

COMPUTING

SSF for MaxCore™ MC3000 Platform

Installation and Use

P/N: 6806800T81K

July 2017

ARTESYN[™]
EMBEDDED TECHNOLOGIES

© Copyright 2017 Artesyn Embedded Technologies, Inc.
All rights reserved.

Trademarks

Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc. All other names and logos referred to are trade names, trademarks, or registered trademarks of their respective owners. © 2017 Artesyn Embedded Technologies, Inc. All rights reserved. For full legal terms and conditions, please visit www.artesyn.com/legal.

Notice

While reasonable efforts have been made to assure the accuracy of this document, Artesyn assumes no liability resulting from any omissions in this document, or from the use of the information obtained therein. Artesyn reserves the right to revise this document and to make changes from time to time in the content hereof without obligation of Artesyn to notify any person of such revision or changes.

Electronic versions of this material may be read online, downloaded for personal use, or referenced in another document as a URL to an Artesyn website. The text itself may not be published commercially in print or electronic form, edited, translated, or otherwise altered without the permission of Artesyn.

It is possible that this publication may contain reference to or information about Artesyn products (machines and programs), programming, or services that are not available in your country. Such references or information must not be construed to mean that Artesyn intends to announce such Artesyn products, programming, or services in your country.

Limited and Restricted Rights Legend

If the documentation contained herein is supplied, directly or indirectly, to the U.S. Government, the following notice shall apply unless otherwise agreed to in writing by Artesyn.

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data clause at DFARS 252.227-7013 (Nov. 1995) and of the Rights in Noncommercial Computer Software and Documentation clause at DFARS 252.227-7014 (Jun. 1995).

Contact Address

Artesyn Embedded Technologies
Marketing Communications
2900 S. Diablo Way, Suite 190
Tempe, Arizona 85282

Contents

About this Manual	11
1 Introduction	15
1.1 PEX Modes and Network Configuration	15
1.1.1 Single-Host Mode	16
1.1.2 Multi-Host Mode	16
1.2 Components of SSF	17
1.3 SSF Access Methods	18
1.3.1 Web Interface	19
1.3.2 XML Interface	19
1.3.3 CLI	20
1.3.4 SNMP Interface	21
2 SSF Installation and Upgrade	23
2.1 Software Deliverables	23
2.2 Installing SSF	27
2.2.1 Installing SSF on PCIE-7410 and PCIE-9205 Cards	27
2.2.2 Installing SSF on PCIE-7210 Card	27
2.2.3 Installing SSF Agents for Target PCIE Cards	28
2.3 Installing SSF with SSF Core on External PC	30
2.3.1 Installing Linux on connected PC	31
2.3.1.1 Installing CentOS on Connected PC	31
2.3.2 Updating Interface Name of External PC	33
2.3.3 Installing SSF	34
2.3.4 Configuring System Network	35
2.3.5 Uninstallation Procedure on External Linux PC	36
2.3.6 Upgrade Procedure on External Linux PC	37
2.3.7 Starting SSF on External PC	37
2.3.7.1 Starting SSF core on External PC	37
2.4 Upgrading Shelf using System Update	38
2.4.1 Checking Shelf Status	39
2.4.2 Initiating System Update	42
2.5 Upgrading SSF	42

3	SSF Service Configuration	47
3.1	Starting SSF	47
3.1.1	Starting SSF Core	47
3.1.2	Starting SSF Agent	47
3.2	Managing SSF MaxCore Agent	47
4	Accessing SSF	49
4.1	Accessing SSF using Web Interface	49
4.2	Accessing SSF using XML Interface	51
4.3	Accessing SSF using CLI	51
4.4	Accessing SSF using SNMP	52
5	Configuration Management	55
5.1	Adding or Remove Emulator Mode Candidate Names	55
5.1.1	Adding a Configuration Candidate	56
5.1.2	Removing a Configuration Candidate	56
A	Additional Information	57
A.1	Resetting Administrator Password	57
A.2	Configuring a Non-Artesyn card for SSF	58
A.3	Updating Shelf Address	59
A.4	Network Boot Configuration	60
A.5	Setting Default Configuration	60
A.6	Enabling Network Time Protocol	61
A.7	Configuring Multiple Shelves on MaxCore	61
A.8	Configuring firewall to Allow SSF Communication	63
A.9	Verifying SSF and BBS Versions Installed on the System	64
A.10	Changing Logging Configuration	64
A.11	SSF Core Configuration	65
A.12	SSF Agent Configuration	68
A.12.1	Service Manager Configuration	69
A.12.2	Service Manager Configuration INI File	70
A.13	Hardware Agent Configuration	70

A.14	Ramdisk Creation for PCIE-7210	72
B	Troubleshooting and FAQ	75
B.1	Overview	75
B.2	System Discover or Active Configuration is Empty	75
B.3	SSF Core Failure	75
B.4	Host OS Not Displayed	76
B.5	Login Failure	76
B.6	Switch Management Tab is Hidden	77
B.7	Reload Failure	77
B.8	GUI Access and Logging Issues	78
B.9	PCIE-9205 Switch Management is Not Populated in GUI	78
B.10	Incorrect Device Id to RRC PEP Port Mapping	79
B.11	How to check whether SSF Services are running fine	80
C	Related Documentation	81
C.1	Artesyn Embedded Technologies - Embedded Computing Documentation	81

