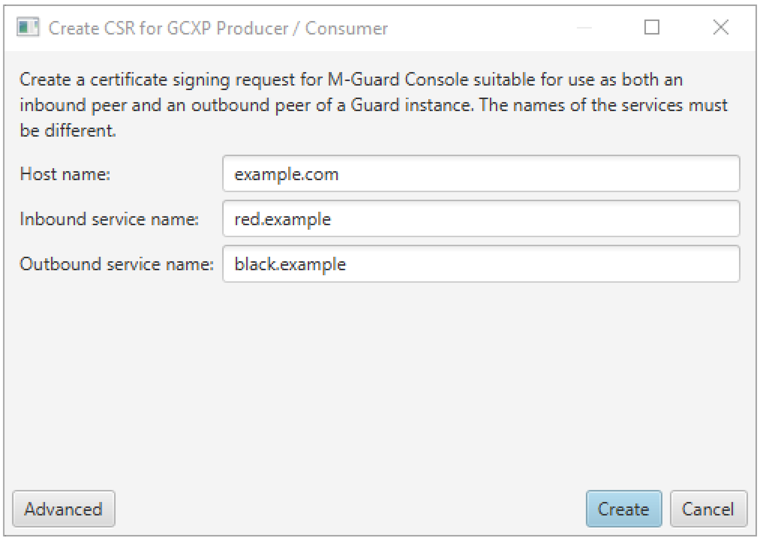


File
Project
Appliance
Guard
Help
New Project...
Open Project...
SSH
GCXP Consumer / Producer
Exit
Create Certificate Signing Request...
Remove Pending Certificate Signing Request...
Import Certificate Chain...
Show Current Certificate
Remove Current Certificate Chain...

Create Certificate Signing Request

Opens a dialog for create a Certificate Signing Request (CSR) for the GCXP Producer/Consumer

Figure A.9. Create CSR for GCXP Producer / Consumer dialog



Create CSR for GCXP Producer / Consumer

Create a certificate signing request for M-Guard Console suitable for use as both an inbound peer and an outbound peer of a Guard instance. The names of the services must be different.

Host name: example.com

Inbound service name: red.example

Outbound service name: black.example

The fields are as follows:

Host name
: Host name to use in the certificate request.

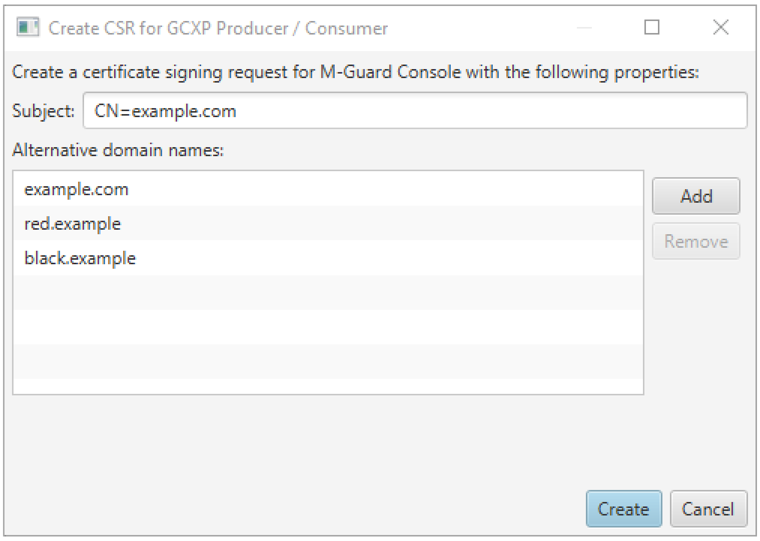
Inbound service name
: Name the Guard instance is expecting for the inbound peer. Will be added to the certificate request as a subject alt name.

Outbound service name
: Name the Guard instance is expecting for the outbound peer. Will be added to the certificate request as a subject alt name.

An advanced mode for the editor can be enabled by selecting the Advanced button

NOTE: Once enabled advanced mode can not be exited apart by exiting the dialog. It is not recommended that it used and is included for advanced user cases only.

Figure A.10. Advanced view for Create CSR for GCXP Producer / Consumer dialog



Create CSR for GCXP Producer / Consumer

Create a certificate signing request for M-Guard Console with the following properties:

Subject: CN=example.com

Alternative domain names:

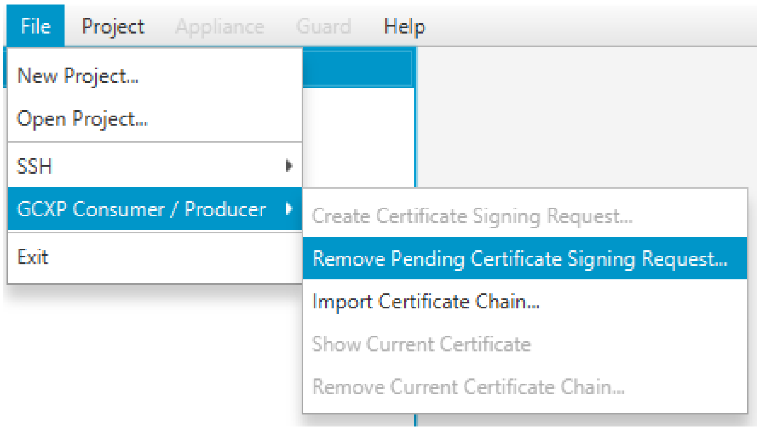
example.com
red.example
black.example

Once the fields are completed the Create button will close the dialog and open a file save dialog to save a Certificate Signing Request (CSR) to. The CSR should then be passed to an appropriate Certificate Authority (CA) for signing. Once the request has being fulfilled the CA should return a full Certificate Chain for importing into the GCXP Producer / Consumer applications

Whilst a certificate signing request is pending

When a certificate signing request (as created above) is pending the menu will look as follow

Figure A.11. GCXP Consumer/Producer when a CSR is pending


File
Project
Appliance
Guard
Help
New Project...
Open Project...
SSH
GCXP Consumer / Producer
Exit
Create Certificate Signing Request...
Remove Pending Certificate Signing Request...
Import Certificate Chain...
Show Current Certificate
Remove Current Certificate Chain...

Remove Pending Certificate Signing Request...

Removes any information on the pending certificate signing request from the GCXP Producer / Consumer applications internal stores.

NOTE: If pending certificate signing request is subsequently fulfilled, it will not be possible to import the issued certificate. A new certificate signing request will need to be created and passed to the CA.

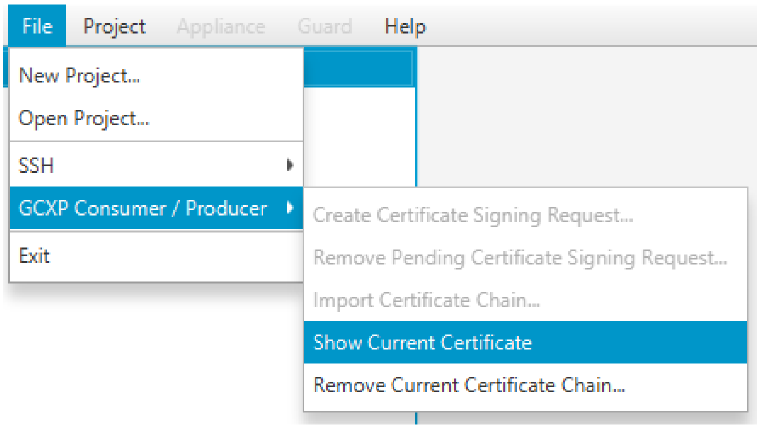
Import Certificate Chain...

Imports the full certificate chain issued by the CA for the certificate signing request.

When TLS is configured

When TLS is configured for the GCXP Producer/Consumer applications the menu will look like this:

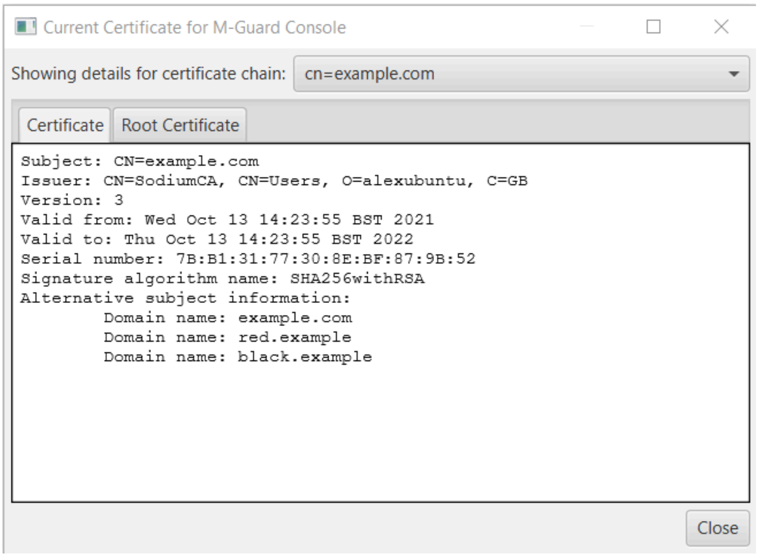
Figure A.12. GCXP Consumer/Producer when TLS is configured


File
Project
Appliance
Guard
Help
New Project...
Open Project...
SSH
GCXP Consumer / Producer
Exit
Create Certificate Signing Request...
Remove Pending Certificate Signing Request...
Import Certificate Chain...
Show Current Certificate
Remove Current Certificate Chain...

Show Current Certificate

Opens a dialog that shows the configured TLS certificate and chain for the GCXP Producer/Consumer applications.

Figure A.13. Current Certificate dialog



Current Certificate for M-Guard Console

Showing details for certificate chain: cn=example.com

```
Subject: CN=example.com
Issuer: CN=SodiumCA, CN=Users, O=alexubuntu, C=GB
Version: 3
Valid from: Wed Oct 13 14:23:55 BST 2021
Valid to: Thu Oct 13 14:23:55 BST 2022
Serial number: 7B:B1:31:77:30:8E:BF:87:9B:52
Signature algorithm name: SHA256withRSA
Alternative subject information:
        Domain name: example.com
        Domain name: red.example
        Domain name: black.example
```

Remove Current Certificate Chain...

Deletes the TLS certificate chain from the GCXP Producer/Consumer configuration.

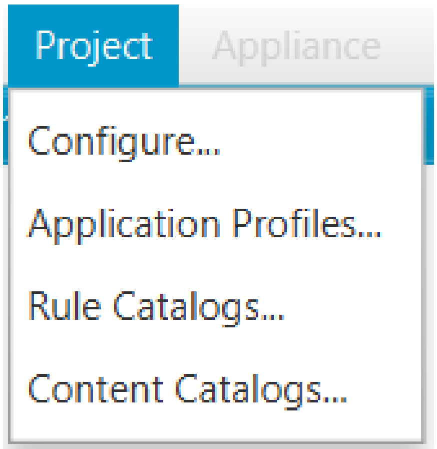
Exit

Closes M-Guard Console.

Project Menu

This menu is also available by right-clicking on the project name.

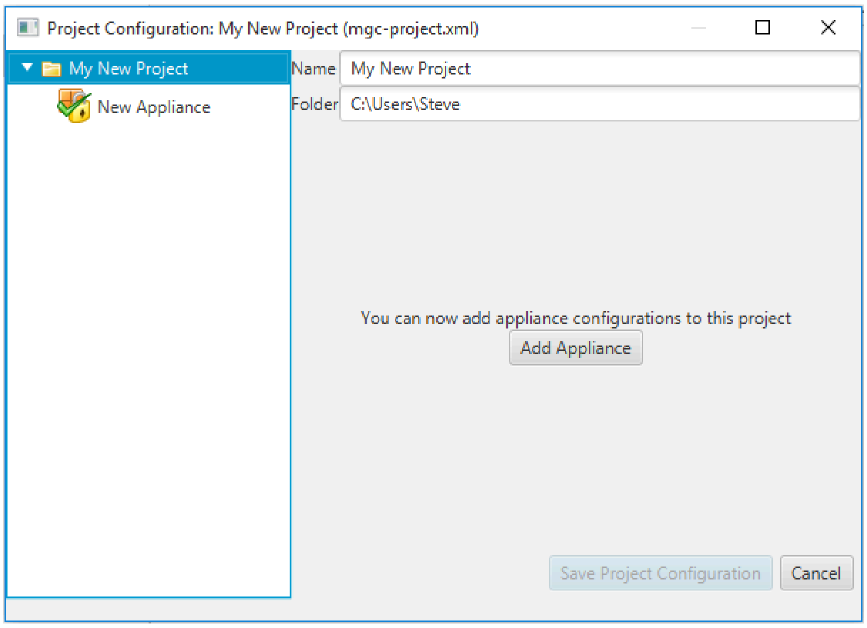
Figure A.14. Project Menu



Configure

This is used to manage projects and to add and remove appliances.

Figure A.15. Configure



Project Configuration: My New Project (mgc-project.xml)

- My New Project
  - New Appliance

Name: My New Project

Folder: C:\Users\Steve

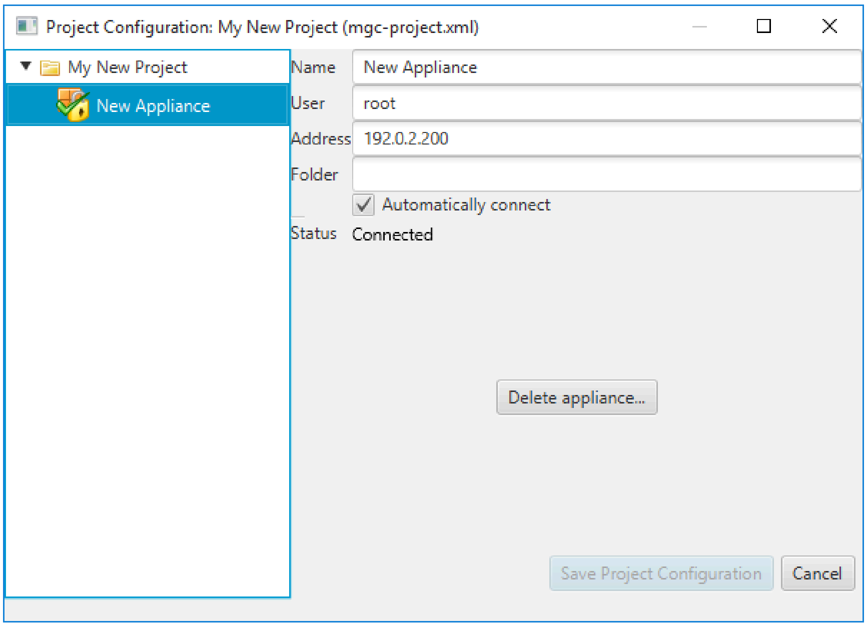
You can now add appliance configurations to this project

The Folder field shows the folder in which the project is stored. It is read-only.

You can change the name of the project by changing the value of the Name field.

A new appliance can be added by clicking on the Add Appliance button on the screen or right-clicking on the project name.

Figure A.16. Existing Appliance Configuration



Project Configuration: My New Project (mgc-project.xml)

- My New Project
  - New Appliance

Name: New Appliance

User: root

Address: 192.0.2.200

Folder:

☑ Automatically connect

Status: Connected

If an appliance is selected, the above dialog is shown. The fields and controls in the dialog are described below:

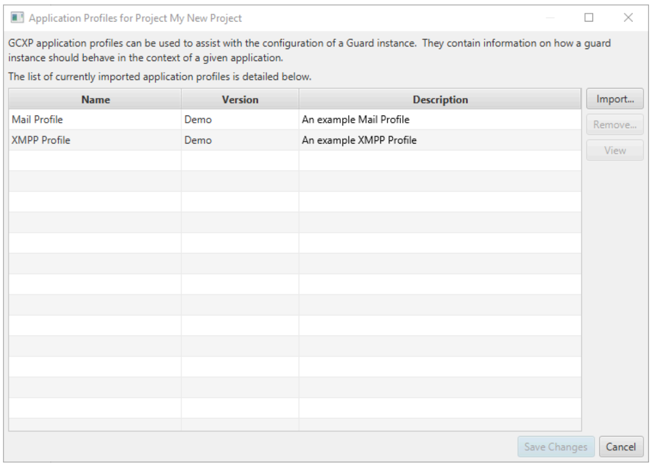
- Name. Setting the name of the appliance. This may be a descriptive name, e.g., `New Appliance`.
- User. User name for authenticating to the appliance. This should be set to root (the default).

- Address. This is the IP address of the appliance.
- Folder. This is the local folder on the host where M-Guard Console is running which will be shown when using the File Transfer command (see Figure A.48, “File Transfer”).
- Automatically connect. If selected, M-Guard Console will automatically connect to the appliance.
- Status. This shows the current connection status to this appliance.
- Delete Appliance. This can be done by clicking the button or right-clicking on the appliance.

Application Profiles

Opens the Application Profiles dialog for the management of GCXP application profiles in the project

Figure A.17. Application Profile Dialog



GCXP application profiles can be used to assist with the configuration of a Guard instance. They contain information on how a guard instance should behave in the context of a given application.

The list of currently imported application profiles is detailed below.

Name	Version	Description
Mail Profile	Demo	An example Mail Profile
XMPP Profile	Demo	An example XMPP Profile

The list of currently imported GCXP application profiles will be shown in the table. It consists of three columns: name, version and description. These values are extracted from the corresponding fields in the application profile.

The buttons in the dialog do as follows:

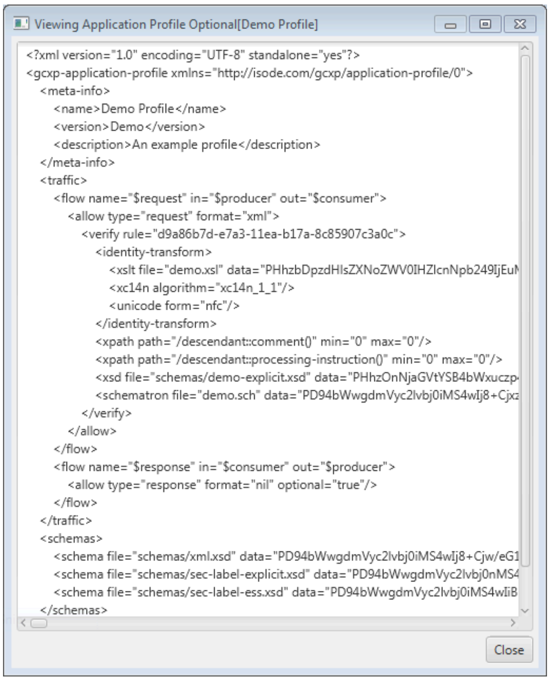
Import... Imports an external application profile into the current project.

Selecting it opens a file selection dialog, for which a valid GCXP application profile should be selected.

Remove... Removes the currently selected GCXP application profile from the project.

View Shows the contents of the currently selected GCXP application profile.