List of Tables

Table 1-1	Command Line Editing Features	20
Table 2-1	System Update Package	23
Table 2-2	Package for PCIE-7410 and PCIE-9205	23
Table 2-3	PCIE-7210 package	23
Table 2-4	List of SSF and Dependent RPMs for Host	24
Table 2-5	Software Update Package Contents	42
Table A-1	SSF Core Configuration Files	66
Table A-2	SSF Server Configuration Files	68
Table A-3	Hardware Agent Configuration	71
Table C-1	Artesyn Embedded Technologies - Embedded Computing Publications	81

List of Figures

Figure 1-1	Components of SSF	17
Figure 2-1	External PC based System Manager	30
Figure 4-1	Login Page	50
Figure A-1	Non-Artesyn Card in MaxCore system - Before Configuration	58
Figure A-2	Non-Artesyn Card in MaxCore system - After Configuration	59

About this Manual

Overview of Contents

This manual provides information on System Services Framework (SSF) configuration and different methods of accessing SSF. It contains the following chapters and appendices.

- [Chapter 1, Introduction, on page 15](#) provides an overview of SSF and its features.
- [Chapter 2, SSF Installation and Upgrade, on page 23](#) provides step-by-step procedure of SSF installation.
- [Chapter 3, SSF Service Configuration, on page 47](#) provides information for configuring SSF.
- [Chapter 4, Accessing SSF, on page 49](#) provides step-by-step procedures to access SSF using various interfaces.
- [Appendix A, Additional Information](#) provides additional information required for SSF configuration.
- [Appendix B, Troubleshooting and FAQ](#) provides troubleshooting and frequently asked questions.
- [Appendix C, Related Documentation](#) lists the relevant manuals.

Abbreviations

These are the abbreviations used in this manual.

Abbreviation	Definition
AAA-Server	Authentication, Authorization, and Accounting Server
BCSIM	Blade Common System Information Model
BMC	Baseboard Management Controller
CSIM	Common System Information Model
FRU	Field Replaceable Unit
LCU	Log Collection Utility
MCPU	Management CPU
RADIUS	Remote Authentication Dial-In User Service
SMAN	Service Manager
SSF	System Services Framework

Abbreviation	Definition
SMC	System Management Controller
TCP	Transmission Control Protocol
TL-Server	Transport Layer Server
MIB	Management Information Base
MO	Managed Objects
PCIe	Peripheral Component Interconnect Express
UDP	User Datagram Protocol
UDS	Unix Domain Socket
VF	Virtual Function

Conventions

The table below describes the conventions used throughout this manual.

Notation	Description
0x00000000	Typical notation for hexadecimal numbers (digits are 0 through F) used for addresses, offsets, and so on
0b0000	Same for binary numbers (digits are 0 and 1)
bold	Used to emphasize a word
<i>Screen</i>	Used for on-screen output and code related elements or commands in body text
Courier + Bold	Used to characterize user input and to separate it from system output
<i>Reference</i>	Used for references and for table and figure descriptions
File > Exit	Notation for selecting a sub-menu
<text>	Notation for variables and keys
[text]	Notation for software buttons to click on the screen and parameter description
...	Repeated item for example node 1, node 2, ..., node 12

Summary of Changes

Part Number	Date	Description
6806800T81K	July 2017	Added Chapter 2, Installing SSF with SSF Core on External PC, on page 30 , and Appendix A, Ramdisk Creation for PCIe-7210 . Updated Appendix B, How to check whether SSF Services are running fine , and Service Manager Configuration INI File on page 70 ,
6806800T81J	April 2017	Added Appendix B, Troubleshooting and FAQ , and Appendix A, Changing Logging Configuration . Updated Chapter 1, Introduction, on page 15 , Chapter 2, SSF Installation and Upgrade, on page 23 , Chapter 3, SSF Service Configuration, on page 47 , and Appendix A, Updating Shelf Address .
6806800T81H	April 2017	Updated Chapter 1, Introduction, on page 15 and Appendix A, Updating Shelf Address .
6806800T81G	January 2017	Updated Table 2-4 on page 24 , Chapter 2, SSF Installation and Upgrade, on page 23 , and Appendix A, Configuring Multiple Shelves on MaxCore, on page 61 .
6806800T81F	November 2016	Removed section 1.1.2.1 Network Connectivity . Added Chapter 2, Installing SSF on PCIe-7210 Card, on page 27 .
6806800T81E	September 2016	Updated Chapter 2, SSF Installation and Upgrade, on page 23 , Chapter 3, SSF Service Configuration, on page 47 , and Chapter 4, Accessing SSF, on page 49 . Added Appendix A, Verifying SSF and BBS Versions Installed on the System, on page 64 .
6806800T81D	August 2016	Added Appendix A, Updating Shelf Address, on page 59 , Appendix A, Configuring Multiple Shelves on MaxCore, on page 61 , Appendix A, Configuring firewall to Allow SSF Communication, on page 63 . Updated Chapter 1, Introduction, on page 15 , Chapter 2, SSF Installation and Upgrade, on page 23 .
6806800T81C	June 2016	Added Notice in Chapter 2, Installing SSF, on page 27 and Appendix A, Configuring a Non-Artesynt card for SSF, on page 58 . Updated Chapter 3, SSF Service Configuration, on page 47 , and Chapter 1, Introduction, on page 15 and Changed Manual Title.
6806800T81B	April 2016	Added Chapter 1, SNMP Interface, on page 21 , Chapter 2, Upgrading SSF, on page 29 , and Chapter 4, Accessing SSF using SNMP, on page 52 .
6806800T81A	January 2016	Initial version.

MaxCore™ MC3000 platform is unique in its ability to combine CPU-attached PCI Express cards with an extremely flexible communication network among these CPUs. Traditional single box server architectures provide either a single multi-core server that can be combined with a small number of PCI Express based I/O cards, or offer multiple independent server nodes with or without minimal local I/O extension. However, the MaxCore platform supports both types of architectures.

System Services Framework (SSF) provides a management and configuration interface to Artesyn's hardware and software products. It facilitates system level configuration and management access to SSF managed hardware and software components through Web, XML, and CLI protocol interfaces.

SSF represents all the managed hardware and software components in a simple and easily manageable hierarchal model. It also supports persistency and playback of MaxCore configuration.

The following are the key features supported by SSF:

- Access, Authentication, and Authorization
- Configuration persistency, Reload, and Rollback
- Hierarchal representation of System model
- Dynamic population of System model
- Remote system configuration Management
- Application Management of Remote systems
- Event and Alarm management
- Graphical Monitoring of Sensors

1.1 PEX Modes and Network Configuration

MaxCore has two different architecture modes:

- Single-Host Mode
- Multi-Host Mode

shelfHost

shelfHost is a central management entity in a MaxCore with many CPUs. It is a combination of BMC, SSF, and BBS. It manages the MaxCore infrastructure, such as power supplies, fans, and USB/SATA/PF/VF assignments.

applicationHost

An applicationHost processes the data it receives through the network functions assigned to it by a shelf host.

systemHost

systemHost can manage a single MaxCore or a stack of many MaxCores which is called the “System” in the SSF terminology. The systemHost can be located on any CPU within the SSF network. This can also be a third-party server or a PC.

1.1.1 Single-Host Mode

In this case, it is a single PCIe domain. Where only one slot (slot1 or slot15) is populated with one Host card and other slots (Downstream Slots) are populated with PCIe endpoint cards. This mode can be compared with any other PCIe Rack server (For example, Dell R720 or R320) with a Host processor and multiple PCIe Slots (14 Slots).

1.1.2 Multi-Host Mode

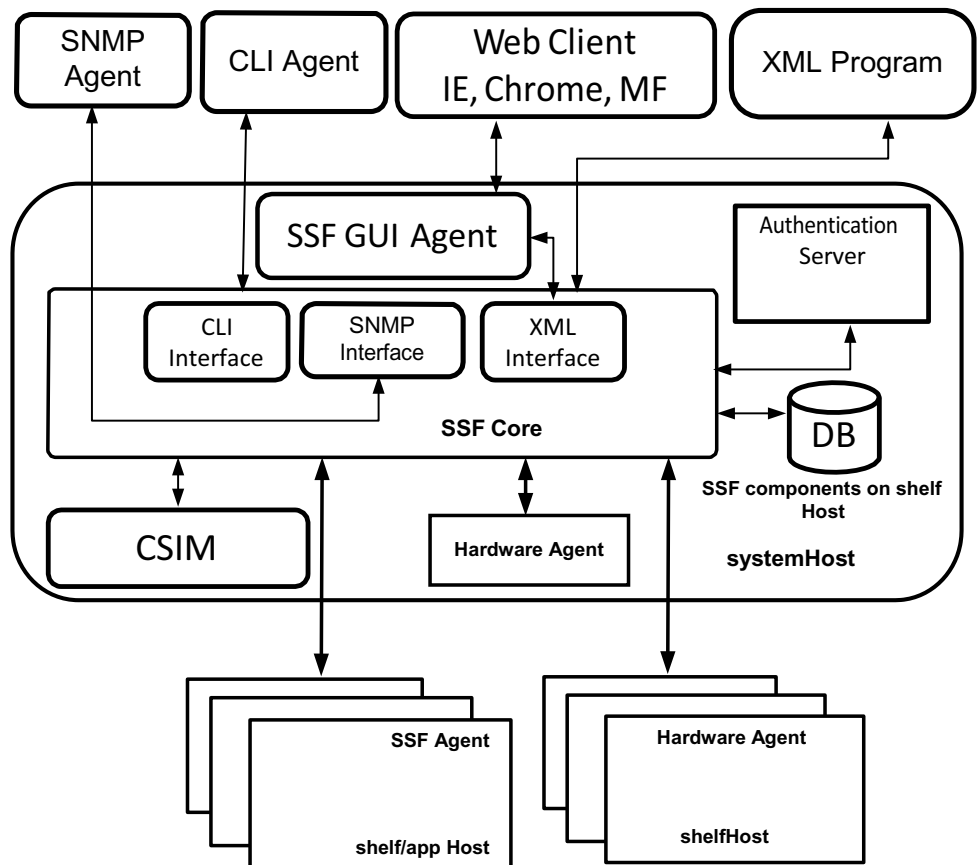
In Multi-Host Mode, multiple Slots are configured as Host Ports. There can be multiple PCIe Hosts, that contains PCIe endpoints associated with them. The Active Shelf Host can either assign the complete PCIe device to a PCIe Host or it assigns the Virtual Function (VF) of a device.

By default, MaxCore is configured in Base Mode. You can switch the MaxCore to Express fabric mode using SSF. This will result in the reset of the shelf host and bring-up MaxCore with the Multi-Host configuration, now set as default. You can decide on the default fabric mode configuration set separately.

1.2 Components of SSF

The following figure illustrates various components of SSF.