Figure A.18. View Application Profile Dialog

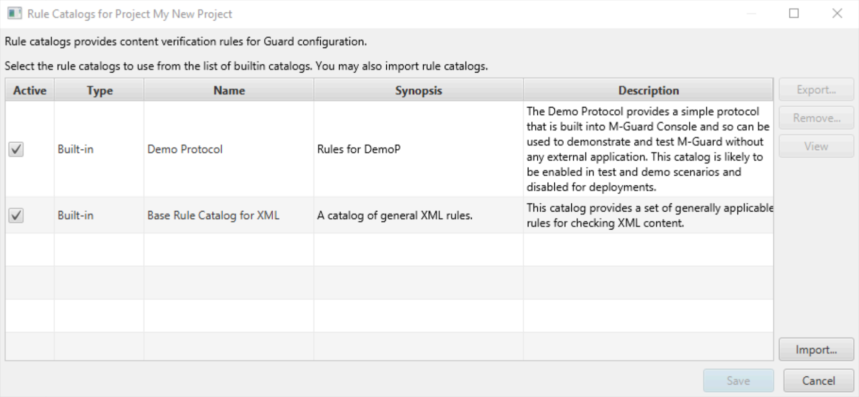


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<gcxp-application-profile xmlns="http://isode.com/gcxp/application-profile/0">
  <meta-info>
    <name>Demo Profile</name>
    <version>Demo</version>
    <description>An example profile</description>
  </meta-info>
  <traffic>
    <flow name="$request" in="$producer" out="$consumer">
      <allow type="request" format="xml">
        <verify rule="d9a86b7d-e7a3-11ea-b17a-8c85907c3a0c">
          <identity-transform>
            <xslt file="demo.xsl" data="PHhzbDpzdHlsZXNoZWV0IHZlcnNpb249IjEuM
            <xc14n algorithm="xc14n_1_1"/>
            <unicode form="nfc"/>
          </identity-transform>
          <xpath path="/descendant::comment()" min="0" max="0"/>
          <xpath path="/descendant::processing-instruction()" min="0" max="0"/>
          <xsd file="schemas/demo-explicit.xsd" data="PHhzOnNjaGVtYSB4bWxuczp
          <schematron file="demo.sch" data="PD94bWwgdmVyc2lvbj0iMS4wIj8+Cjxz
        </verify>
      </allow>
    </flow>
    <flow name="$response" in="$consumer" out="$producer">
      <allow type="response" format="nil" optional="true"/>
    </flow>
  </traffic>
  <schemas>
    <schema file="schemas/xml.xsd" data="PD94bWwgdmVyc2lvbj0iMS4wIj8+Cjw/eG1
    <schema file="schemas/sec-label-explicit.xsd" data="PD94bWwgdmVyc2lvbj0nMS4
    <schema file="schemas/sec-label-ess.xsd" data="PD94bWwgdmVyc2lvbj0iMS4wIiB
  </schemas>
```

Rule Catalogs

Opens the Rule Catalogs dialog which enables management of the rule catalogs associated with a project.

Figure A.19. Catalog Management



Rule catalogs provides content verification rules for Guard configuration.

Select the rule catalogs to use from the list of builtin catalogs. You may also import rule catalogs.

Active	Type	Name	Synopsis	Description
☑	Built-in	Demo Protocol	Rules for DemoP	The Demo Protocol provides a simple protocol that is built into M-Guard Console and so can be used to demonstrate and test M-Guard without any external application. This catalog is likely to be enabled in test and demo scenarios and disabled for deployments.
☑	Built-in	Base Rule Catalog for XML	A catalog of general XML rules.	This catalog provides a set of generally applicable rules for checking XML content.

Each rule catalog has a row in the table. Rule catalogs marked as Active are made available for Guard configuration within the project. The other fields in the column are taken from the catalog and are read only. When a catalog is selected, two options are available:

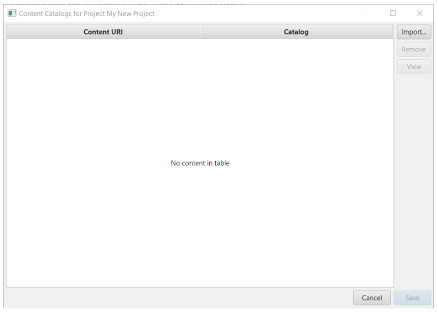
- Export - This will export the catalog to a folder of your choice. A catalog comprises a tree of folders and files, and must be exported to an empty folder.
- Remove - This removes the catalog from the project. Catalogs of type Built-in cannot be removed.

Additional rule catalogs can be imported into a project using the Import button.

Content Catalogs

Opens the Content Catalogs dialog used for managing the content catalogs for a project.

Figure A.20. Content Catalogs dialog



Content Catalogs for Project My New Project

Content URI	Catalog

No content in table

The buttons in the dialog do the following:

Import... — Opens the Import Content Catalog dialog for importing a content catalog into the project.

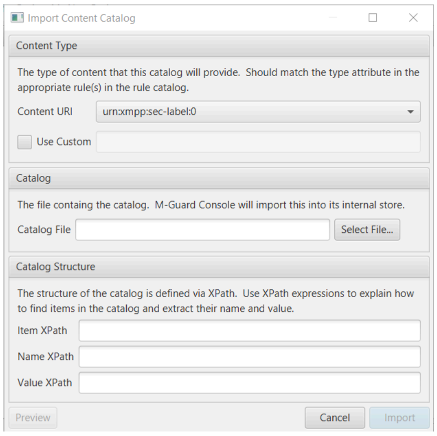
Remove — Removes the currently selected content catalog from the project.

View — Displays the contents of the currently selected content catalog.

Import Content Catalog dialog

Used for importing a content catalog into a project.

Figure A.21. Import Content Catalog dialog



Import Content Catalog

Content Type

The type of content that this catalog will provide. Should match the type attribute in the appropriate rule(s) in the rule catalog.

Content URI: urn:xmpp:sec-label:0

□ Use Custom

Catalog

The file containg the catalog. M-Guard Console will import this into its internal store.

Catalog File

Catalog Structure

The structure of the catalog is defined via XPath. Use XPath expressions to explain how to find items in the catalog and extract their name and value.

Item XPath

Name XPath

Value XPath

The sections of the dialog are outlined below:

Content

The type of content that catalog will be used for.

Content URI — The content URI used in a rule catalog to mark a rule as using this catalog. The values of the combo box are extracted from the project's configured rule catalogs.

Use Custom — Check box and text field to allow the entry of a custom string for content URI instead of using one extracted from the rule catalogs.

Catalog

The XML file containing the catalog to be imported into the project.

Catalog File Path to the the file.

Select File... Open a file selection browser for select the file.

Catalog Structure

Xpath expressions detailing how to extract and parse items from the catalog.

Item XPath Expression detailing how to extract items in the catalog.

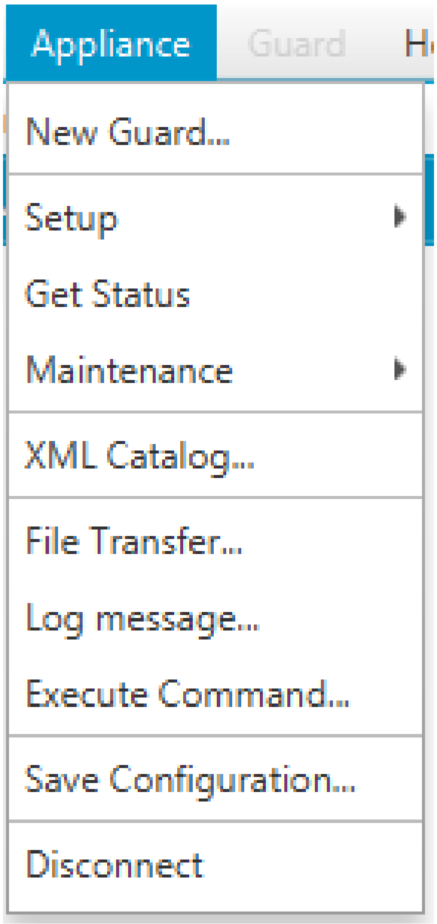
Name XPath Expression detailing how to extract a name from an item.

Value XPath Expression detailing how to extract a value from an item.

Appliance Menu

This menu is also available by right-clicking on an appliance's name. The top menu applies to the selected appliance.

Figure A.22. Appliance Menu

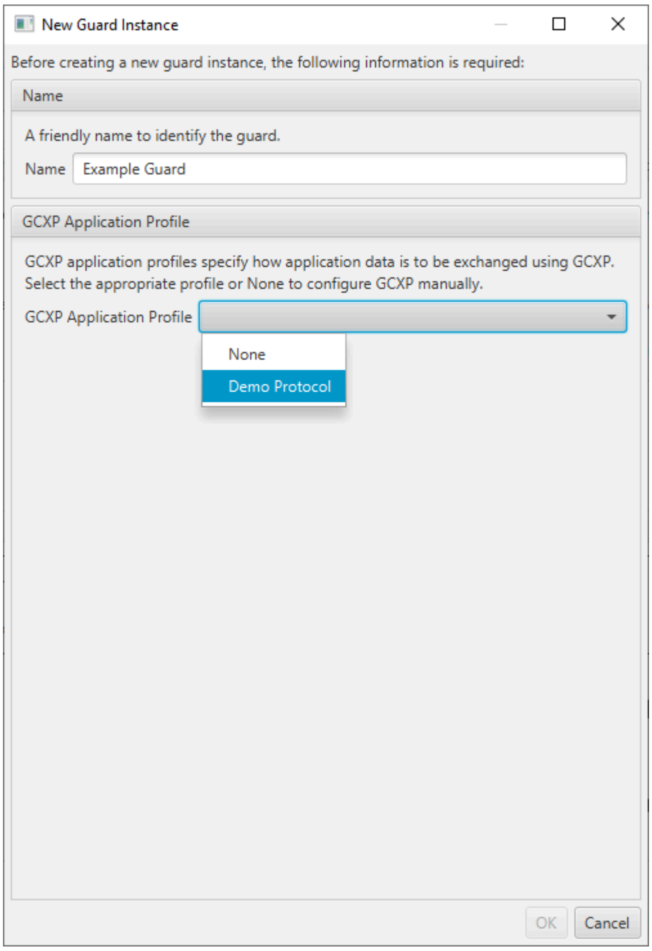


These options are described below.

New Guard

Creates a new Guard instance on the appliance

Figure A.23. Create New Guard


New Guard Instance
Before creating a new guard instance, the following information is required:
Name
A friendly name to identify the guard.
Name
GCXP Application Profile
GCXP application profiles specify how application data is to be exchanged using GCXP. Select the appropriate profile or None to configure GCXP manually.
GCXP Application Profile
None
Demo Protocol
OK
Cancel

Name: Example Guard

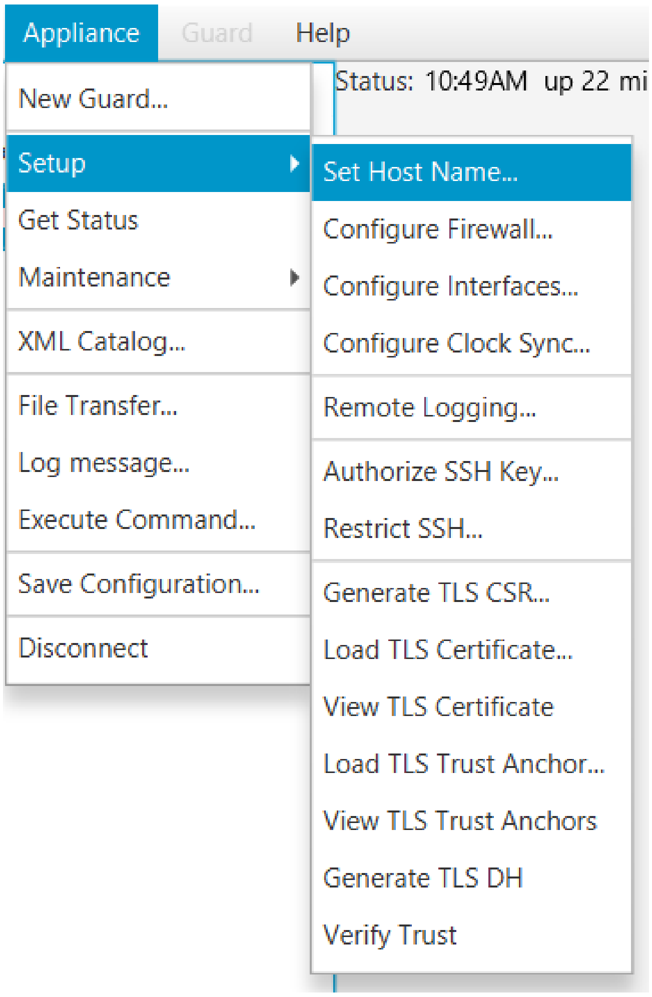
The following information is required:

Name	A descriptive name to identify the guard
GCXP Application Profile	The GCXP Application profile to use to configure the new guard, or None to manually configure it.

Setup

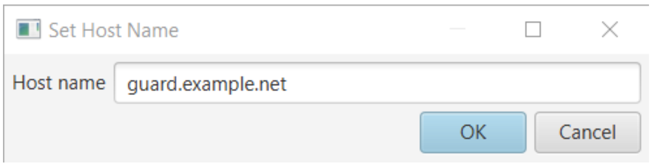
Setup raises a submenu with a number of items as shown in the following figure and are described below.

Figure A.24. Setup Menu



Set Host Name

Figure A.25. Set Host Name



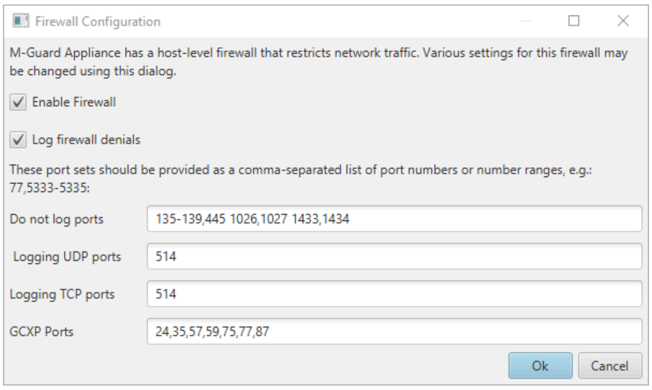
Host name guard.example.net

This sets the host name for the appliance. The name should be a fully-qualified domain name. Private, local domains, e.g. `.test`, where the appliance is not attached to any network on the Internet. However, the domain `.local` and other special domains should be avoided.

Configure Firewall

Used to configure the Guard appliance's firewall.

Figure A.26. Firewall Configuration dialog



Firewall Configuration

M-Guard Appliance has a host-level firewall that restricts network traffic. Various settings for this firewall may be changed using this dialog.

☑ Enable Firewall

☑ Log firewall denials

These port sets should be provided as a comma-separated list of port numbers or number ranges, e.g.: 77,5333-5335:

Do not log ports	135-139,445 1026,1027 1433,1434
Logging UDP ports	514
Logging TCP ports	514
GCXP Ports	24,35,57,59,75,77,87

Enable Firewall — If selected the firewall will enabled

Log firewall denials — If selected firewall denials will be logged

The following fields are port lists. Their values should consist of either TCP/UDP port numbers or ranges separated by commas or spaces:

Do not log ports — List of TCP/UDP ports for which denied incoming packets are not logged

Logging UDP ports — List of UDP ports to allow logging

Logging TCP ports — List of TCP ports to allow logging

GCXP Ports — List of ports to allow GCXP

Configure Interfaces

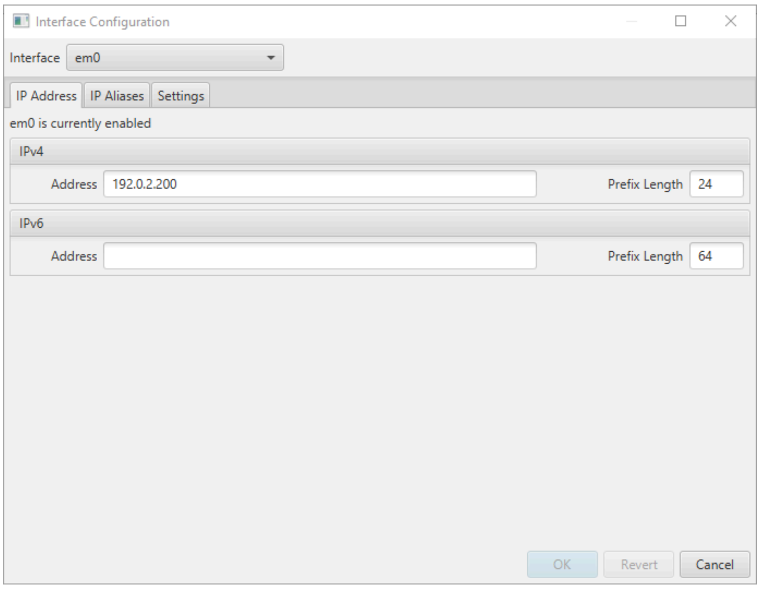
Configure Interfaces is used to manage network interfaces on the appliance.

The interface being configured is controlled via the top combo box.

Each tab provides configuration for a different aspect of the interface.

IP Address

Figure A.27. IP Address tab



Interface Configuration

Interface: em0

em0 is currently enabled

IPv4

Address: 192.0.2.200 Prefix Length: 24

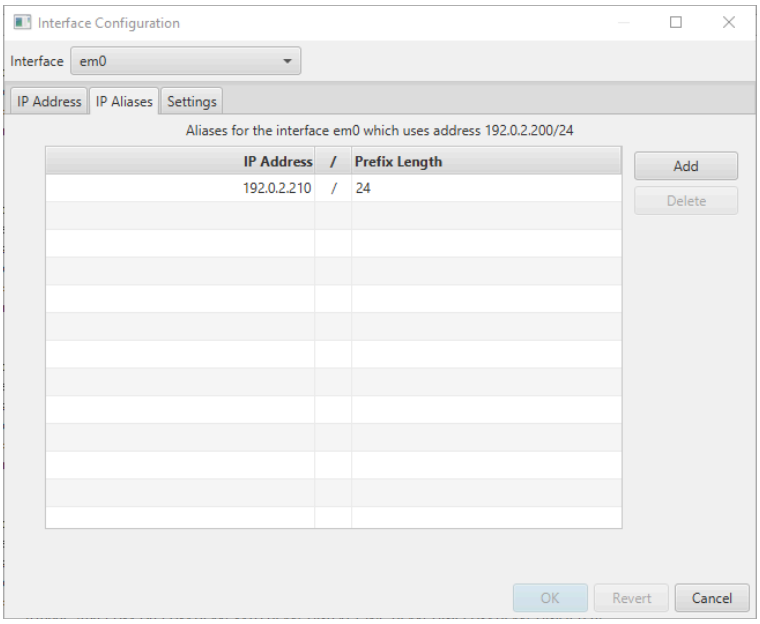
IPv6

Address: Prefix Length: 64

This tab allows configuration of the primary IPv4 or IPv6 address for each enabled network interface on the appliance. The network prefix length must also be configured.