Figure 1-1 Components of SSF



SSF has the following components:

- **SSF Core** - It is the central component of SSF. It provides a single point access to the complete system. It interacts with the SSF External Interface Agent (XML Interface Agent, CLI Interface Agent) and multiple SSF Agents. It receives all the requests from different external interface agents, forwards the request to the one or many targeted SSF Agents, receives the response and respond it back to the external interface agent. It also receives asynchronous events from the SSF Agents and forwards them to the external interface agents, who have registered for events. SSF Core only interacts with Authentication, Authorization, and Accounting Server (AAA- Server) for user authentication and Database server for persistency. An AAA server is a server program that handles user requests for access to computer resources and for an enterprise, provides authentication, authorization, and accounting (AAA) services. The AAA server typically interacts with network access and gateway servers and with databases and directories containing user information. The current standard by which devices or applications communicate with an AAA server is the Remote Authentication Dial-In User Service (RADIUS).
- **SSF GUI Agent** - It is an application that interacts with the SSF Core to provide web interface for accessing SSF managed system. Similar to SSF core, SSF GUI Agent has to be installed on shelf host.
- **SSF Agent (Host OS Management)** - It is present in the Shelf Host. It communicates with the SSF Core and provides access to platform services, such as `syslog-ng`, and a user defined service `syslcu` installed on the SSF Core. You can also add your own platform services that can be managed using SSF. Please refer to [Service Manager Configuration on page 69](#).
- **SSF Agent (Application Management)** - It is an interface between the south-bound interface of SSF and a north-bound interface of an application. It forwards all the requests from the SSF Core to an application and vice versa.

1.3 SSF Access Methods

You can access SSF using the following interfaces:

- Web interface
- XML interface
- CLI
- SNMP

These interfaces provide:

- Hierarchic system view
- Complete System Management
- System Configuration
- General Linux services and also proprietary application services

1.3.1 Web Interface

You can access SSF using a web-based application for configuring, managing, and monitoring Artesyn's systems and their components. The web interface represents the system in a hierarchical structure that helps in identifying various components at different levels. This provides a pictorial view of the system with navigation links to view component details and perform actions.

The web interface of SSF has the following capabilities:

Accessibility

- HTTP-based access to the web interface.

User Management

- Configuration and management of different user categories.

Monitoring

- Centralized logging system based on `syslog-ng`
- Logging of alarms and events
- Graphical monitoring of Platform Parameters such as Fan Speed, power sensors, and temperature sensors

For more details, see [Chapter 4, Accessing SSF, on page 49](#).

1.3.2 XML Interface

XML Interface enables you to communicate with SSF core via the XML agent. XML agent forwards requests from the XML interface to the SSF core and sends back responses and notifications from the SSF core to the XML interface.

For more information about the XML interface, refer to the *System Services Framework for MaxCore™ Platform XML Interface Guide*.

1.3.3 CLI

SSF provides a fully functional CLI with auto complete, history, and help features.

You can access CLI through a telnet session. Multiple sessions can be created simultaneously and the number of sessions supported is only limited by the available system resources.

SSF CLI enables you to traverse the system hierarchically and provide configuration and management access to the SSF system. All these hierarchies are represented as nodes.

The CLI of SSF has the following capabilities:

- CLI Access to different users can be restricted using different privilege levels.
- Multiple client sessions for a single user to access the system can be configured.

The following table provides command line editing features of SSF CLI.

Table 1-1 Command Line Editing Features

Keyboard keys	Description
Left and Right arrow	Allow you to move the cursor within the current command line.
Up and Down arrow	Allow you to browse through a command history.
BACKSPACE	Enables you to remove the character towards left.
TAB	Completes the keyword being entered automatically.
“?”	Provides you context help.
<cr>	Carriage return. System displays this command when you provide all mandatory arguments of a particular CLI command. It represents the command syntax completion.

1.3.4 SNMP Interface

Simple Network Management Protocol (SNMP) is a protocol for network management which monitors and manage devices on networks. It is used for collecting information from, and configuring, network devices, such as servers, switches, and routers on an IP network. Also, it exposes management data in the form of variables on the managed systems, which describe the system configuration. For more details, see [Chapter 4, Accessing SSF using SNMP, on page 52](#).

SSF Installation and Upgrade

SSF can be downloaded using the SWORDS download server available from the Artesyn Customer Resource Center website (<http://crportal.artesyn.com>).

2.1 Software Deliverables

SSF software is delivered as a complete package, which includes SSF RPMs, dependent RPMs, Basic Blade Services (BBS), and an installation script. The software deliverables include packages for PCIE-7410, PCIE-9205, and PCIE-7210. These packages also include deliverables for associated PCIE target cards, such as PCIE-8120, and PCIE-7207. The following provide package details of PCIE-7410, PCIE-9205, and PCIE-7210 cards.

Table 2-1 System Update Package

Package	Description
SSFMXCORE-R_<package-ver>.iso	MC3K software (update)- A single update package for System Update feature. It contains update BMC, BBS, and SSF update packages for PCIE-7410, PCIE-7210, and PCIE-9205 cards.

Table 2-2 Package for PCIE-7410 and PCIE-9205

Package	Description
install_disk.img	MC3k software (full) - Contains SSF, Kernel, BBS, and so on.
mc3ksw_update_7410_<os>-<os-ver>_<package-ver>.iso	MC3k software (update) - Contains BBS and SSF repositories for PCIE-7410, PCIE-8120, and PCIE-7207.
ramdisk.image.gz	MC3K software (full) - Updated ramdisk image contains SSF, Kernel, and BBS.
rootfs.tar.bz2	Compressed root filesystem updated with SSF packages.

Table 2-3 PCIE-7210 package

Package	Description
mc3ksw_update_7210_<os>-<os-ver>_<package-ver>.iso	Contains BBS and SSF repositories for PCIE-7210, PCIE-8120, and PCIE-7207.

Table 2-4 lists the contents (SSF and Dependent RPMs) in the delivered packages.

Table 2-4 List of SSF and Dependent RPMs for Host

RPM	Description
SSF Core	
ssf_maxcore_main_rel-<Version>-el7.x86_64.rpm	Contains binary and configuration files of SSF Core and Common System Information Model (CSIM). This has to be installed on shelfHost.
Hardware Agent	
ssf_maxcore_mxcagent_rel-<version>-el7.x86_64	Contains Hardware Platform Agent binaries and configuration files. This should be installed on shelfHost of the MaxCore. In multi-shelf environment this should be installed on shelfHost of all MaxCores. Hardware Agent performs Hardware discovery by communicating with BMC and PEX and provides the information to SSF.
SSF GUI Agent	
ssf_maxcore_gui_rel-<version>-ssf_<Version>-el7.x86_64.rpm	Contains Zend Framework, httpd server, and GUI configuration scripts to be installed on a Shelf Host.
ssf_maxcore_gui_help_rel-<version>-el7.x86_64	Provides SSF GUI help pages.
ssf_maxcore_gui_console_rel-<Version>-el7.x86_64.rpm	Contains console support for SSF web interface to be installed on a Shelf Host.
SSF Agent	
ssf_maxcore_bcsim_rel-<7410 9205 7210>-<version>-el7.x86_64.rpm	<p>Contains binary and configuration files of the SSF Agent. This also includes the Linux service configuration file.</p> <p>The SSF Agent is used to configure and maintain the OS services, Interfaces, OS information, and so on. This SSF Agent is for PCIE-7410, PCIE-9205, and PCIE-7210 delivered or packaged along with the BBS</p> <p>Release mentioned in <i>MaxCore™-MC3000BBS-Release-Notes</i>.</p>

Table 2-4 List of SSF and Dependent RPMs for Host (continued)

RPM	Description
ssf_maxcore_rrcTLS_rel-9205-<version number>-el7.x86_64.rpm	Contains the binary and configuration files for SSF switch agent.
switchsw-pcie9205-<version number>-el7.x86_64.rpm	Switch Software Release RPM.
SSF Version	
ssf_version_rel-<7410 9205>-<version>-el7.x86_64	Contains SSF compatibility versions.
ViewCheck Agent	
ssf_maxcore_viewcheckTLS_rel-<Card Number>-<version number>-el7.x86_64.rpm	Provides binary and configuration files of the ViewCheck SSF Agent. This RPM is dependent on Artesyn ViewCheck RPM.
ViewCheck	
viewcheck_rel-<Card Number>-<version number>-el7.x86_64.rpm	<p>This RPM contains the ViewCheck service. The ViewCheck RPM image is functionally comprises diagnostics framework, specific test cases, and test suites. The ViewCheck RPM always uses the same OS variant and compile time environment based on the BBS release of the target PCIE card. ViewCheck service RPM consists of:</p> <ul style="list-style-type: none"> ● ViewCheck Core -Daemon ● Static Test Suite Configuration files for the specific card ● Start/Stop/Restart /Status scripts for ViewCheck Core
sysLCU	
syslcu.pcie<7210 7410 9205>-<version>.noarch	Contains scripts and configuration files of sysLCU utility.
PCIE-7207 Specific RPMs	
viewchecklib_rel-7207-<version>-el7.x86_64.rpm	Provides binary and configuration files for the ViewCheck library for PCIE-7207.
ssf_maxcore_mcpdiag_rel-7207-<version>-el7.x86_64.rpm	Provides the binary and configuration files for SSF MCP ViewCheck agent.

Table 2-4 List of SSF and Dependent RPMs for Host (continued)

RPM	Description
ssf_maxcore_mcpagt_rel-7207-1.1.0.13-el7.x86_64.rpm	Provides the configuration and binary files for MCP agent.
PCIE-7207 Ramdisk Specific RPMs	
viewcheck_rel-7207-<version>-el7.x86_64.rpm	Provides binary and configuration files for the ViewCheck RPM for PCIE-7207.
ssf_maxcore_mcpagtcInt_rel-7207-<version>-el7.x86_64.rpm	Provides the binary and configuration files for SSF MCP client, which interacts with SSF MCP agent.
PCIE-8120 Specific RPMs	
qt-<version>.el7_2.x86_64.rpm	Dependent RPMs for the octasic-sdk.
qt-settings-<version>.el7.centos.noarch.rpm	
sofia-sip-<version>.el7.centos.x86_64.rpm	
octasic-sdk-<version>-1A.el7.centos.x86_64.rpm	Contains mainly firmware image and some tools required for booting up the DSPs.
pcie8120-<version>-MC_1.el7.centos.x86_64.rpm	Contains set of binaries, C-library, octmezz commands and mainly PCIE-8120 kernel module
ssf_maxcore_smcTLS_rel-8120-<version>-el7.x86_64.rpm	Provides the binary and configuration files for SSF PCIE-8120 card configuration agent.
Dependent RPMs	
freeradius-1.1.7-el7.x86_64.rpm	Contains access and authentication services on SSF.
libpqxx-2.6.8-el7.x86_64.rpm	Library to communicate with PostGRE database.
postgresql-9.1.3-el7.x86_64.rpm	Database to provide configuration persistency.
vsftpd-3.0.2-9-el7.x86_64.rpm	Supports file transfer from remote user.
expect-5.45-12.el7.x86_64.rpm	Contains scripts and configuration files of expect utility.
tcl-8.5.13-8.el7.x86_64.rpm	Dependent rpm for expect utility.
ftp binary	Supports file transfer from remote user.