IP Aliases

Figure A.28. IP Aliases tab



Interface: em0

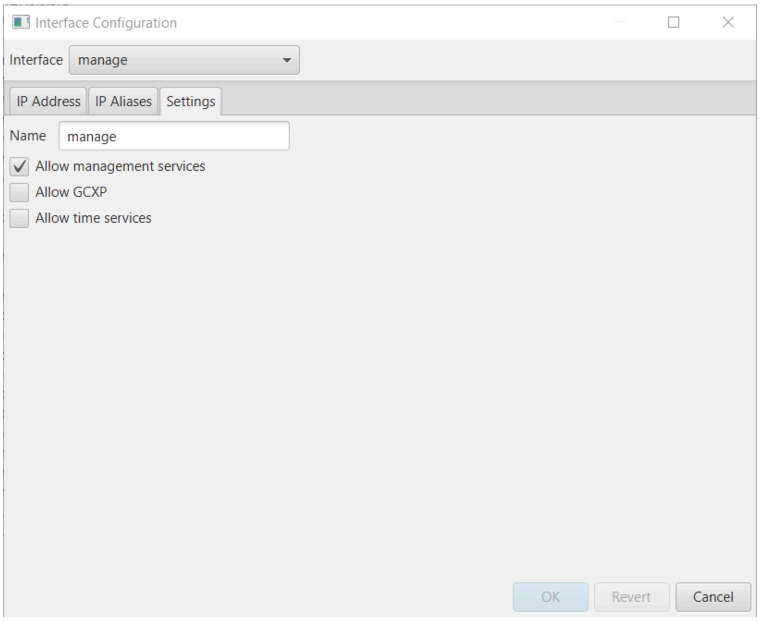
Aliases for the interface em0 which uses address 192.0.2.200/24

IP Address	/	Prefix Length
192.0.2.210	/	24

This tab allows additional IP addresses to be assigned to a network interface.

Settings

Figure A.29. Settings tab



Interface: manage

Name: manage

☑ Allow management services

☐ Allow GCXP

☐ Allow time services

This tab allows configuration of additional settings for the interface

Name	Can be used to assign an alternative name to the interface. This can be used to assign names which reflect the function

of an interface. These names will be visible in the Guard configuration, which can help to ensure correct use of interfaces.

Allow management services
If checked the interface can be used for management of M-Guard Appliances by M-Guard Console.

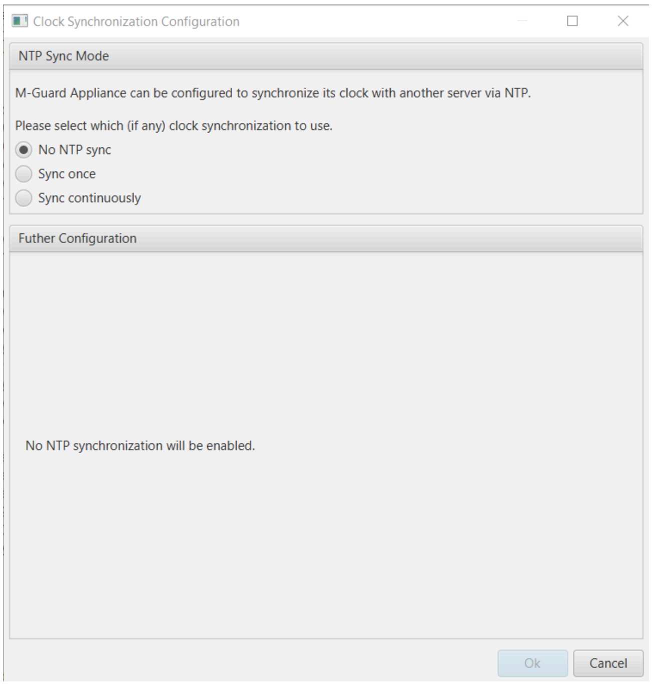
Allow GCXP
If checked GCXP can be used on the interface.

Allow time services
If checked time services (e.g., NTP) can be used on the interface, allowing the appliance to both send NTP queries and respond to NTP queries received on the interface. Note that NTP queries are also allowed management interfaces.

Configure Clock Sync

This allows the use of an Network Time Protocol (NTP) server to synchronize the appliance's clock.

Figure A.30. Clock Synchronization Configuration dialog



M-Guard Appliance can be configured to synchronize its clock with another server via NTP.

Please select which (if any) clock synchronization to use.

● No NTP sync
○ Sync once
○ Sync continuously

No NTP synchronization will be enabled.

The mode of synchronization can be set to one of: No NTP syncing, Sync once or Sync continuously.

Note

For virtual appliances, the hypervisor’s clock synchronization must be disabled for Sync once and Sync continuously to work properly.

When the virtual appliance is an Oracle VirtualBox guest system, M-Guard Console will be able to disable this automatically. For other platforms this will need to be disabled manually. Please consult the hypervisor documentation for information on how this can be done.

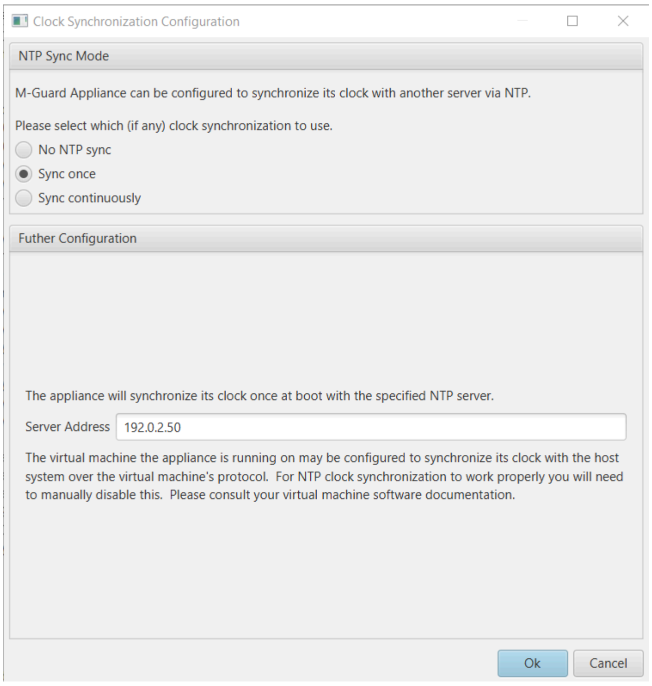
No NTP Syncing

The appliance clock will not be synchronized using NTP.

Sync once

The appliance will synchronize its clock once at boot with the specified NTP server. No further synchronization will occur.

Figure A.31. Sync once



M-Guard Appliance can be configured to synchronize its clock with another server via NTP.

Please select which (if any) clock synchronization to use.

○ No NTP sync
● Sync once
○ Sync continuously

The appliance will synchronize its clock once at boot with the specified NTP server.

Server Address: 192.0.2.50

The virtual machine the appliance is running on may be configured to synchronize its clock with the host system over the virtual machine's protocol. For NTP clock synchronization to work properly you will need to manually disable this. Please consult your virtual machine software documentation.

Server Address The IP address of the NTP server to sync to.

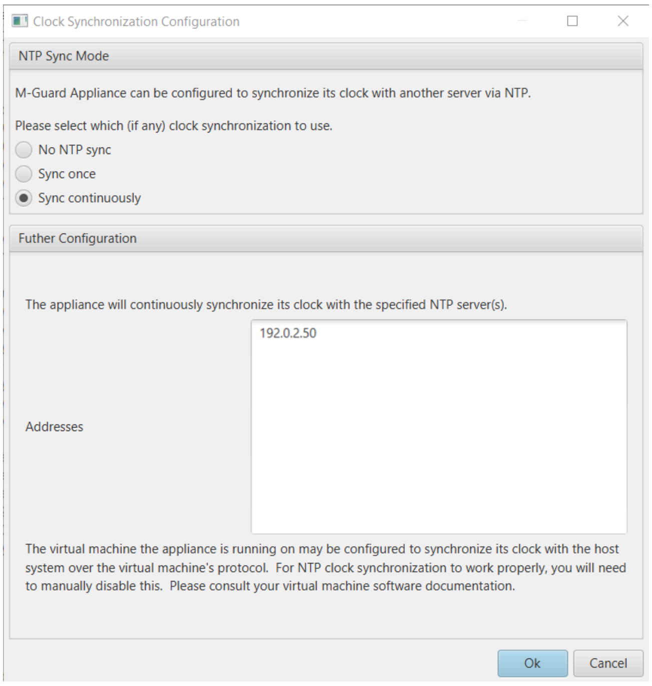
Note

The specified server must be on a network attached to an interface belonging to either the manage or time group. See the section called “Configure Interfaces” the section called “Settings”.

Sync continuously

The appliance will continuously synchronize its clock with the specified NTP server(s).

Figure A.32. Sync continuously



Clock Synchronization Configuration

NTP Sync Mode

M-Guard Appliance can be configured to synchronize its clock with another server via NTP.

Please select which (if any) clock synchronization to use.

○ No NTP sync

○ Sync once

● Sync continuously

Futher Configuration

The appliance will continuously synchronize its clock with the specified NTP server(s).

Addresses: 192.0.2.50

The virtual machine the appliance is running on may be configured to synchronize its clock with the host system over the virtual machine's protocol. For NTP clock synchronization to work properly, you will need to manually disable this. Please consult your virtual machine software documentation.

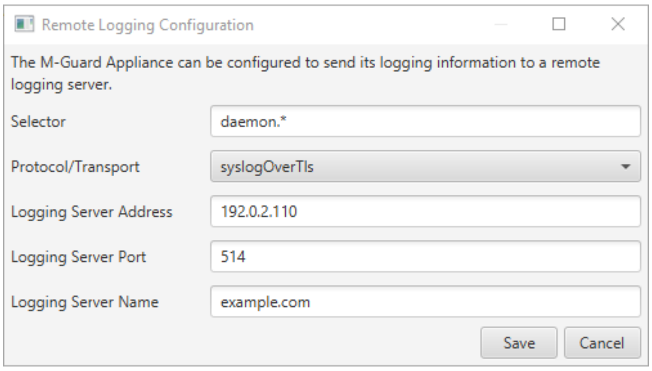
Addresses The IP address(es) of the NTP server(s) to synchronize to.

Note

Each specified server must be on a network attached to an interface belonging to either the manage or time group. See the section called “Configure Interfaces” the section called “Settings”.

Remote Logging

Figure A.33. Remote Logging Configuration dialog



Remote Logging Configuration

The M-Guard Appliance can be configured to send its logging information to a remote logging server.

Selector: daemon.*

Protocol/Transport: syslogOverTls

Logging Server Address: 192.0.2.110

Logging Server Port: 514

Logging Server Name: example.com

The Remote Logging Configuration allows configuration of the Guard appliance remote logging facility. Allow it to send syslog or RELP log messages to a remote server.

Selector rsyslog style selector that select which messages are logged to the remote server.

Protocol/Transport The Protocol/Transport method used to deliver the log message to the remote server.

Logging Server Address The IP address of the remote server.

Logging Server Port

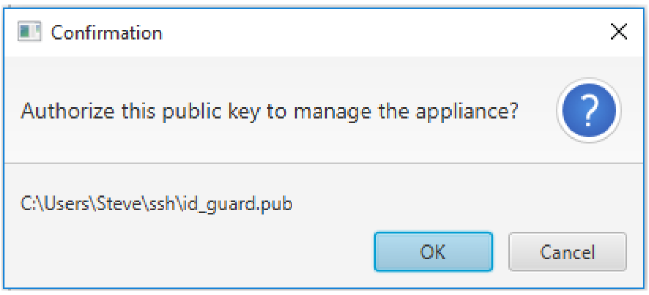
Port on the remote server to send the log messages to

Logging Server Name

Name used by the remote logging server to identify itself in its TLS certificate. Only used if the Protocol/Transport method is over TLS

Authorize SSH Key

Figure A.34. Authorise SSH Key



Confirmation

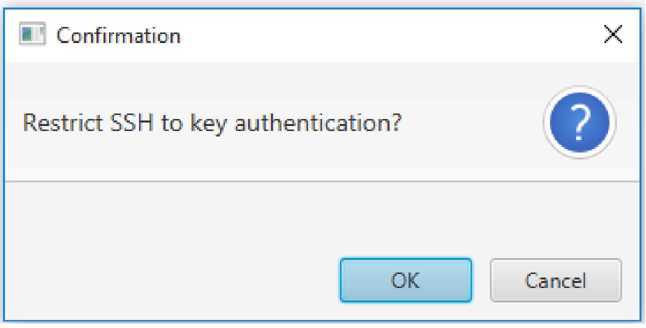
Authorize this public key to manage the appliance?

C:\Users\Steve\ssh\id_guard.pub

The Authorize SSH Key command instructs the appliance to authorize the SSH key on the local system. This will usually be used on Appliance setup, where initial authorization is made by authenticating with the root password on the appliance.

Restrict SSH

Figure A.35. Restrict SSH



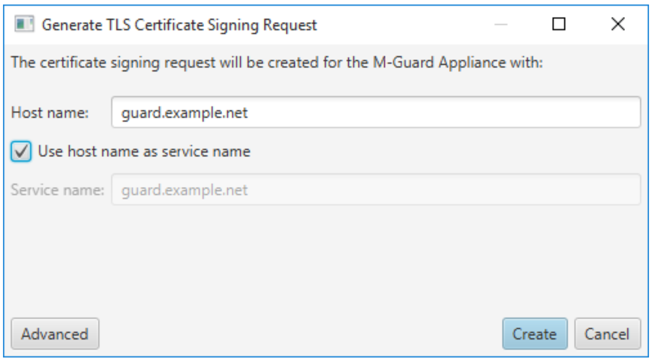
Confirmation

Restrict SSH to key authentication?

Restrict SSH configures the appliance so that it will only authenticate SSH clients using SSH keys. This precludes use of password authentication for remote access to the appliance. This restriction is recommended for an operational system.

Generate TLS CSR

Figure A.36. Generate TLS CSR



Generate TLS Certificate Signing Request

The certificate signing request will be created for the M-Guard Appliance with:

Host name: guard.example.net

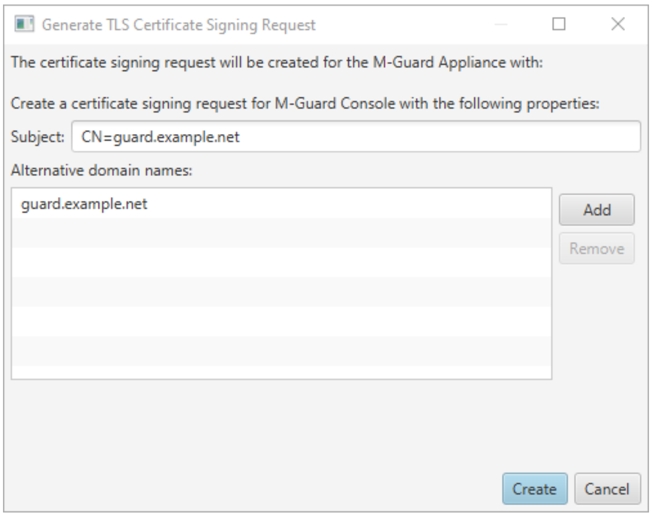
☑ Use host name as service name

Service name: guard.example.net

The Generate TLS CSR command causes the appliance to generate a public/private key pair, with the private key held by the appliance. The appliance then generates a certificate signing request (CSR), which M-Guard Console stores on the management desktop in a file of your choice.

Selecting the Advanced brings up an advanced view that enabled greater configuration of the generated CSR

Figure A.37. Generate TLS CSR - Advanced View



Generate TLS Certificate Signing Request

The certificate signing request will be created for the M-Guard Appliance with:

Create a certificate signing request for M-Guard Console with the following properties:

Subject: CN=guard.example.net

Alternative domain names:

guard.example.net

The CSR must be sent to a certification authority (CA) to provide a TLS certificate. The CA should provide the certificate in a PEM file containing the certificate and full chain of signing certificates, as well as a certificate for the trust anchor containing its self signed certificate.

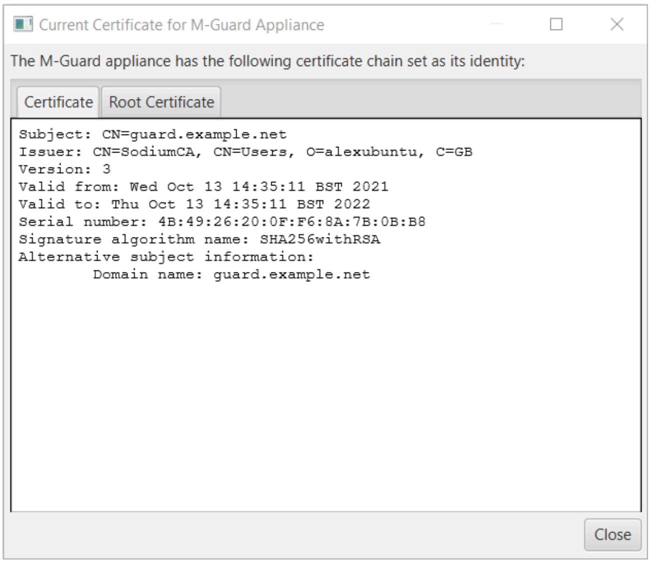
Load TLS Certificate

The Load TLS Certificate item is used to install a TLS certificate. You will be prompted to select a file that contains the certificate and the full chain of signing certificates. The file should be PEM encoded with the certificates ordered *end-entity* certifRate limiting is accomplished by enforcing a minimum time between consecutive messages through the Guard.icate first to r*oot certificate authority* certificate last.

View TLS Certificate

Shows the TLS certificate that the Guard appliance is using.

Figure A.38. Current Certificate for M-Guard Appliance dialog



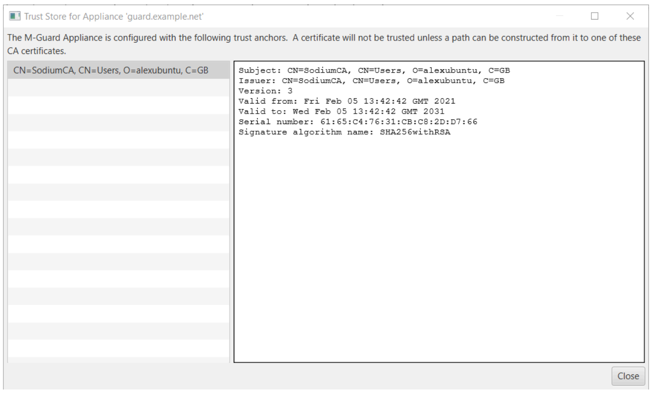
The M-Guard appliance has the following certificate chain set as its identity:

```
Subject: CN=guard.example.net
Issuer: CN=SodiumCA, CN=Users, O=alexubuntu, C=GB
Version: 3
Valid from: Wed Oct 13 14:35:11 BST 2021
Valid to: Thu Oct 13 14:35:11 BST 2022
Serial number: 4B:49:26:20:0F:F6:8A:7B:0B:B8
Signature algorithm name: SHA256withRSA
Alternative subject information:
        Domain name: guard.example.net
```

View TLS Trust Anchors

Shows the TLS anchors that the Guard appliance is using for verifying TLS certificates.

Figure A.39. Trust Store for Appliance dialog


Trust Store for Appliance 'guard.example.net'
CN=SodiumCA, CN=Users, O=alexubuntu, C=GB
Close

The M-Guard Appliance is configured with the following trust anchors. A certificate will not be trusted unless a path can be constructed from it to one of these CA certificates.

```
Subject: CN=SodiumCA, CN=Users, O=alexubuntu, C=GB
Issuer: CN=SodiumCA, CN=Users, O=alexubuntu, C=GB
Version: 3
Valid from: Fri Feb 05 13:42:42 GMT 2021
Valid to: Wed Feb 05 13:42:42 GMT 2031
Serial number: 61:65:C4:76:31:CB:C8:2D:D7:66
Signature algorithm name: SHA256withRSA
```

Load TLS Trust Anchor

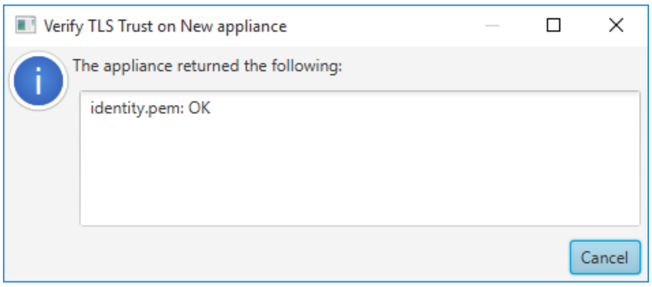
The Load TLS Trust Anchor... command is used to install additional trust anchors on the appliance. You will be prompted to select a file that contains one or more certificate authority certificates for the trust anchors. The file should be PEM encoded.

Generate TLS DH

Generates a set of DH parameters for the appliance.

Verify Trust

Figure A.40. Verify-Trust



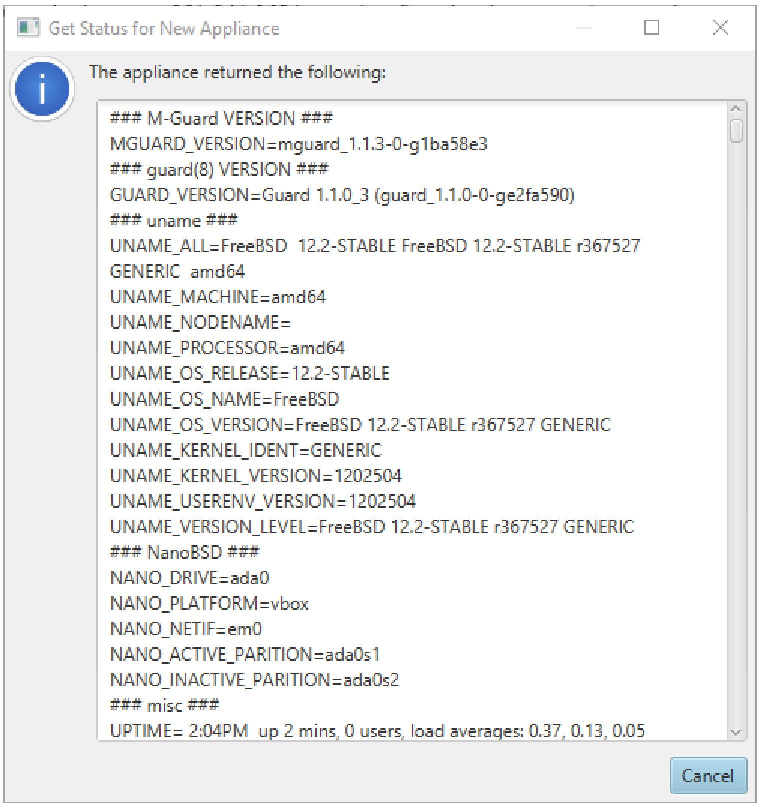
The appliance returned the following:

```
identity.pem: OK
```

The Verify Trust command checks whether the appliance's TLS certificate is trusted by the appliance's trusted anchors.

Get Status

Figure A.41. Get Status



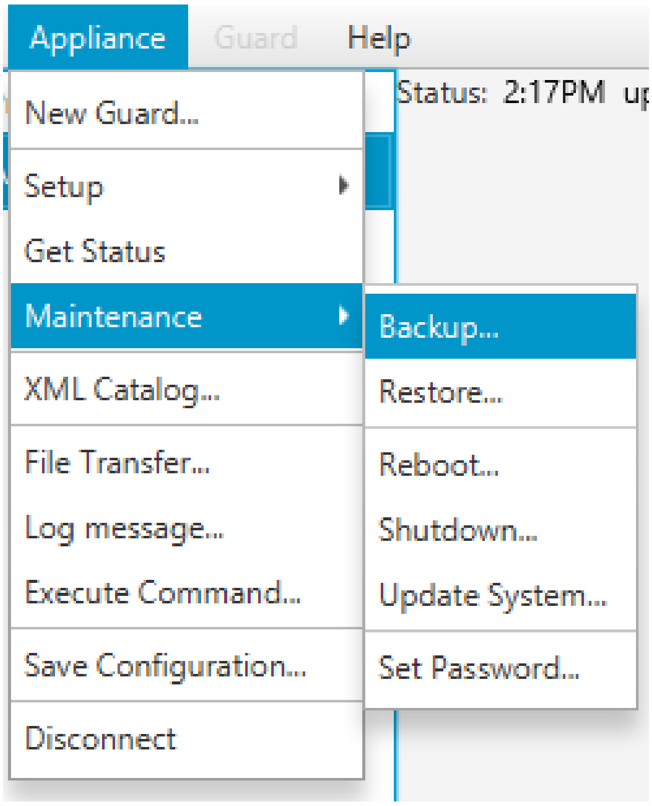
The appliance returned the following:

```
### M-Guard VERSION ###
MGUARD_VERSION=mguard_1.1.3-0-g1ba58e3
### guard(8) VERSION ###
GUARD_VERSION=Guard 1.1.0_3 (guard_1.1.0-0-ge2fa590)
### uname ###
UNAME_ALL=FreeBSD  12.2-STABLE FreeBSD 12.2-STABLE r367527
GENERIC  amd64
UNAME_MACHINE=amd64
UNAME_NODENAME=
UNAME_PROCESSOR=amd64
UNAME_OS_RELEASE=12.2-STABLE
UNAME_OS_NAME=FreeBSD
UNAME_OS_VERSION=FreeBSD 12.2-STABLE r367527 GENERIC
UNAME_KERNEL_IDENT=GENERIC
UNAME_KERNEL_VERSION=1202504
UNAME_USERENV_VERSION=1202504
UNAME_VERSION_LEVEL=FreeBSD 12.2-STABLE r367527 GENERIC
### NanoBSD ###
NANO_DRIVE=ada0
NANO_PLATFORM=vbox
NANO_NETIF=em0
NANO_ACTIVE_PARITION=ada0s1
NANO_INACTIVE_PARITION=ada0s2
### misc ###
UPTIME= 2:04PM  up 2 mins, 0 users, load averages: 0.37, 0.13, 0.05
```

The Get Status command shows information about the appliance. This may be helpful to provide detailed appliance information to Isode Support.

Maintenance

Figure A.42. Maintenance



Maintenance has a number of menu options, described in the following sections.

Back Up

The Back Up command saves the *last saved* appliance configuration, including configuration of all Guard instances, to a file. It prompts to select a file.

Note that the backup files contain a copy of the appliance's password database as well as SSH and TLS private keys and other potentially sensitive configuration information and, hence, should be appropriately secured.

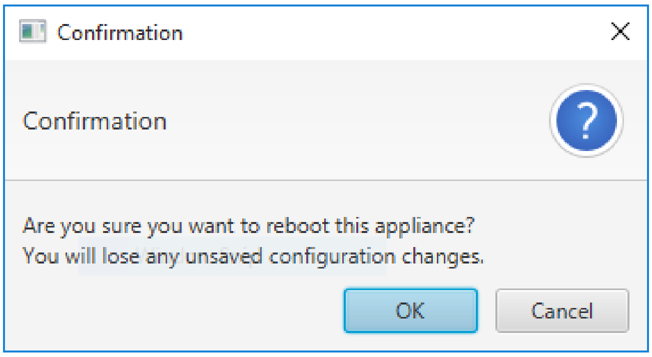
Restore

The Restore command restores the appliance configuration from a file created by the Back Up command. It prompts to select a file.

It is recommended you backup your configuration before performing this operation.

Reboot

Figure A.43. Reboot



Confirmation

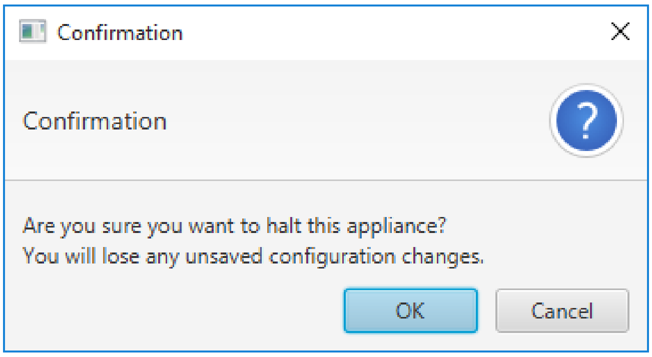
Confirmation

Are you sure you want to reboot this appliance?
You will lose any unsaved configuration changes.

The Reboot command causes the appliance to perform an orderly reboot of itself without saving changes to the live-system configuration.

Shut Down

Figure A.44. Shut Down



Confirmation

Confirmation

Are you sure you want to halt this appliance?
You will lose any unsaved configuration changes.

The Shut Down command causes the appliance to perform an orderly shutdown without saving changes to the live-system configuration.

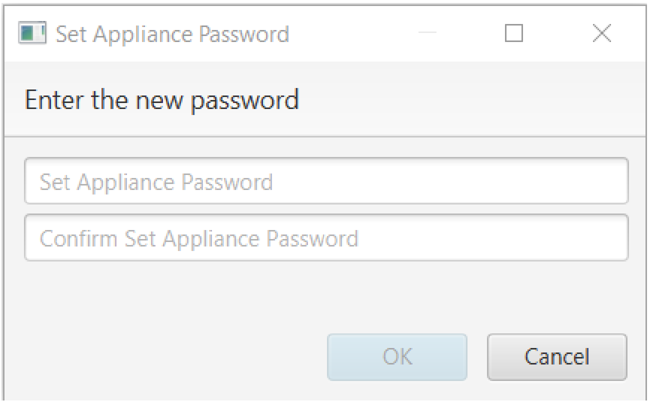
Update System

The Update System command is used to update the appliance. The dialog prompts for an appliance update system image file (compressed image as distributed by Isode). This updates the appliance and restarts it. The configuration of the appliance is unchanged. When upgrading, it is recommended that you upgrade to the latest version of Guard Appliance software using the latest version of M-Guard Console.

It is recommended you backup your configuration before performing this operation.

Set Password

Figure A.45. Set Password



Set Appliance Password

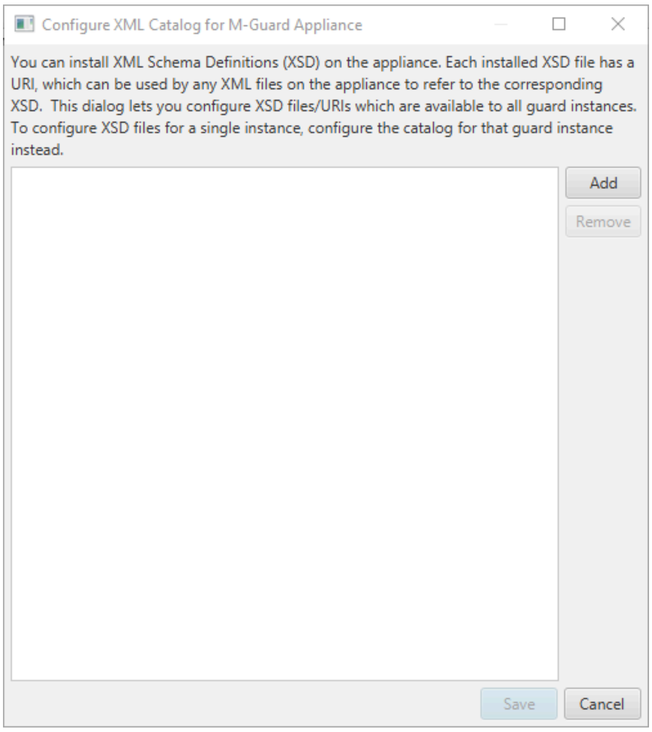
Enter the new password

The Set Password command changes the Administration (root) password.

XML Catalog

Opens the Configure XML Catalog dialog for the M-Guard appliance. This allows the management of the M-Guard appliance's XML catalog.

Figure A.46. Configure XML Catalog for the M-Guard appliance



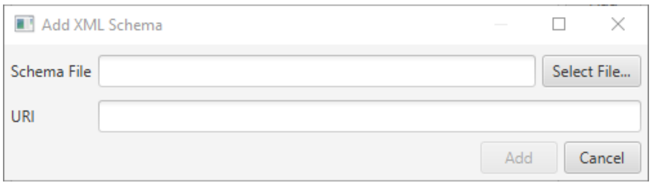
Configure XML Catalog for M-Guard Appliance

You can install XML Schema Definitions (XSD) on the appliance. Each installed XSD file has a URI, which can be used by any XML files on the appliance to refer to the corresponding XSD. This dialog lets you configure XSD files/URIs which are available to all guard instances. To configure XSD files for a single instance, configure the catalog for that guard instance instead.

Add

Opens a dialog for selecting an XML Schema Definition (XSD) file and uploading it to the M-Guard appliance's XML catalog.

Figure A.47. Add XML Schema dialog

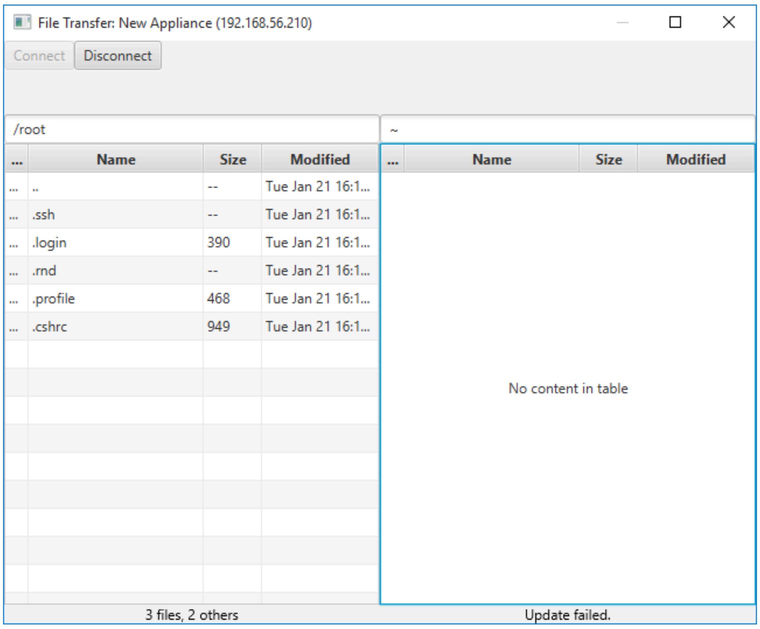


Remove

Removes the selected entry from the M-Guard appliance's XML catalog.

File Transfer

Figure A.48. File Transfer



/root

...	Name	Size	Modified
...	..	--	Tue Jan 21 16:1...
...	.ssh	--	Tue Jan 21 16:1...
...	.login	390	Tue Jan 21 16:1...
...	.rnd	--	Tue Jan 21 16:1...
...	.profile	468	Tue Jan 21 16:1...
...	.cshrc	949	Tue Jan 21 16:1...

3 files, 2 others

~

...	Name	Size	Modified

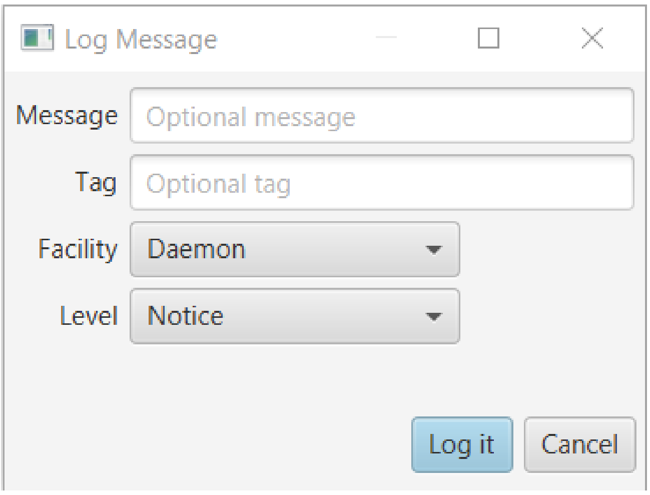
No content in table

Update failed.

The File Transfer command provides an easy way to move files between the appliance and the host running M-Guard Console.

Log Message

Figure A.49. Log Message



Log Message

Message: Optional message

Tag: Optional tag

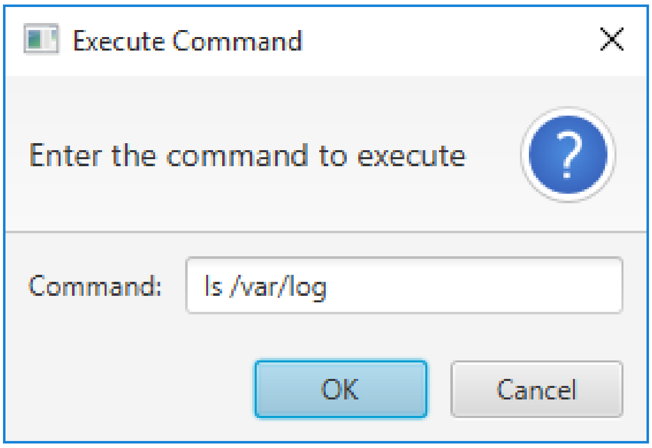
Facility: Daemon

Level: Notice

The Log Message command causes a syslog event to be generated by the appliance with the specified parameters. This may be helpful for diagnosing syslog configuration issues.

Execute Command

Figure A.50. Execute Command



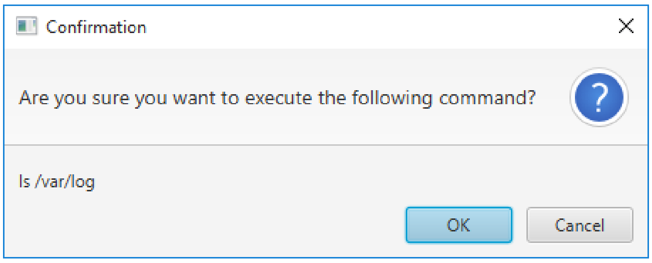
Execute Command

Enter the command to execute

Command: ls /var/log

The Execute Command command allows any command to be run on the appliance. This capability should be used with care, typically under guidance from Isode support. It can be used to run simple commands on the appliance. It, however, is not a replacement for the appliance command line accessible from the appliance console or via a standalone SSH client. The example command above gives the following confirmation.

Figure A.51. Execute Command Confirmation



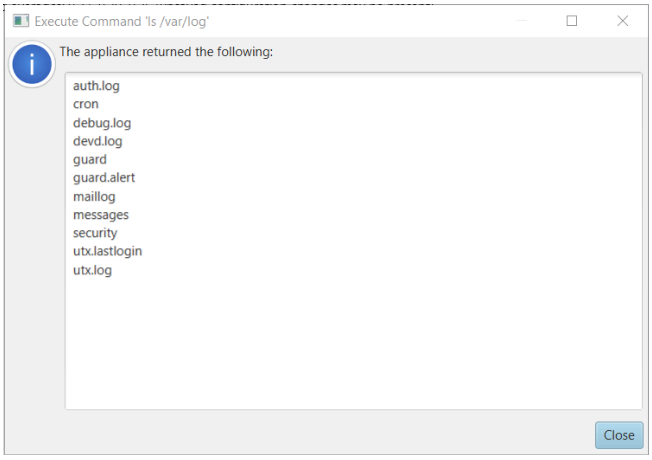
Confirmation

Are you sure you want to execute the following command?

ls /var/log

And then the following result.

Figure A.52. Execute Command Result

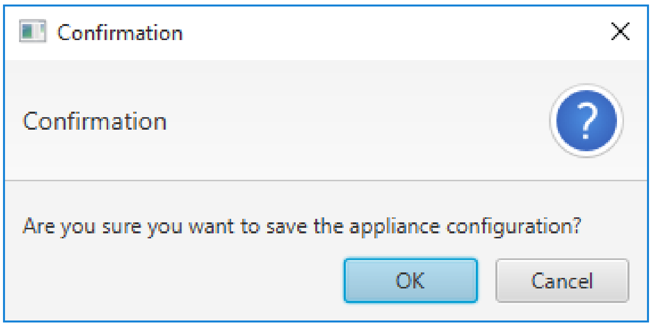


The appliance returned the following:

```
auth.log
cron
debug.log
devd.log
guard
guard.alert
maillog
messages
security
utx.lastlogin
utx.log
```

Save Configuration

Figure A.53. Save Configuration



Are you sure you want to save the appliance configuration?

When changes are made to the appliance they are applied immediately to the live configuration. However, they are not saved and will be lost when the appliance is restarted. This enables configuration changes to be tested safely. If a change breaks appliance operation, it can be reverted by restarting the appliance. The Save Configuration command applies configuration changes so that they are permanent and will not be lost on restart.

Disconnect / Connect

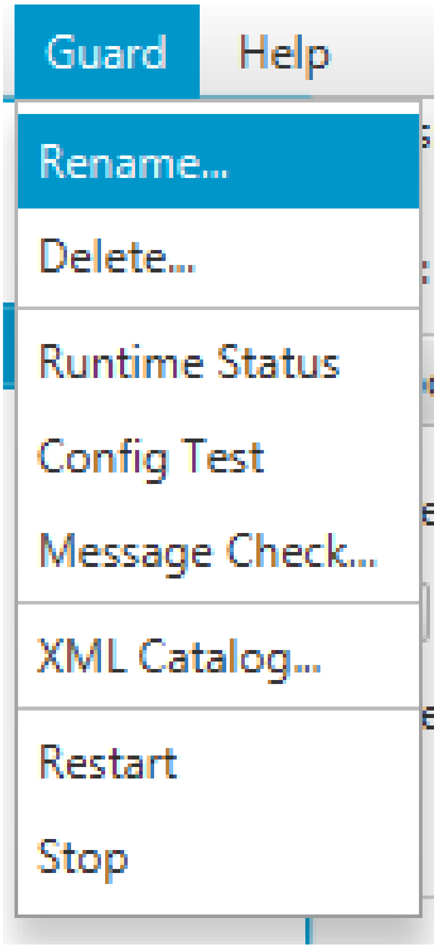
If M-Guard Console is connected to the appliance, the Disconnect command is offered. When selected, the console will disconnect the console from the appliance.

If M-Guard Console is not connected to the appliance, the Connect command is offered. When selected, the console will attempt to connect to the appliance. The console will prompt for the appliance's administration password if needed.

Guard Menu

This menu is also available by right-clicking on Guard.

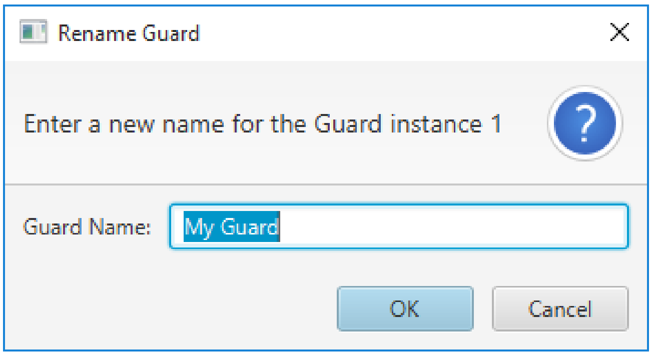
Figure A.54. Guard Menu



The Guard menu options are described in the following sections.

Rename

Figure A.55. Rename Guard



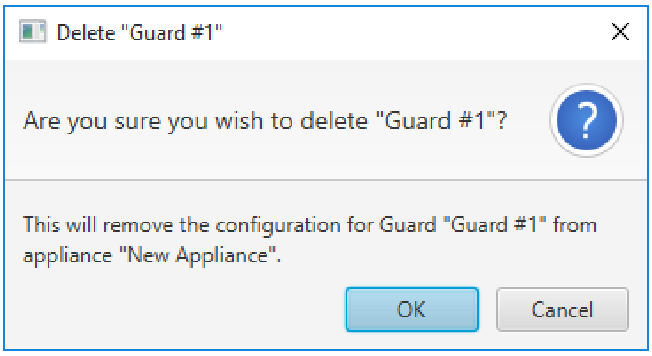
Enter a new name for the Guard instance 1

Guard Name: My Guard

The Rename command allows the name of a Guard to be changed.

Delete

Figure A.56. Delete Guard



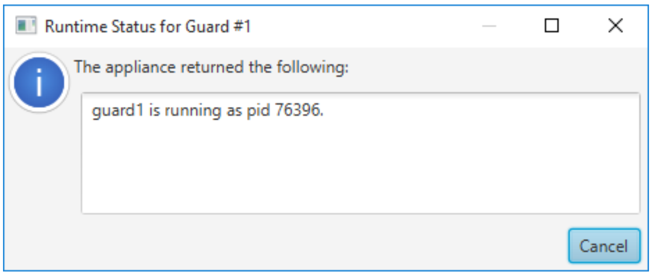
Are you sure you wish to delete "Guard #1"?

This will remove the configuration for Guard "Guard #1" from appliance "New Appliance".

The Delete command deletes the selected Guard.

Runtime Status

Figure A.57. Runtime Status



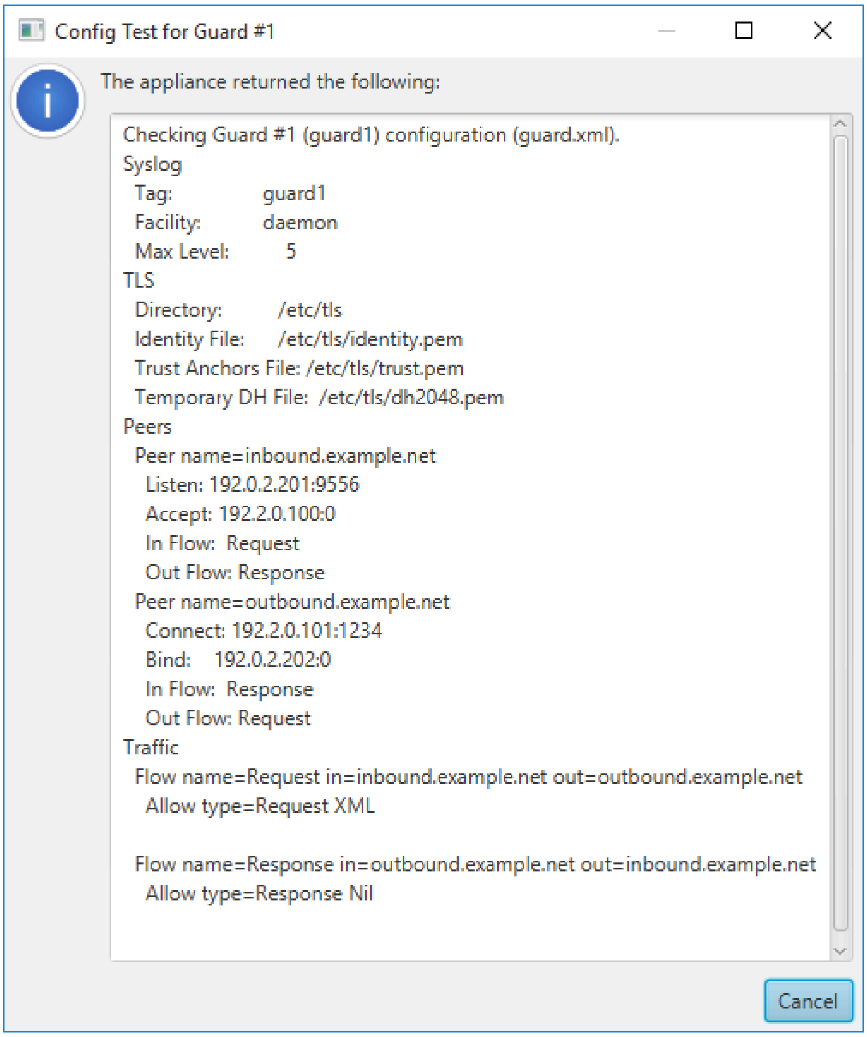
The appliance returned the following:

```
guard1 is running as pid 76396.
```

The Runtime Status command returns the current status of the Guard. This is also shown on the primary Guard display described later.

Config Test

Figure A.58. Config Test



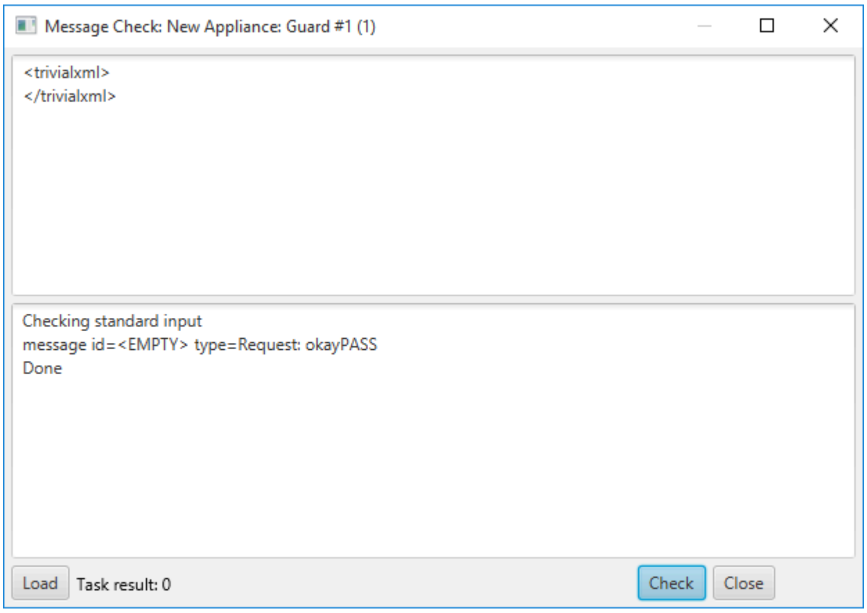
```
Checking Guard #1 (guard1) configuration (guard.xml).
Syslog
  Tag:            guard1
  Facility:       daemon
  Max Level:        5
TLS
  Directory:        /etc/tls
  Identity File:    /etc/tls/identity.pem
  Trust Anchors File: /etc/tls/trust.pem
  Temporary DH File:  /etc/tls/dh2048.pem
Peers
  Peer name=inbound.example.net
    Listen: 192.0.2.201:9556
    Accept: 192.2.0.100:0
    In Flow:  Request
    Out Flow: Response
  Peer name=outbound.example.net
    Connect: 192.2.0.101:1234
    Bind:    192.0.2.202:0
    In Flow:  Response
    Out Flow: Request
Traffic
  Flow name=Request in=inbound.example.net out=outbound.example.net
    Allow type=Request XML

  Flow name=Response in=outbound.example.net out=inbound.example.net
    Allow type=Response Nil
```

The Config Test command returns a summary of the Guard's configuration or, if the configuration is invalid, an indication of why it is not valid.

Message Check

Figure A.59. Message Check



```
<trivialxml>
</trivialxml>
```

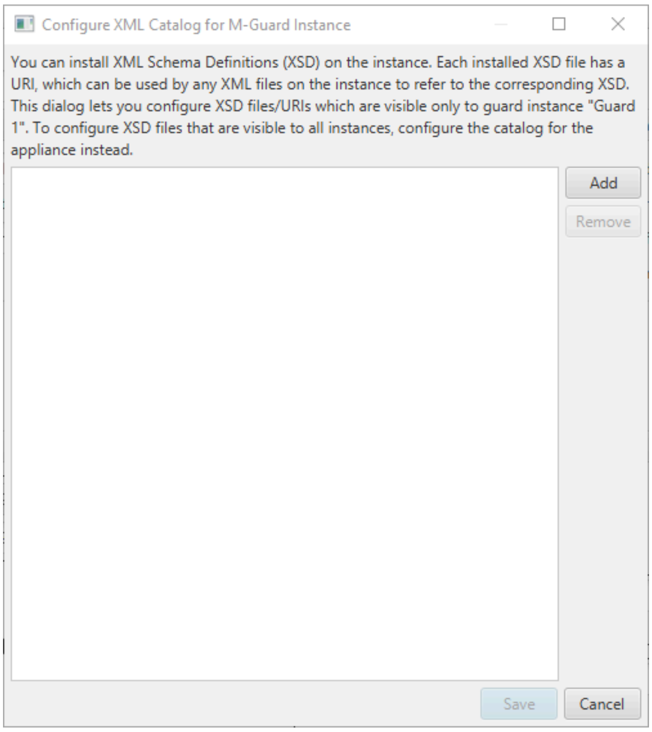
```
Checking standard input
message id=<EMPTY> type=Request: okayPASS
Done
```

The Message Check command allows the operator to check whether a Guard, as configured, would accept or reject some content. If the configuration is invalid, an indication of why it is not valid will be provided. The above example shows that the Guard accepts the content provided, as indicated by PASS.

XML Catalog

Opens the Configure XML Catalog dialog for the Guard instance. This allows the management of the Guard instance's XML catalog.

Figure A.60. Configure XML Catalog for the Guard instance



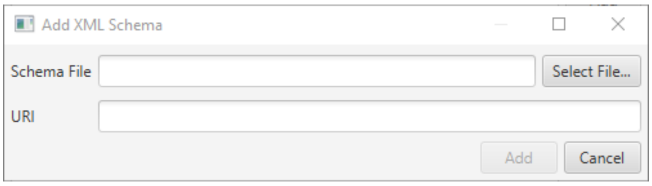
Configure XML Catalog for M-Guard Instance

You can install XML Schema Definitions (XSD) on the instance. Each installed XSD file has a URI, which can be used by any XML files on the instance to refer to the corresponding XSD. This dialog lets you configure XSD files/URIs which are visible only to guard instance "Guard 1". To configure XSD files that are visible to all instances, configure the catalog for the appliance instead.

Add

Opens a dialog for selecting an XML Schema Definition (XSD) file and uploading it to the Guard instance's XML catalog.

Figure A.61. Add XML Schema dialog



Remove

Removes the selected entry from the Guard instance's XML catalog.

Restart

The Restart command restarts a running Guard instance.

Stop

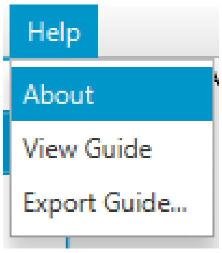
The Stop command stops a running Guard instance.

Start

The Start command starts a stopped Guard instance.

Help Menu

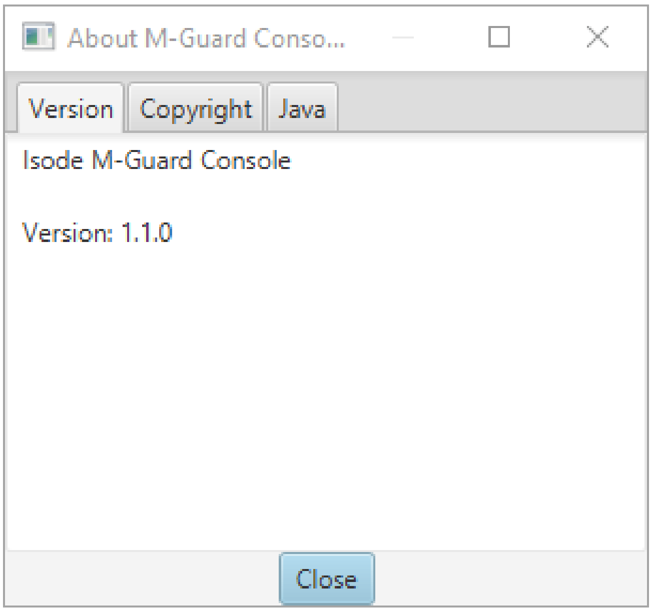
Figure A.62. Help Menu



About

Shows a dialog giving information about M-Guard Console.

Figure A.63. About M-Guard Console



Isode M-Guard Console

Version: 1.1.0

The tabs contain the following information:

Version Version information for M-Guard Console

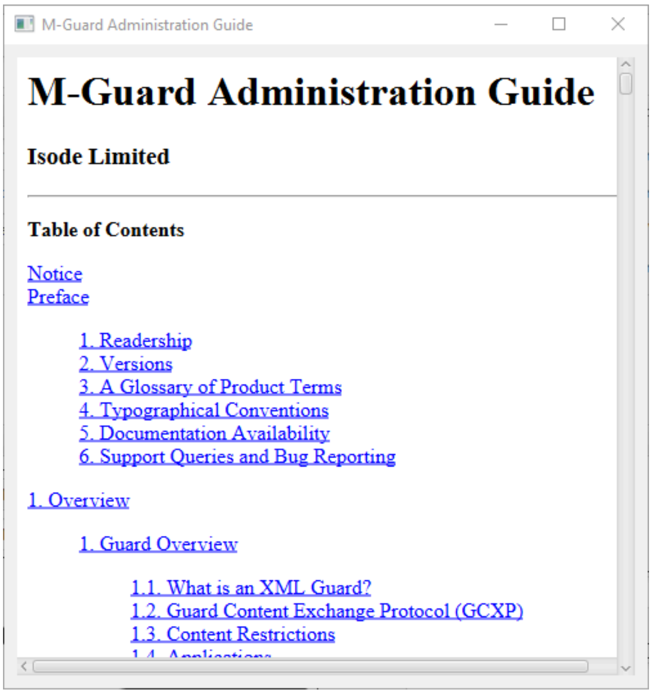
Copyright Copyright information for M-Guard Console

Java Information on the version of Java being used to run M-Guard Console

View Guide

Open a dialog to show a HTML version of this administration guide.

Figure A.64. M-Guard Administration Guide dialog



M-Guard Administration Guide

Isode Limited

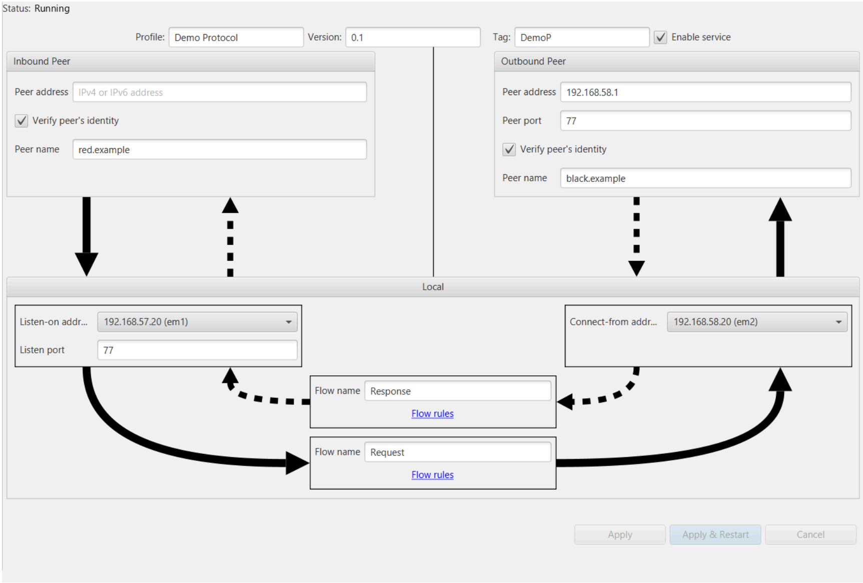
Table of Contents

Export Guide

Exports a PDF copy of this administration guide to a selected location.

Guard Configuration

Figure A.65. Guard Configuration



Status: Running

Profile: Demo Protocol Version: 0.1 Tag: DemoP ☑ Enable service

Inbound Peer

Peer address IPv4 or IPv6 address

☑ Verify peer's identity

Peer name red.example

Outbound Peer

Peer address 192.168.58.1

Peer port 77

☑ Verify peer's identity

Peer name black.example

Local

Listen-on addr... 192.168.57.20 (em1)

Listen port 77

Connect-from addr... 192.168.58.20 (em2)

Flow name Response

Flow rules

Flow name Request

Flow rules

When a Guard instance is selected in M-Guard Console, a configuration and status pane is shown, as in the screenshot above.

The operational status of the guard is shown (Running or Stopped).

The guard service can be enabled or disabled. When enabled, the operating system will start the service upon boot. When disabled, the Guard will not be started automatically. Disabled guards may be started manually.

The fields in the form may be used to configure the guard as outlined below:

Outline of form

Top Section

Profile
The name of the GCXP application profile used to create this Guard instance. This field can not be edited.

Version
The version of the GCXP application profile used to create this Guard instance. This field can not be edited.

Tag
A (typically short) string used in syslog to identify this Guard instance

Enable Service
A flag indicating if the Guard instance should be enabled on the appliance

Inbound Side

This consists of the Inbound Peer pane, which specifies information about the GCXP peer that is expected to connect to the Guard instance on the input side and the section immediately below it that specifies how the Guard instance is to listen to connections from on the input side.

Peer Address
The IP address that the peer will connect from. This is an optional field. When specified the Guard instance will only accept connections from this address

Verify peer's identity
Check box indicating that the Guard instance is to verify the inbound peer's identity using TLS authentication. It is recommended that be checked.

Peer name
The domain name of the inbound peer. This is used in verification of the peer's identity when enabled. It is also used in event logging.

Listen-on address
The IP address the Guard instance is to listen on for inbound connections. It is recommended that this be set to the Guard instance's input side IP address. If not set, the Guard instance will listen for connections on all interfaces.

Listen port
The TCP port number the Guard instance is to listen on for input side connections.

Outbound Side

This consists of the Outbound Peer pane, which specifies information about the GCXP peer that the Guard instance will connect to on the output side; and the section immediately below it that specifies how the Guard instance will make outbound connections.

Peer address
The IP address of the GCXP peer that Guard should connect to

Peer port
The TCP port number that the Guard instance will connect to.

Verify peer's Identity
Check box indicating that the Guard instance is to verify the outbound peer's identity using TLS authentication. It is recommended that be checked.

Peer name

The domain name of the outbound peer. This is used in verification of the peer's identity when enabled. It is also used in event logging.

Connect-from address

The IP address the Guard instance is to use to make its outbound connections from. It is recommended that this be set to the Guard instance's output side IP address. If not set, the Guard instance will leave the choice of which IP address to use to the appliance.

Flows

These are the two panes at the bottom. These are for configuring the two data flows. Inbound to outbound at the bottom, and outbound to inbound above it.

Name

Name for each of the flows. They must be distinct from each other.

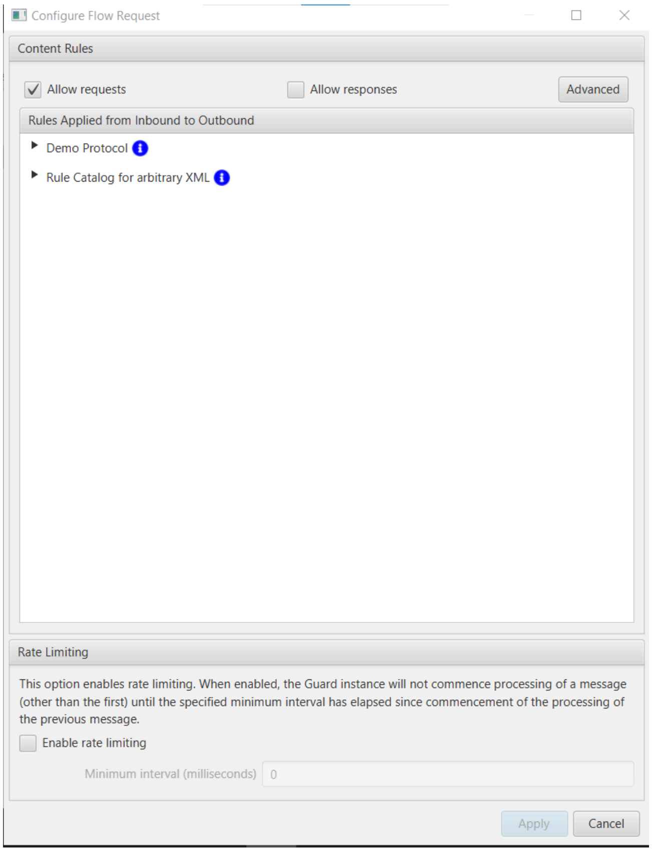
Flow Rules

Bring up the Configure Flow dialog for the flow. This is a dialog that can be used to configure what content the flow carries and what rules to apply to application data on the flow.

Configure Flow dialog

This dialog allows the configuration of the content rules for the flow.

Figure A.66. Flow Rules



Configure Flow Request

Content Rules

☑ Allow requests □ Allow responses Advanced

Rules Applied from Inbound to Outbound

▸ Demo Protocol

▸ Rule Catalog for arbitrary XML

Rate Limiting

This option enables rate limiting. When enabled, the Guard instance will not commence processing of a message (other than the first) until the specified minimum interval has elapsed since commencement of the processing of the previous message.

□ Enable rate limiting

Minimum interval (milliseconds) 0

Apply Cancel

Content Rules

This section allows the configuration of the type and content of GCXP messages allowed across the flow.

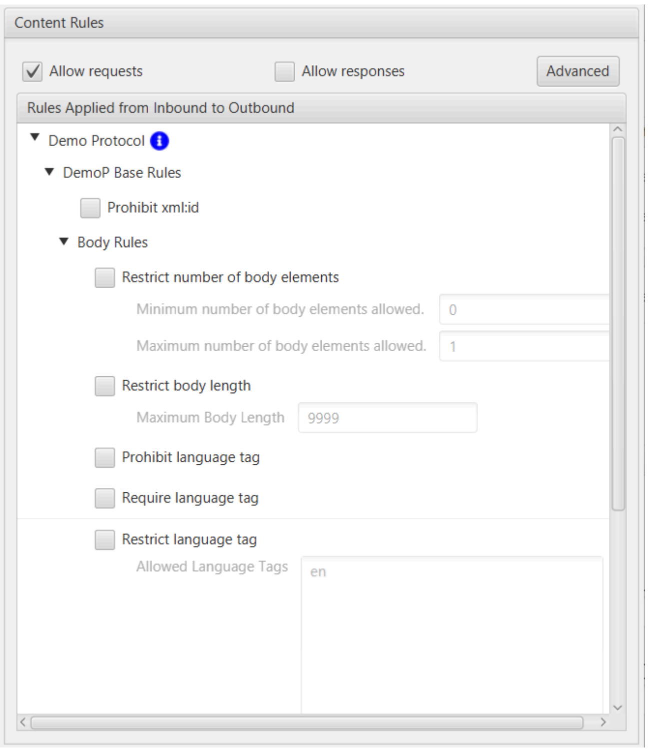
The type of GCXP message allowed can be configured with two options:

Allow Requests — This option, when checked, allows flow of content checked by the configured rules. This option should only be enabled on the forward flow.

Allow responses — This option, when checked, allows return of responses indicating whether the associated request was received by the content consumer or rejected by the Guard instance. The option should only be enabled on reverse flows and set in accordance with the application's GCXP profile, such as to meet the application's need for reliable delivery.

If requests are allowed then the applied rules section will be enabled. This displays the names of the rule catalogs, if any, enabled for the project. These can be expanded to show all the rules in the catalog.

Figure A.67. Flow Rules



Content Rules

☑ Allow requests   □ Allow responses   Advanced

Rules Applied from Inbound to Outbound

- ▼ Demo Protocol
  - ▼ DemoP Base Rules
    - □ Prohibit xml:id
    - ▼ Body Rules
      - □ Restrict number of body elements
        - Minimum number of body elements allowed. 0
        - Maximum number of body elements allowed. 1
      - □ Restrict body length
        - Maximum Body Length 9999
      - □ Prohibit language tag
      - □ Require language tag
      - □ Restrict language tag
        - Allowed Language Tags en

Rules can be selected by selecting the check box next to them.

Some rules come with parameters. These will come with default values, but once the rule is selected these can be changed to customize the rule. Once selected the Guard instance will ensure content that passes across it matches the rule.

Rate Limiting

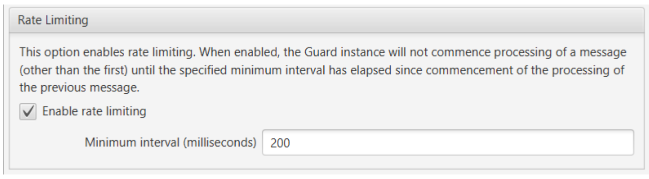
This section allows GCXP message rating limiting to be enabled and configured. Rate limiting is accomplished by enforcing a minimum interval between processing of consecutively received messages.

Rate limiting is intended to be used where the producer, at a fixed interval, transmits a single GCXP message and is otherwise silent. As such, this feature is not generally applicable to most deployments.

Warning

Rate limiting, when enabled, creates an artificial communication bottleneck which can degrade the guarded application service. Bottlenecks can be used to mount denial (or degradation) of service attacks.

Figure A.68. Rate Limiting



Rate Limiting

This option enables rate limiting. When enabled, the Guard instance will not commence processing of a message (other than the first) until the specified minimum interval has elapsed since commencement of the processing of the previous message.

☑ Enable rate limiting

Minimum interval (milliseconds) 200

Enable Rate Limiting

When selected rate limiting is enabled. When not selected rate limiting is disabled.

Minimum Interval

The minimum interval, in milliseconds, required between consecutively received messages in the flow. When enabled, the Guard instance will not commence processing of a received message (other than the first message) until the specified amount of time as the minimum interval has elapsed since the processing of previously received message commenced.

Appendix B. Guard Content Exchange Protocol (GCXP)

Abstract

This appendix describes the *Guard Content Exchange Protocol* (GCXP).

What is GCXP?

The *Guard Content Exchange Protocol* (GCXP) is an *application-level* protocol used for exchange content between two systems through a guard device or possibly a series of guard devices.

For instance, two messaging services may exchange content through a series of guard devices. A GCXP end-to-end session is typically used to provide for a unidirectional flow of traffic between a producing service (the producer) and a consuming service (the consumer). The producer is a peer to the first guard in this series. The consumer is a peer to the last guard in the series. Each guard has two peers, one which it receives content from, the inbound peer, and one which it forwards traffic to, the outbound peer.

Where two messaging systems need to transfer bidirectionally, two separate GCXP end-to-end sessions must be configured and used, each with an independent series of guard devices. For instance, to support instant messaging dialog between a user of one system and a user of another, two separate and independent GCXP streams are required.

Depending on the application protocol and operational requirements, GCXP can be configured to allow signaling of successful/non-successful transfer of application data to be given to the provider, as may be needed for reliable delivery. However, this is optional as, in other cases, such signaling is prohibited. GCXP is designed to allow guard devices to integrate with physical-layer data diodes.

GCXP can be used to exchange application data in any form.

Reference Implementation

Isode has published an open-source reference implementation of GCXP, as well as an implementation of CBOR, and example GCXP consumer and producer programs. This software is available for free at https://bitbucket.org/isode. This can be used to develop applications to communicate with the Guard or other applications supporting GCXP.

Specifying a GCXP Application Profile

The *Guard Content Exchange Protocol* (GCXP) can be used in a variety of ways to transfer data in support of an application (e.g., e-mail). To promote interoperability between application implementations, a GCXP profile should be specified which details how all implementations of application are to use GCXP. These profiles also provide guidance on how guard devices and other intermediaries should behave.

Within a GCXP stream, GCXP requests flow in one direction and, where permitted, responses flow in the reverse direction. When responses are allowed, it is recommended that they have no payload (content type is *nil*). Where application protocols provide, within the application layer, a request/response protocol, it is recommended that the application level requests and responses use separate GCXP streams. GCXP responses generally should have no payload. The application profile must specify what content is carried in GCXP requests.

GCXP supports optional responses to requests to transfer content. These are used to indicate whether the content was accepted at the final entity or rejected by any entity. They are used to provide *transactional* behavior. The response may optionally carry application content (not recommended).

The profile should detail whether GCXP responses are expected and, if they are, what content, if any, can be carried in them.

Typically content exchanged (when exchanged) is XML (recommended). It is recommended that restrictions be placed upon the XML, such as UTF-8 only (required) and no XML comments (recommended). It is recommended that content restrictions be specified using formal languages such as XSD, RelaxNG, and/or Schematron. It is recommended that the profile not allow arbitrary content structure and, in particular, the schema specification be inclusive all content that can be exchanged. Where *any* constructs are used (not recommended), they should specify strict validation and that the fulfillment of the any be restricted by other means (e.g. Schematron). It is recommended that the content root node be restricted to a particular element. This can be done by restricting the primary XSD to a single global element (recommended) or otherwise.

It is recommended that content be required to be normalized. For instance, one can apply an XSLT stylesheet to normalize the XML, such as to use a particular prefix for each namespace. Whitespace should be normalized where appropriate. Additionally, the XML should be in canonical form, such as *Canonical XML* (1.0 or 1.1) or *Exclusive Canonical XML* (1.0). Also, Unicode representations of characters should be normalized. It is recommended that *Normalization Form Canonical Composition* (NFC) be used.

It may also be appropriate to normalize particular application datatypes that have multiple representations of an abstract value. For instance, a user in the application might be identified as a case-insensitive string and, where so, normalizing to a particular case (e.g., lower) would be appropriate to reduce the number of equivalences to one.

In many cases, in addition to general application content restrictions, deployments may want to enforce restrictions above and beyond those of the application profile. Profile specifications should generally note this and offer any guidance here that might be appropriate.

Specifying a GCXP Application Profile in XML for M-Guard

M-Guard Console uses an XML representation of a protocol's GCXP application profile to configure guard instances for protocol. The profile may also reference rule catalogs and example catalogs. Rule catalogs specify rules which may be optionally used in a Guard instance's configuration. Example catalogs provide example payloads for testing purposes, such as use with the GCXP Producer built into M-Guard Console.

M-Guard Console has two built-in application profiles, the Arbitrary XML profile and Demo Protocol (DemoP) profile. Isode provides JAR-formatted file containing these profiles, as well as schemas for application profile, rule catalog, and example catalog file formats. The DemoP profile is intended to serve as an example to profile authors. It is recommend utilize the same file layout used for the DemoP profile in the Profiles JAR, including containing their profile folder within a folder called `profiles` and similarly constructed signed JAR files for distribution. Subsequent versions of M-Guard are expected to support importation of profiles from such a signed JAR file.

The contents of the Profiles JAR may be extracted into a folder using the jar(1) command. It is recommended the contents be extracted into a new (empty) folder.