Table 2-4 List of SSF and Dependent RPMs for Host (continued)

RPM	Description
<code>Syslcu tar file</code>	Contains scripts and configuration files of sysLCU utility.
<code>psmisc-22.6-19.el6_5.x86_64.rpm</code>	Provides utilities for installing ViewCheck on PCIE-7410, PCIE-9205, and PCIE-7210.
HTTP-2.2.29 package	A compiled version of http that supports Zend.
Zend-framework for SSF GUI	Zend with PHP 5.3.

2.2 Installing SSF

2.2.1 Installing SSF on PCIE-7410 and PCIE-9205 Cards

SSF and BBS comes pre-installed on these PCIE cards. If you want to reinstall latest version, use the `install_disk.img` provided in the release. SSF is a part of disk's `rootfs` and it gets installed along with the BBS and OS. For more on how to install `install_disk.img`, refer to *MaxCore™ MC3000 Platform Installation and Use* manual.

SSF automatically identifies the location of these cards placed in a MaxCore shelf and enables only the required services based on their role (whether a shelfHost card or an application Host card).

2.2.2 Installing SSF on PCIE-7210 Card

Before installing SSF, install BBS on PCIE-7210 card. For more information on how to install BBS, refer to *SharpStreamerPro PCIE-7210 Installation and Use* manual.

After installing BBS, follow the below procedure to install SSF on PCIE-7210 card:

1. Copy the `mc3ksw_update_7210_<os>-<os_ver>_<ver>.iso` image to the card.
2. Create a directory to mount the image.

```
#mkdir -p /mnt/ssf-update
```

3. Mount the image.

```
#mount -o loop mc3ksw_update_<7210>_<os>-<os_ver>_<ver>.iso /mnt/ssf-update
```

4. Install SSF.

1. Run `mc3k_<7210>_software_update.sh` script.

```
#cd /mnt/ssf-update/SSF  
#./mc3k_7210_software_update.sh
```

2. Enter **1** to install the software.
3. Verify the installed RPMs.

```
#rpm -qa | grep ssf
```

5. Unmount `ssf-update` and reboot the card.

```
#umount /mnt/ssf-update  
#reboot
```

NOTICE

If the RPM database is corrupted (Error: rpmdb open failed), you can rebuild it by using `rpm --rebuilddb` command as follows:

```
#rm -f /var/lib/rpm/___db*  
#db_verify /var/lib/rpm/Packages  
#rpm --rebuilddb  
#yum clean all
```

2.2.3 Installing SSF Agents for Target PCIE Cards

Artesyn provides SharpStreamer PCIE-7207 and SharpMedia PCIE-8120 target PCIE cards, which are compatible with MaxCore.

Prerequisite

Before software on these cards, ensure that related software of these target cards is already installed on the host card, currently acting as a shelfHost.

To install SSF Agents for target PCIE cards on host card (PCIE-7410, PCIE-9205, or PCIE-7210):

1. Copy the `mc3ksw_update_<7410 | 9205 | 7210>_<os>-<os_ver>_<ver>.iso` image to the host card.

2. Create a directory to mount the image.

```
#mkdir -p /mnt/ssf-update
```

3. Mount the image.

```
#mount -o loop mc3ksw_update_<7410 | 9205 | 7210>_<os>-<os_ver>_<ver>.iso /mnt/ssfupdate
```

To install software on a PCIE-7207 card:

1. Run `mc3k_7207_software_update.sh` script to upgrade SSF on PCIE-7207.

```
#cd /mnt/ssf-update
#./mc3k_7207_software_update.sh
```

2. Enter **1** to install the software.

3. Verify the installed RPMs.

```
#rpm -qa | grep 7207
```

4. Unmount `ssf-update`

```
#umount /mnt/ssf-update
```

To install software on a PCIE-8120 card:

1. Run `mc3k_8120_software_update.sh` script to upgrade SSF on PCIE-8120.

```
#cd /mnt/ssf-update
#./mc3k_8120_software_update.sh
```

2. Enter **1** to install the software.

3. Verify the installed RPMs.

```
#rpm -qa | grep 8120
```

4. Unmount `ssf-update`

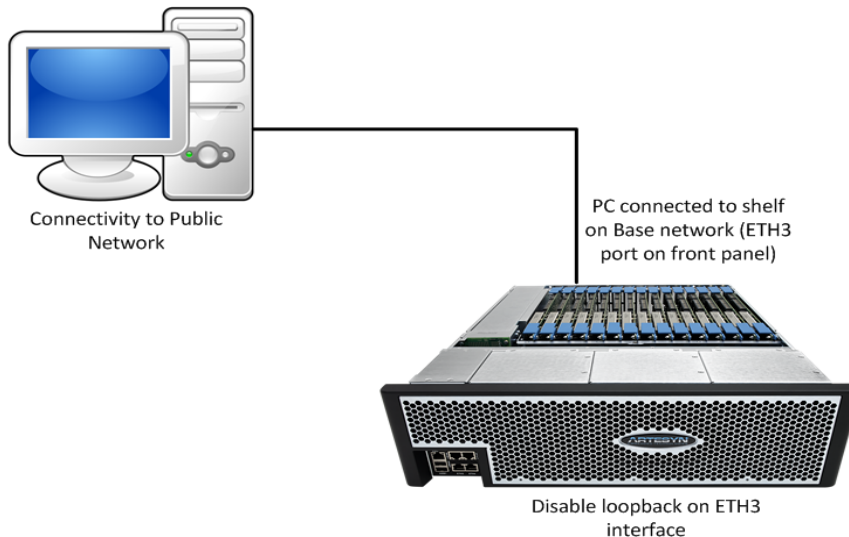
```
#umount /mnt/ssf-update
```

5. Edit the configuration file `/opt/ssf/etc/config/smc/ssfApi.conf` to populate the correct Shelf-Host IP-address. The IP-address will be in "172.27.<shelf-id>.2" format.

2.3 Installing SSF with SSF Core on External PC

In this configuration, SSF core and SSF GUI agent runs on a external PC. But, the SSF application agents will run as usual on respective cards. To manage the system, SSF core needs IP connectivity to the SSF application agents. SSF can be installed on x86_64 based PC. The PC should be reachable to base network, so that SSF core running on PC can be reachable for system management. [Figure 2-1](#) illustrates how an external PC is connected to MaxCore system. A complete package which includes dependent packages, SSF packages and installation script is provided with SSF release. Following sections will explain the steps to install SSF with SSF core on connected PC in details.

Figure 2-1 External PC based System Manager



2.3.1 Installing Linux on connected PC

A complete installation packages `mc3ksw_update_PC_centos-<os_version>_<version>.iso` is provided with the SSF release. SSF supports following Linux OS distribution.

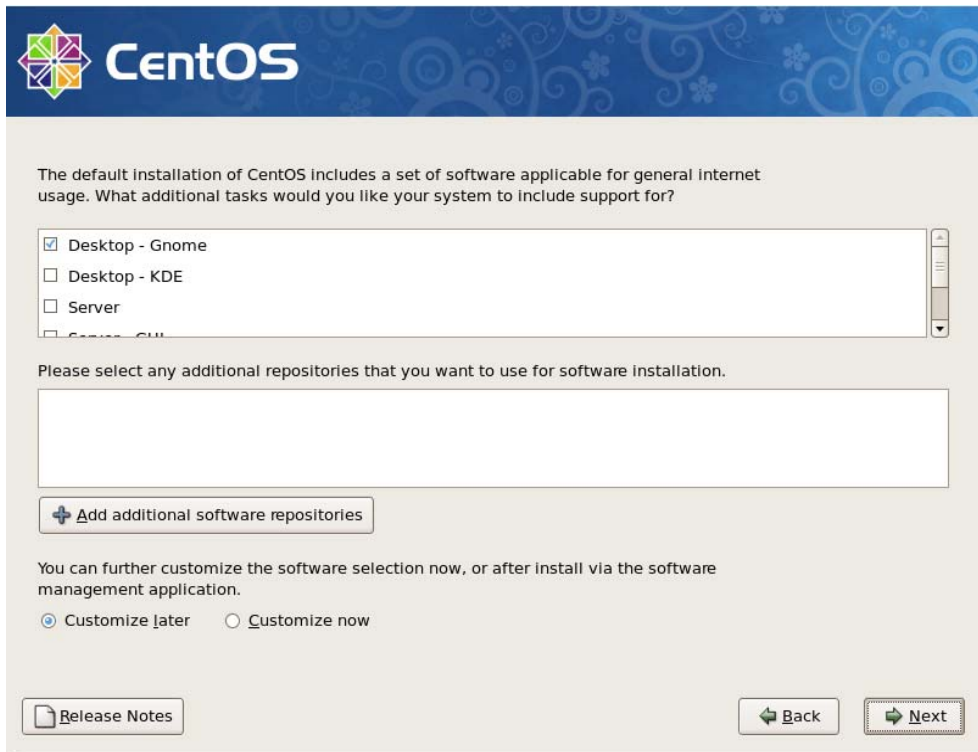
- CentOS release 7.2 on 64-bit platform

2.3.1.1 Installing CentOS on Connected PC

This section describes installing CentOS on external PC.

- Install CentOS from either of the locations mentioned in the above section and select Desktop option.

Note: Do not upgrade the Linux installation using any automatic upgrade tool (for example, yum) after installation. Any such upgrade can lead to failure of installation of SSF Core. You may perform such upgrade after the installation of SSF. For CentOS installation, select “Desktop-Gnome” option, when prompted for the type of install option as shown in the figure below. Any other selection can result into some dependency issues, which can lead to failure of smooth installation.



Desktop
 Minimal Desktop
 Minimal
 Basic Server
 Database Server
 Web Server
 Virtual Host
 Software Development Workstation

Please select any additional repositories that you want to use for software installation.

CentOS

You can further customize the software selection now, or after install via the software management application.