```
jar -xf M-Guard-Console-M-N-U-Profiles.jar
```

The XML representation is expected to conform to following schema:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://isode.com/guard/console/app-profile/0"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    targetNamespace="http://isode.com/guard/console/app-profile/0"
 elementFormDefault="qualified"
    attributeFormDefault="unqualified">
```

```
   <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="xml.xsd"/>
   <xs:element name="application-profile">
       <xs:annotation>
           <xs:documentation>M-Guard Application Profile</xs:documentation>
       </xs:annotation>
       <xs:complexType>
           <xs:sequence>
               <xs:element name="meta-info" minOccurs="0">
                   <xs:complexType>
                       <xs:sequence>
                           <xs:element name="name" maxOccurs="unbounded"
type="localized-string"/>
                           <xs:element name="version" maxOccurs="unbounded"
minOccurs="0"
                               type="localized-string"/>
                           <xs:element name="description" maxOccurs="unbounded"
minOccurs="0"
                               type="localized-string"/>
                           <xs:any minOccurs="0" maxOccurs="unbounded"
namespace="##other"
                               processContents="lax"/>
                       </xs:sequence>
                   </xs:complexType>
               </xs:element>
               <xs:element name="traffic" type="traffic">
                   <xs:unique name="uniqueTraffics">
                       <xs:selector xpath="flow"/>
                       <xs:field xpath="@name"/>
                   </xs:unique>
               </xs:element>
               <xs:element name="schemas" minOccurs="0">
                   <xs:complexType>
                       <xs:sequence>
                           <xs:element name="schema" maxOccurs="unbounded"
type="fileElement"/>
                       </xs:sequence>
                       <xs:attributeGroup ref="fileAttributes"/>
                   </xs:complexType>
               </xs:element>
               <xs:element name="rules" type="catalogs" minOccurs="0"/>
               <xs:element name="examples" type="catalogs" minOccurs="0"/>
           </xs:sequence>
       </xs:complexType>
   </xs:element>
   <xs:complexType name="catalogs">
       <xs:sequence>
           <xs:element name="catalog" maxOccurs="unbounded" type="fileElement"/>
       </xs:sequence>
   </xs:complexType>
   <xs:complexType name="fileElement">
       <xs:attributeGroup ref="fileAttributes"/>
   </xs:complexType>
   <xs:complexType name="localized-string">
       <xs:simpleContent>
           <xs:extension base="xs:string">
               <xs:attribute ref="xml:lang"/>
           </xs:extension>
       </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="traffic">
       <xs:annotation>
           <xs:documentation>Traffic configuration</xs:documentation>
       </xs:annotation>
       <xs:sequence>
           <xs:element name="flow" type="flow" minOccurs="2" maxOccurs="2">
               <xs:unique name="uniqueAllows">
                   <xs:selector xpath="allow"/>
                   <xs:field xpath="@type"/>
               </xs:unique>
           </xs:element>
       </xs:sequence>
   </xs:complexType>
   <xs:complexType name="flow">
       <xs:annotation>
           <xs:documentation>Traffic flow configuration</xs:documentation>
       </xs:annotation>
       <xs:sequence>
           <xs:element name="allow" type="allow" minOccurs="0"
maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="name" type="xs:token" use="required"/>
       <xs:attribute name="in" type="xs:token" use="required"/>
       <xs:attribute name="out" type="xs:token" use="required"/>
   </xs:complexType>
   <xs:complexType name="allow">
       <xs:annotation>
           <xs:documentation>Traffic allow configuration</xs:documentation>
       </xs:annotation>
       <xs:sequence>
           <xs:element name="verify" type="verify" minOccurs="0"
maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attribute name="type" use="required">
           <xs:simpleType>
               <xs:annotation>
                   <xs:documentation>GCXP message type</xs:documentation>
               </xs:annotation>
               <xs:restriction base="xs:token">
                   <xs:enumeration value="request"/>
                   <xs:enumeration value="response"/>
               </xs:restriction>
           </xs:simpleType>
       </xs:attribute>
       <xs:attribute name="format" use="optional">
           <xs:simpleType>
               <xs:annotation>
                   <xs:documentation>GCXP payload format</xs:documentation>
               </xs:annotation>
               <xs:restriction base="xs:token">
                   <xs:enumeration value="xml"/>
                   <xs:enumeration value="nil"/>
               </xs:restriction>
           </xs:simpleType>
       </xs:attribute>
       <xs:attribute name="optional" type="xs:boolean" default="false"/>
   </xs:complexType>
   <xs:attributeGroup name="fileAttributes">
       <xs:attribute name="file">
           <xs:simpleType>
               <xs:annotation>
                   <xs:documentation>File path</xs:documentation>
               </xs:annotation>
               <xs:restriction base="xs:token">
                   <xs:minLength value="1"/>
               </xs:restriction>
           </xs:simpleType>
       </xs:attribute>
       <xs:attribute name="data" type="xs:base64Binary"/>
   </xs:attributeGroup>
   <xs:attributeGroup name="min-max">
       <xs:attribute name="min" type="xs:nonNegativeInteger" default="0"/>
       <xs:attribute name="max" type="xs:nonNegativeInteger" default="0"/>
   </xs:attributeGroup>
   <xs:simpleType name="xpath">
       <xs:annotation>
           <xs:documentation>XPath expression</xs:documentation>
       </xs:annotation>
       <xs:restriction base="xs:token">
           <xs:minLength value="1"/>
       </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="non-empty-string">
       <xs:annotation>
```

```
            <xs:documentation>XPath expression</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:attributeGroup name="ruleAttributes">
        <xs:attribute name="rule" type="xs:token"/>
    </xs:attributeGroup>
    <xs:complexType name="param">
        <xs:annotation>
            <xs:documentation>Verify parameter</xs:documentation>
        </xs:annotation>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="value" type="xs:string" use="required"/>
        <xs:attribute name="desc" type="xs:string"/>
    </xs:complexType>
    <xs:complexType name="verify">
        <xs:annotation>
            <xs:documentation>Traffic verify configuration</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="identity-transform" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="xslt" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="param" type="param"
 minOccurs="0"
                                        maxOccurs="unbounded"/>
                                </xs:sequence>
                                <xs:attributeGroup ref="fileAttributes"/>
                                <xs:attributeGroup ref="ruleAttributes"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="xc14n" minOccurs="0">
                            <xs:complexType>
                                <xs:attribute name="algorithm" use="required">
                                    <xs:simpleType>
                                        <xs:annotation>
                                            <xs:documentation>XC14N Algorithm</
xs:documentation>
                                        </xs:annotation>
                                        <xs:restriction base="xs:token">
                                            <xs:enumeration value="xc14n_1_0"/>
                                            <xs:enumeration value="xc14n_1_1"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:attribute>
                                <xs:attributeGroup ref="ruleAttributes"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="unicode" minOccurs="0">
                            <xs:complexType>
                                <xs:attribute name="form" use="required">
                                    <xs:simpleType>
                                        <xs:annotation>
                                            <xs:documentation>Unicode Normal Form</
xs:documentation>
                                        </xs:annotation>
                                        <xs:restriction base="xs:token">
                                            <xs:enumeration value="nfc"/>
                                            <xs:enumeration value="nfd"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:attribute>
                                <xs:attributeGroup ref="ruleAttributes"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    <xs:attributeGroup ref="ruleAttributes"/>
                </xs:complexType>
            </xs:element>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="count">
                    <xs:complexType>
                        <xs:attribute name="units">
                            <xs:simpleType>
                                <xs:annotation>
                                    <xs:documentation>Units to be counted</
xs:documentation>
                                </xs:annotation>
                                <xs:restriction base="xs:token">
                                    <xs:enumeration value="bytes"/>
                                    <xs:enumeration value="codepoints"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:attribute>
                        <xs:attributeGroup ref="min-max"/>
                        <xs:attributeGroup ref="ruleAttributes"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="relaxng">
                    <xs:complexType>
                        <xs:attributeGroup ref="fileAttributes"/>
                        <xs:attributeGroup ref="ruleAttributes"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="schematron">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="param" type="param" minOccurs="0"
                                maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attributeGroup ref="fileAttributes"/>
                        <xs:attributeGroup ref="ruleAttributes"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="xpath">
                    <xs:complexType>
                        <xs:attribute name="path" use="required" type="xpath"/>
                        <xs:attributeGroup ref="min-max"/>
                        <xs:attributeGroup ref="ruleAttributes"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="xsd">
                    <xs:complexType>
                        <xs:attribute name="xpath" type="xpath"/>
                        <xs:attributeGroup ref="fileAttributes"/>
                        <xs:attributeGroup ref="ruleAttributes"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="xslt">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="param" type="param" minOccurs="0"
 maxOccurs="unbounded"/>
                            <xs:element name="accept" minOccurs="0"
 maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:attributeGroup ref="fileAttributes"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="xpath" type="xpath"/>
                        <xs:attributeGroup ref="fileAttributes"/>
                        <xs:attributeGroup ref="ruleAttributes"/>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:sequence>
        <xs:attributeGroup ref="ruleAttributes"/>
    </xs:complexType>
```

```
</xs:schema>
```

A copy of this schema is available in the Profiles JAR in the file `jaxb/app-profile.xsd`.

The profile is expected to adhere to the following conventions:

- Relative file paths must be used.
- `<schema/>` element file names must be of the form `schemas/name.ext` where `.ext` is an appropriate extension for the content type.
- `<schemas/>` element file name, if present, must be `catalog.xml` and must be an XML catalog which maps URIs to schema files in the `schemas` folder.
- This XML schema catalog, if provided, must use URIs which are the same as the corresponding `<schema/>` file name attribute value.
- The rules catalog file name, if specified, must be `rules/catalog.xml` and must conform to rule catalog schema. This schema is available in the Profiles JAR in the file `jaxb/rule-catalog.xsd`. The DemoP rule catalog provides examples of how various rules are to be specified. The Appendix C, *Rule Catalogs* appendix provides additional information about specify rule catalogs. All files referenced by the catalog should be placed directly within the `rules` folder.
- The examples catalog file name, if specified, must be `examples/catalog.xml` and must conform to example catalog schema. This schema is available in the Profiles JAR in the file `jaxb/example-catalog.xsd`. All files referenced by the catalog should be placed directly within the `examples` folder.

Appendix C. Rule Catalogs

Abstract

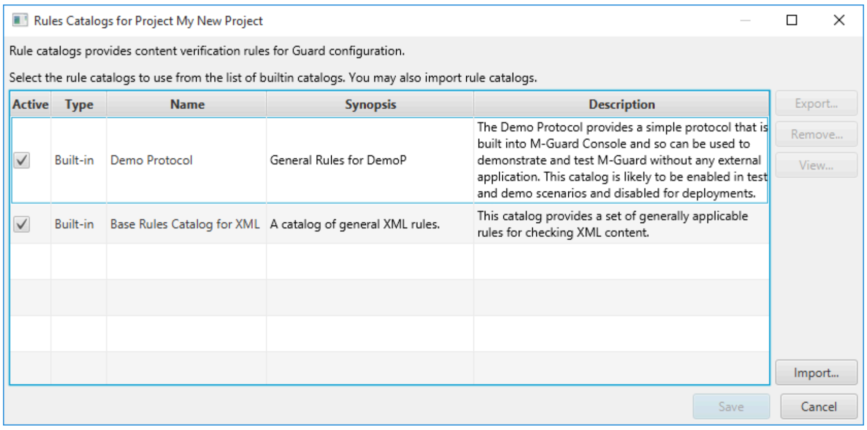
This appendix contains rule formats, syntax and catalogs.

Overview

Many users of M-Guard Console will simply import an appropriate set of rule catalogs and then use them to configure Guard instances. For such users, details of the rule catalog format are not of interest.

This appendix is written for users who wish to develop a rule catalog from scratch or wish to modify an existing rule catalog.

Figure C.1. List of M-Guard Catalogs



Rules Catalogs for Project My New Project

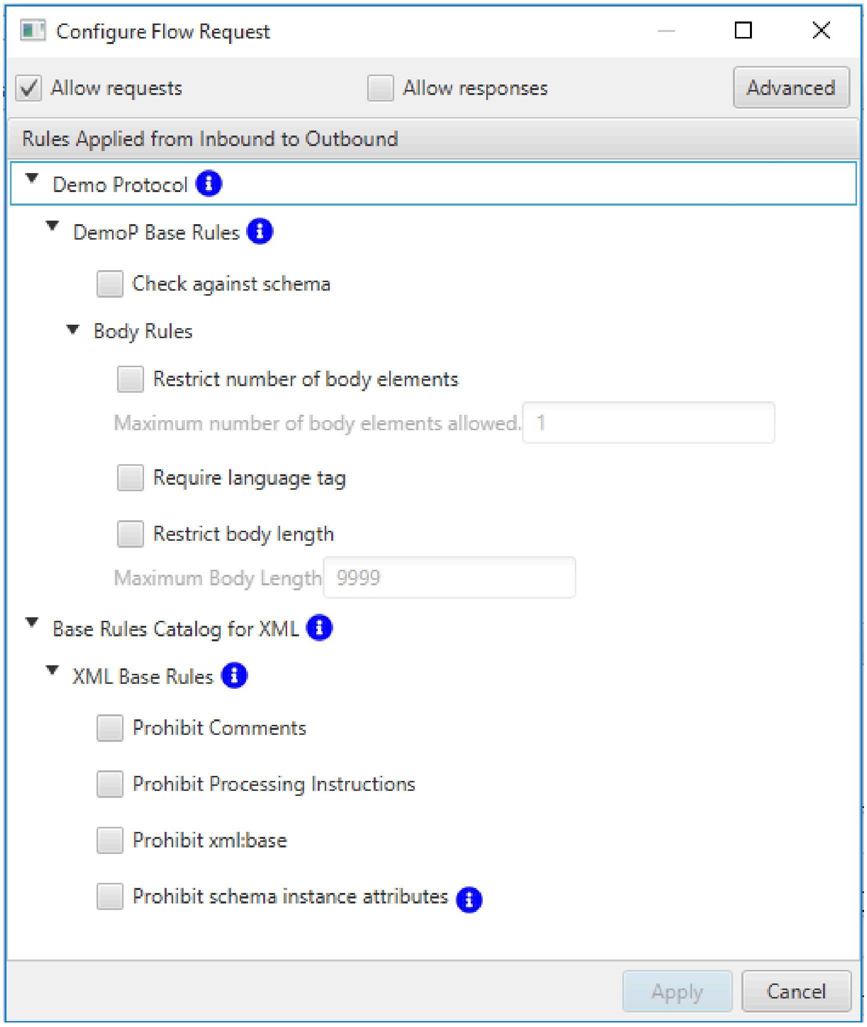
Rule catalogs provides content verification rules for Guard configuration.

Select the rule catalogs to use from the list of builtin catalogs. You may also import rule catalogs.

Active	Type	Name	Synopsis	Description
☑	Built-in	Demo Protocol	General Rules for DemoP	The Demo Protocol provides a simple protocol that is built into M-Guard Console and so can be used to demonstrate and test M-Guard without any external application. This catalog is likely to be enabled in test and demo scenarios and disabled for deployments.
☑	Built-in	Base Rules Catalog for XML	A catalog of general XML rules.	This catalog provides a set of generally applicable rules for checking XML content.

M-Guard Console uses rules grouped into catalogs to configure a Guard instance's XML content checks. Rule catalogs provide a convenient hierarchical collection of rules that can be imported into an M-Guard Console Project. A list of rule catalogs is shown above. When configuring a Guard instance, any combination of rules from any rule catalog in the Project can be used. Rules may have parameters which need to specified. Rule configuration for a Guard instance is shown below:

Figure C.2. Rule Configuration



Configure Flow Request

☑ Allow requests

□ Allow responses

Rules Applied from Inbound to Outbound

▼ Demo Protocol

▼ DemoP Base Rules

□ Check against schema

▼ Body Rules

□ Restrict number of body elements

Maximum number of body elements allowed: 1

□ Require language tag

□ Restrict body length

Maximum Body Length: 9999

▼ Base Rules Catalog for XML

▼ XML Base Rules

□ Prohibit Comments

□ Prohibit Processing Instructions

□ Prohibit xml:base

□ Prohibit schema instance attributes

Supported Rule Types

Rules use standard XML formats to apply checks. These are:

- XML Schema [https://en.wikipedia.org/wiki/XML_schema]. Content can be checked against a standard XML schema specification.

- XPath [https://en.wikipedia.org/wiki/XPath] (XML Path Language). XPath queries which match a set of nodes can be evaluated and checked against a min/max parameters.

- Schematron [https://en.wikipedia.org/wiki/Schematron]. Schematron can be used for making a wide range of content assertions.

- RELAX NG [https://en.wikipedia.org/wiki/RELAX_NG] (REgular LAnguage for XML. Next Generation). Content can be checked against this alternative to XML Schemas.

All of these formats are useful, and the Rule designer will choose a format that is convenient for the Rule that is being expressed.

This appendix assumes that the reader is familiar with these syntaxes. They are well documented in other places.

Catalog Syntax

Overview

Rule Catalogs are distributed as an XML file, which may be packaged with a set of files referenced from the base XML file. Any rules may be specified in line in the file. Rule catalogs will generally be exported and distributed as a single file.

Rules may be placed in separate files referenced from the catalog. This will usually be a convenient structure when catalogs are being developed. M-Guard Console can import catalogs structured in this manner.

The syntax of the catalog allows descriptions (synopsis and full description) of the catalog and each rule to be included. M-Guard Console will display this information.

Examples

The catalog syntax is simple, and easily illustrated by the two built-in catalogs.

Base Rules Catalog

The first catalog has two basic rules that are generally applicable to application content being checked by M-Guard. The two rules are both included XPath rules:

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="http://isode.com/guard/console/rules/0">
    <name>Rule Catalog for arbitrary XML</name>
    <synopsis>A catalog of general XML rules</synopsis>
    <description>This catalog provides a set of generally applicable rules for
 checking arbitrary XML content.</description>
    <collection>
        <name>XML Rules</name>
        <synopsis>A set of XML rules to provide generally applicable content
 checks.</synopsis>
        <description>These rules are generally applicable to XML content exchanged
 using GCXP.</description>
        <rule id="91C8980C-CA4B-4730-9128-1CE71BF8DA0A">
            <name>Prohibit Comments</name>
            <synopsis>Deny XML content which contains comments.</synopsis>
            <description/>
            <xpath path="/descendant::comment()" min="0" max="0"/>
        </rule>
        <rule id="86BD7EF0-F916-4CB8-AE00-69AD640B011D">
            <name>Prohibit Processing Instructions</name>
            <synopsis>Deny XML content which contains processing instructions.</
synopsis>
            <description/>
            <xpath path="/descendant::processing-instruction()" min="0" max="0"/>
        </rule>
        <rule id="FD49CF5D-F6C5-4C51-8C81-A738FF5F9BD0">
            <name>Prohibit xml:base</name>
            <schematron file="xml-base.sch"/>
        </rule>
        <rule id="8AF1A91A-01A6-424F-BE15-93B3DB9D2AF0">
            <name>Prohibit schema instance attributes</name>
            <description>Prohibits elements from having any XML schema instance
 attributes: type, base, schemaLocation, noNamespaceSchemaLocation.</description>
            <schematron file="xml-schema-instance.sch"/>
        </rule>
    </collection>
</catalog>
```

Demo Protocol Catalog

The second catalog relates to the Demonstration Protocol that can be used to demonstrate and evaluate M-Guard capabilities. It also serves as an example catalog.

```
<?xml version="1.0" encoding="UTF-8"?>
<catalog xmlns="http://isode.com/guard/console/rules/0">
    <name>Demo Protocol</name>
    <synopsis>Rules for DemoP</synopsis>
    <description>The Demo Protocol provides a simple protocol that is built into
 M-Guard Console and so can be used to demonstrate and test M-Guard without any
 external application. This catalog is likely to be enabled in test and demo
 scenarios and disabled for deployments.</description>
    <collection>
        <name>DemoP Base Rules</name>
        <synopsis>Base Rules for DemoP</synopsis>
        <description/>
        <rule id="0D377D4C-BD07-4D68-8ACC-1E9FB0A9E36B">
            <name>Prohibit xml:id</name>
            <schematron file="xml-id.sch"/>
        </rule>
        <collection>
            <name>Body Rules</name>
            <synopsis>Rules applying to body element</synopsis>
            <rule id="B528015F-80F7-41D2-A2C1-E1E30F7599C6">
                <name>Restrict number of body elements</name>
                <schematron file="body-count.sch">
                    <parameter ref="min">
                        <name>Minimum number of body elements allowed.</name>
                        <synopsis/>
                        <nonNegativeInteger>0</nonNegativeInteger>
                    </parameter>
                    <parameter ref="max">
                        <name>Maximum number of body elements allowed.</name>
                        <synopsis/>
                        <nonNegativeInteger>1</nonNegativeInteger>
                    </parameter>
                </schematron>
            </rule>
            <rule id="25D7CBA2-801F-4A67-9FDF-1DAC77EE60BB">
                <name>Restrict body length</name>
                <schematron file="body-length.sch">
                    <parameter ref="length">
                        <name>Maximum Body Length</name>
                        <synopsis>Body length (in characters) over specified number
 will not be allowed</synopsis>
                        <positiveInteger>9999</positiveInteger>
                    </parameter>
                </schematron>
            </rule>
            <rule id="5618820A-282A-4DBD-8E2B-F0E67123EEB2">
                <name>Prohibit language tag</name>
                <excludes>4FDAF953-AF57-4A1E-9E0A-82515AEC45E7</excludes>
                <schematron file="body-lang-prohibit.sch"/>
            </rule>
            <rule id="4FDAF953-AF57-4A1E-9E0A-82515AEC45E7">
                <name>Require language tag</name>
                <excludes>5618820A-282A-4DBD-8E2B-F0E67123EEB2</excludes>
                <schematron file="body-lang-require.sch"/>
            </rule>
            <rule id="D2706194-10C4-4E8E-907C-EDF5E90F5383">
                <name>Restrict language tag</name>
                <excludes>5618820A-282A-4DBD-8E2B-F0E67123EEB2</excludes>
                <schematron file="body-lang-restrict.sch">
                    <parameter ref="langs" multi="true">
                        <name>Allowed Language Tags</name>
                        <synopsis>Allow only the specified language tags (and their
 subtags)</synopsis>
                        <language>en</language>
                    </parameter>
                </schematron>
```

```
           </rule>
           <rule id="57943D9E-4EB6-40D5-9C6D-2A414439F27C">
               <name>Prohibit Dirty Words</name>
               <schematron file="body-dirty.sch">
                   <parameter ref="dirty" multi="true">
                       <name>Prohibited Words</name>
                       <synopsis>Prohibit the body element from containing any of
the specified words.</synopsis>
                       <string>SECRET</string>
                   </parameter>
               </schematron>
           </rule>
       </collection>
       <collection>
           <name>Date-Time Rules</name>
           <rule id="F91067A0-AE50-4416-AE4F-97C19FF69B73">
               <name>Restrict number of date-time elements</name>
               <schematron file="date-time-count.sch">
                   <parameter ref="min">
                       <name>Minimum number of date-time elements allowed.</name>
                       <nonNegativeInteger>0</nonNegativeInteger>
                   </parameter>
                   <parameter ref="max">
                       <name>Maximum number of date-time elements allowed.</name>
                       <nonNegativeInteger>0</nonNegativeInteger>
                   </parameter>
               </schematron>
           </rule>
       </collection>
   </collection>
   <collection>
       <name>DemoP Extensions</name>
       <rule id="2DF91294-06F1-4AF1-B144-1F1CCDAB24B7">
           <name>Prohibit Extensions</name>
           <excludes>5dd56af1-5177-11ea-88a8-080027161cd5</excludes>
           <schematron file="extensions-prohibit.sch"/>
       </rule>
       <rule id="5dd56af1-5177-11ea-88a8-080027161cd5">
           <name>Require Extensions</name>
           <excludes>2DF91294-06F1-4AF1-B144-1F1CCDAB24B7</excludes>
           <schematron file="extensions-require.sch"/>
       </rule>
       <collection>
           <name>Security Labels (XEP 258)</name>
           <rule id="29E0DEB1-68C0-4E9D-B929-2CC9895CAF2C">
               <name>Restrict number of Security Labels</name>
               <schematron file="xep258-count.sch">
                   <parameter ref="min">
                       <name>Minimum number of XEP 258 securitylabel elements
allowed.</name>
                       <synopsis/>
                       <nonNegativeInteger>0</nonNegativeInteger>
                   </parameter>
                   <parameter ref="max">
                       <name>Maximum number of XEP 258 securitylabel elements
allowed.</name>
                       <synopsis/>
                       <nonNegativeInteger>1</nonNegativeInteger>
                   </parameter>
               </schematron>
           </rule>
           <rule id="41A6B893-FF72-47F5-B267-94295CC5838E">
               <name>Prohibit equivalent labels</name>
               <schematron file="xep258-equiv.sch"/>
           </rule>
           <rule id="61630A84-274A-4A5F-81B6-F16F70164F3B">
               <name>Restrict Security Labels</name>
               <description>Restricts XEP-258 Security Label in the GCXP payload to
selected values.</description>
               <xslt
                   xpath="//*[local-name() = 'securitylabel' and namespace-uri() =
'urn:xmpp:sec-label:0']"
```

```
                    file="xep258.xsl" content-uri="urn:xmpp:sec-label:0"
                />
            </rule>
        </collection>
        <collection>
            <name>NATO Metadata Label (STANAG 4474/4478)</name>
            <excludes>2DF91294-06F1-4AF1-B144-1F1CCDAB24B7</excludes>
            <rule id="09BE7877-A890-4EAE-A72C-5C42197A063A">
                <name>Restrict NATO Metadata</name>
                <description>Restricts NATO Metadata information in the GCXP payload
 to selected values. Ignores Digital Signatures.

Restrictions on digital signatures can be enforced by using additional rules.</
description>
                <xslt
                    xpath="//*[local-name() = 'BindingInformation' and namespace-
uri() = 'urn:nato:stanag:4778:bindinginformation:1:0']"
                    file="metadata.xsl" content-
uri="urn:nato:stanag:4778:bindinginformation:1:0"
                />
            </rule>
            <rule id="167743C9-88E8-430E-ACDA-15576BAE398C">
                <name>Restrict number of BindingInformation elements</name>
                <schematron file="metadata-bi-count.sch">
                    <parameter ref="min">
                        <name>Minimum number of BindingInformation elements
 allowed.</name>
                        <synopsis/>
                        <nonNegativeInteger>0</nonNegativeInteger>
                    </parameter>
                    <parameter ref="max">
                        <name>Maximum number of BindingInformation elements
 allowed.</name>
                        <synopsis/>
                        <nonNegativeInteger>1</nonNegativeInteger>
                    </parameter>
                </schematron>
            </rule>
            <rule id="AE9A4130-7893-4EEA-9CD3-298AEF27B8DA">
                <name>Restrict number of Signature elements</name>
                <schematron file="metadata-sig-count.sch">
                    <parameter ref="min">
                        <name>Minimum number of Signatures elements allowed.</name>
                        <synopsis/>
                        <nonNegativeInteger>0</nonNegativeInteger>
                    </parameter>
                    <parameter ref="max">
                        <name>Maximum number of Signature elements allowed.</name>
                        <synopsis/>
                        <nonNegativeInteger>1</nonNegativeInteger>
                    </parameter>
                </schematron>
            </rule>
            <collection>
                <name>Detail Rules</name>
                <excludes>09BE7877-A890-4EAE-A72C-5C42197A063A</excludes>
                <rule id="5B07EF27-19FA-45F3-AFE4-A83B5F08DCC1">
                    <name>Restrict number of MetaData Binding Container elements</
name>
                    <schematron file="metadata-mdbc-count.sch">
                        <parameter ref="min">
                            <name>Minimum number of MetaDataBindingContainer
 elements allowed.</name>
                            <synopsis/>
                            <positiveInteger>1</positiveInteger>
                        </parameter>
                        <parameter ref="max">
                            <name>Maximum number of MetadataBindingContainer
 elements allowed.</name>
                            <synopsis/>
                            <positiveInteger>1</positiveInteger>
                        </parameter>
                    </schematron>
                </rule>
                <rule id="1E0B453A-5914-44C8-88C9-9652AEB310CC">
                    <name>Restrict number of MetaDataBinding elements per
 Container</name>
                    <schematron file="metadata-mdb-count.sch">
                        <parameter ref="min">
                            <name>Minimum number of MetaDataBinding elements
 allowed.</name>
                            <synopsis/>
                            <positiveInteger>1</positiveInteger>
                        </parameter>
                        <parameter ref="max">
                            <name>Maximum number of MetadataBinding elements
 allowed.</name>
                            <synopsis/>
                            <positiveInteger>1</positiveInteger>
                        </parameter>
                    </schematron>
                </rule>
                <rule id="0F544E1B-36D5-4605-9501-959F0978DB7D">
                    <name>Restrict number of MetaData elements per MetaDataBinding</
name>
                    <schematron file="metadata-md-count.sch">
                        <parameter ref="min">
                            <name>Minimum number of MetaData elements allowed.</
name>
                            <synopsis/>
                            <nonNegativeInteger>1</nonNegativeInteger>
                        </parameter>
                        <parameter ref="max">
                            <name>Maximum number of Metadata elements allowed.</
name>
                            <synopsis/>
                            <nonNegativeInteger>1</nonNegativeInteger>
                        </parameter>
                    </schematron>
                </rule>
                <rule id="B1CC6D9E-B735-42A6-AC7C-5D7AAA107BDB">
                    <name>Restrict number of AlternativeConfidentialLabel MetaData
 elements per MetaDataBinding</name>
                    <schematron file="metadata-md-ac-count.sch">
                        <parameter ref="min">
                            <name>Minimum number of AlternativeConfidentialLabel
 MetaData elements allowed.</name>
                            <synopsis/>
                            <nonNegativeInteger>1</nonNegativeInteger>
                        </parameter>
                        <parameter ref="max">
                            <name>Maximum number of AlternativeConfidentialLabel
 Metadata elements allowed.</name>
                            <synopsis/>
                            <nonNegativeInteger>1</nonNegativeInteger>
                        </parameter>
                    </schematron>
                </rule>
                <rule id="84FB24F3-159B-4DD3-BD14-D00CEBA8008F">
                    <name>Restrict number of DataReference elements per
 MetaDataBinding</name>
                    <schematron file="metadata-dr-count.sch">
                        <parameter ref="min">
                            <name>Minimum number of DataReference elements
 allowed.</name>
                            <synopsis/>
                            <nonNegativeInteger>1</nonNegativeInteger>
                        </parameter>
                        <parameter ref="max">
                            <name>Maximum number of DataReference elements
 allowed.</name>
                            <synopsis/>
                            <nonNegativeInteger>1</nonNegativeInteger>
                        </parameter>
                    </schematron>
                </rule>
                <rule id="CFA56AF4-D40A-11EB-A5C3-B5528CB8E476">
                    <name>Restrict number of Transforms elements per DataReference</
name>
                    <schematron file="metadata-trans-count.sch">
                        <parameter ref="min">
                            <name>Minimum number of Transforms elements allowed.</
name>
                            <synopsis/>
                            <nonNegativeInteger>0</nonNegativeInteger>
                        </parameter>
                        <parameter ref="max">
                            <name>Maximum number of Transforms elements allowed.</
name>
                            <synopsis/>
                            <nonNegativeInteger>1</nonNegativeInteger>
                        </parameter>
                    </schematron>
                </rule>
                <rule id="1994F423-370E-4708-870A-84062F87C611">
                    <name>Restrict DataReference URI to same-document with fragment
 reference</name>
                    <excludes>14A11DEE-7AFD-4491-8638-7DACBC18CEEE</excludes>
                    <schematron file="metadata-dr-same-document-w-fragment.sch"/>
                </rule>
                <rule id="14A11DEE-7AFD-4491-8638-7DACBC18CEEE">
                    <name>Restrict DataReference URI to same-document without
 fragment reference</name>
                    <excludes>1994F423-370E-4708-870A-84062F87C611</excludes>
                    <schematron file="metadata-dr-same-document-wo-fragment.sch"/>
                </rule>
            </collection>
        </collection>
    </collection>
</catalog>
```

Schema

The formal schema of a rules catalog is defined by the following XML schema:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://isode.com/guard/console/rules/0"
    targetNamespace="http://isode.com/guard/console/rules/0"
 elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xs:element name="catalog" type="collection">
        <xs:annotation>
            <xs:documentation>Guard(8) Rule Catalog</xs:documentation>
            <xs:documentation>A catalog of content rules, organized by possibly
 nested
                collections</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:group name="common">
        <xs:annotation>
            <xs:documentation>Elements common to most elements.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element name="name" type="xs:string">
                <xs:annotation>
                    <xs:documentation>Names the item</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="synopsis" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Provides a short summary of the item, suitable
 for a
                        tooltip.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="description" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Provides a description of the item, suitable
 for a help
                        dialog.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="note" type="xs:string" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>A note intended for developers and
 integrators. Should not be
                        presented to the user.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="requires" minOccurs="0" maxOccurs="unbounded"
 type="xs:token"/>
            <xs:element name="excludes" minOccurs="0" maxOccurs="unbounded"
 type="xs:token"/>
        </xs:sequence>
    </xs:group>
    <xs:complexType name="collection">
        <xs:sequence>
            <xs:group ref="common"/>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="collection" type="collection" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>Holds a set of rules, possibly organized
 by inner
                            collections.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="rule" type="rule" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>A content rule.</xs:documentation>
                    </xs:annotation>
                </xs:element>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="rule">
        <xs:sequence>
            <xs:group ref="common"/>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="identity-transform" type="identityTransform"/>
                <xs:element name="count" type="count"/>
                <xs:element name="relaxng" type="relaxng"/>
                <xs:element name="schematron" type="schematron"/>
                <xs:element name="xpath" type="xpath"/>
                <xs:element name="xsd" type="xsd"/>
                <xs:element name="xslt" type="xslt-xpath"/>
            </xs:choice>
        </xs:sequence>
        <xs:attribute name="id" type="xs:token" use="required"/>
    </xs:complexType>
    <xs:complexType name="identityTransform">
        <xs:sequence>
            <xs:element name="xslt" minOccurs="0" maxOccurs="unbounded" type="xslt"/
>
            <xs:element name="xc14n" minOccurs="0" type="xc14n"/>
            <xs:element name="unicode" minOccurs="0" type="unicode"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="count">
        <xs:attribute name="units">
            <xs:simpleType>
                <xs:annotation>
                    <xs:documentation>Units to be counted</xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:token">
                    <xs:enumeration value="bytes"/>
                    <xs:enumeration value="codepoints"/>
```

```
               </xs:restriction>
           </xs:simpleType>
       </xs:attribute>
       <xs:attributeGroup ref="min-max"/>
   </xs:complexType>
   <xs:complexType name="relaxng">
       <xs:attributeGroup ref="fileAttributes"/>
   </xs:complexType>
   <xs:complexType name="schematron">
       <xs:sequence>
           <xs:element name="parameter" type="parameter" minOccurs="0"
maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attributeGroup ref="fileAttributes"/>
   </xs:complexType>
   <xs:complexType name="xslt-xpath">
       <xs:sequence>
           <xs:element name="parameter" type="parameter" minOccurs="0"
maxOccurs="unbounded"/>
           <xs:element name="accept" minOccurs="0" maxOccurs="unbounded">
               <xs:complexType>
                   <xs:attributeGroup ref="fileAttributes"/>
               </xs:complexType>
           </xs:element>
       </xs:sequence>
       <xs:attribute name="xpath" type="xpath-expression"/>
       <xs:attributeGroup ref="fileAttributes"/>
       <xs:attribute name="content-uri" type="xs:anyURI"/>
   </xs:complexType>
   <xs:complexType name="unicode">
       <xs:attribute name="form" use="required">
           <xs:simpleType>
               <xs:annotation>
                   <xs:documentation>Unicode Normal Form</xs:documentation>
               </xs:annotation>
               <xs:restriction base="xs:token">
                   <xs:enumeration value="nfc"/>
                   <xs:enumeration value="nfd"/>
               </xs:restriction>
           </xs:simpleType>
       </xs:attribute>
   </xs:complexType>
   <xs:complexType name="xc14n">
       <xs:attribute name="algorithm" use="required">
           <xs:simpleType>
               <xs:annotation>
                   <xs:documentation>XC14N Algorithm</xs:documentation>
               </xs:annotation>
               <xs:restriction base="xs:token">
                   <xs:enumeration value="xc14n_1_0"/>
                   <xs:enumeration value="xc14n_1_1"/>
               </xs:restriction>
           </xs:simpleType>
       </xs:attribute>
   </xs:complexType>
   <xs:complexType name="xslt">
       <xs:sequence>
           <xs:element name="parameter" type="parameter" minOccurs="0"
maxOccurs="unbounded"/>
       </xs:sequence>
       <xs:attributeGroup ref="fileAttributes"/>
   </xs:complexType>
   <xs:complexType name="xpath">
       <xs:attribute name="path" type="xpath-expression"/>
       <xs:attributeGroup ref="min-max"/>
   </xs:complexType>
   <xs:complexType name="xsd">
       <xs:attributeGroup ref="fileAttributes"/>
       <xs:attribute name="xpath" type="xpath-expression"/>
   </xs:complexType>
   <xs:attributeGroup name="fileAttributes">
       <xs:attribute name="file" type="non-empty-string"/>
```

```
        <xs:attribute name="data" type="xs:base64Binary"/>
    </xs:attributeGroup>
    <xs:attributeGroup name="min-max">
        <xs:attribute name="min" type="xs:nonNegativeInteger" default="0"/>
        <xs:attribute name="max" type="xs:nonNegativeInteger" default="0"/>
    </xs:attributeGroup>
    <xs:complexType name="parameter">
        <xs:sequence>
            <xs:group ref="common"/>
            <xs:choice>
                <xs:annotation>
                    <xs:documentation>Indicates the syntax of the parameter and
 contains the
                        (required) default value.</xs:documentation>
                </xs:annotation>
                <xs:element name="string" type="non-empty-string"/>
                <xs:element name="integer" type="xs:integer"/>
                <xs:element name="nonNegativeInteger" type="xs:nonNegativeInteger"/>
                <xs:element name="positiveInteger" type="xs:positiveInteger"/>
                <xs:element name="boolean" type="xs:boolean"/>
                <xs:element name="language" type="xs:language"/>
            </xs:choice>
        </xs:sequence>
        <xs:attribute name="ref" type="xs:string" use="required"/>
        <xs:attribute name="multi" type="xs:boolean" default="false"/>
    </xs:complexType>
    <xs:simpleType name="xpath-expression">
        <xs:annotation>
            <xs:documentation>XPath expression</xs:documentation>
        </xs:annotation>
        <xs:restriction base="xs:token">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="non-empty-string">
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

Rule Types

This section provides some examples of each of the supported rule types.

XML Schema

XML Schema is the standard for specifying XML schemas. XML schemas are useful for specifying what XML content is allowed to be transferred. The XML Schema for the Demo Protocol is shown below. The contents of the file `demop.xsd` are shown below:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="http://isode.com/guard/demo-protocol/0"
    targetNamespace="http://isode.com/guard/demo-protocol/0"
 elementFormDefault="qualified"
    attributeFormDefault="unqualified">


    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
 schemaLocation="http://www.w3.org/2009/01/xml.xsd"/>
    <xs:import namespace="urn:xmpp:sec-label:0" schemaLocation="urn:xmpp:sec-
label:0"/>
    <xs:import namespace="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"
 schemaLocation="urn:nato:stanag:4774:confidentialitymetadatalabel:1:0"/>
    <xs:import namespace="urn:nato:stanag:4778:bindinginformation:1:0"
 schemaLocation="urn:nato:stanag:4778:bindinginformation:1:0"/>

    <xs:element name="demo">
        <xs:annotation>
```

```
            <xs:documentation>Guard(8) Demo Protocol</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="body" minOccurs="0" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:simpleContent>
                            <xs:extension base="xs:string">
                                <xs:attribute ref='xml:lang' use='optional'/>
                            </xs:extension>
                        </xs:simpleContent>
                    </xs:complexType>
                </xs:element>
                <xs:element name="date-time" type="xs:dateTime" minOccurs="0"/>
                <xs:element name="extensions" minOccurs="0">
                    <xs:complexType>
                        <xs:choice minOccurs="1" maxOccurs="unbounded">
                            <xs:any namespace="##other"/>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute ref='xml:id' use='optional'/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

XPath

XPath is a convenient way to specify some restrictions, such as those beyond the scope of schema. For instance, it is used to prohibit comments and processing instructions. Rules for both of these are included in the basic XML rule catalogs. The comment prohibition rule is:

```
<xpath path="/descendant::comment()" min="0" max="0"/>
```

XPath rules are comparing the number of nodes matched by the expression with the min=, max= parameters.

Schematron

Schematron provides a flexible XML rule based specification. It is a convenient language for specifying a range of constraints and checks on XML messages.

The following example checks the body length and ensures that the value is less than a specified length. This is in a file `body-length.sch` which is referenced from the catalog.

```
<?xml version="1.0"?>
<schema xmlns="http://purl.oclc.org/dsdl/schematron">
    <ns prefix="demop" uri="http://isode.com/guard/demo-protocol/0"/>
    <pattern id="demop">
        <rule context="/demop:demo/demop:body" id="body-length">
            <assert test="string-length(.) &lt;= $length">too long</assert>
        </rule>
    </pattern>
</schema>
```

This provides a simple check on comparing the length of a matching XML element with the Schematron parameter of the rule.