Customize later
 Customize now

2.3.2 Updating Interface Name of External PC

Interface name of external PC should be br0/ETH3. To change the Interface name follow the below steps:

1. Find out the MAC address of the network device.

```
#ifconfig enp63s0
enp63s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.130.100.98 netmask 255.255.255.0 broadcast
    10.130.100.255
    inet6 fe80::216:35ff:fe60:daa7 prefixlen 64 scopeid
    0x20<link>
    ether 00:16:35:60:da:a7 txqueuelen 1000 (Ethernet)
    RX packets 303760 bytes 419193497 (399.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
```

```
TX packets 170123 bytes 15587165 (14.8 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
device interrupt 17
```

In this example, MAC address is 00:16:35:60:da:a7

2. Edit the file `/etc/udev/rules.d/70-persistent-net.rules`
Add names of the network devices listed in this file as follows, and replace the `xx:xx:xx:xx:xx:xx` with the proper MAC address.

```
# interface with MAC address "xx:xx:xx:xx:xx:xx" will be assigned
"ETH3"
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
ATTR{address}=="xx:xx:xx:xx:xx:xx", ATTR{dev_id}=="0x0",
ATTR{type}=="1", KERNEL=="enp*", NAME="ETH3"
```
3. Rename the network interface configuration file.

```
# cd /etc/sysconfig/network-scripts/
# mv ifcfg-enp63s0 ifcfg-ETH3
```
4. Edit the network interface configuration file and replace all occurrences of the old name `enp63s0` with the new one `ETH3`.

```
# vi /etc/sysconfig/network-scripts/ifcfg-ETH3
```
5. Reboot the system to test changes.

```
# reboot
```
6. To verify new settings

```
# ifconfig -a
```

2.3.3 Installing SSF

This section describes installing SSF on external PC.

Note: Make sure that Linux OS is already installed on the external PC before installing SSF.

1. Copy the `mc3ksw_update_PC_<os>-<os_ver>_<ver>.iso` image to the Linux PC.
2. Create a directory to mount the image.

```
#mkdir -p /mnt/ssf-update
```
3. Mount the image.

```
#mount -o loop mc3ksw_update_PC_<os>-<os_ver>_<ver>.iso
/mnt/ssf-update
```

To install software on a PC:

- a. Run `mc3k_pc_software_update.sh` script to install/upgrade SSF on PC#`cd /mnt/ssf-update`
`#!/mc3k_pc_software_update.sh`
- b. Enter 1 to install the software.
- c. Verify the installed RPMs.
`#rpm -qa | grep ssf`
- d. Unmount `ssf-update` and reboot the PC.
`#umount /mnt/ssf-update`
`#reboot`

2.3.4 Configuring System Network

Before configuring the system network, make sure that, your PC is connected to the ETH3 port of MaxCore, which enables PC to communicate with all the boards through Base network.

1. IP Address assignment
 By default, SSF is configured to listen on all interfaces of its Host.
 All the application SSF agents are configured to contact SSF core on IP address set through MCparams. Assign the static IP address `172.27.0.1` to the port of PC connected to the Base Network (connected to ETH3).
2. MaxCore configuration procedure: The file `/opt/ssf/etc/config/main/maxcore.conf` on the desktop hosting SSF core, should be modified with the listening interface that is the port of desktop connected to the base network.

```
log_level=INFO
ssfhostinterface=ETH3 <- interface name of the desktop port
connected to the base network
applicationhost_apply_policy=true
default_configuration=true
domain=11                # Domain ID
rack=1                   # Rack ID
shelf=75
name=Shelf               # Shelf Name
master=true              # shelf type
```

```
mcpu_ipaddr=127.0.0.1    <- shelfHost IP address, which is
reachable from external Host
mcpu_port=8888           # MCPU Port
mcpu_evt_port=8890      # MCPU Evt Port
```

Note: Make sure that the physical connection to PC from MaxCore is via ETH3 of front panel.

3. Disable loopback on ETH3 interface on MaxCore.
`#mccs_tool.py --method set-loopback --func 16,2 --mode off`
4. Restart SSF on the PC:
`#systemctl restart ssfCore.service`
Power cycle the connected MaxCore Shelf.

2.3.5 Uninstallation Procedure on External Linux PC

This section explains how to uninstall the package on the Linux OS distribution:

1. Copy the `mc3ksw_update_PC_<os>-<os_ver>_<ver>.iso` image to the Linux PC.
2. Create a directory to mount the image.
`#mkdir -p /mnt/ssf-update`
3. Mount the image.
`#mount -o loop mc3ksw_update_PC_<os>-<os_ver>_<ver>.iso /mnt/ssf-update`

To install software on a PC:

- a. Run `mc3k_pc_software_update.sh` script to install/upgrade SSF on PC
`#cd /mnt/ssf-update`
`#!/mc3k_pc_software_update.sh`
- b. Enter 2 to Uninstall the software.
- c. Verify the installed RPMs.
`#rpm -qa | grep ssf`
- d. Unmount `ssf-update` and reboot the PC.
`#umount /mnt/ssf-update`
`#reboot`

2.3.6 Upgrade Procedure on External Linux PC

This section explains how to upgrade the package from one release version to another release version on the Linux OS distribution:

1. Copy the `mc3ksw_update_PC_<os>-<os_ver>_<ver>.iso` image to the Linux PC.

2. Create a directory to mount the image.

```
#mkdir -p /mnt/ssf-update
```

3. Mount the image.

```
#mount -o loop mc3ksw_update_PC_<os>-<os_ver>_<ver>.iso  
/mnt/ssf-update
```

To install software on a PC:

a. Run `mc3k_pc_software_update.sh` script to install/upgrade SSF on PC

```
#cd /mnt/ssf-update  
#./mc3k_pc_software_update.sh
```

b. Enter 3 to upgrade the software.

c. Verify the installed RPMs.

```
#rpm -qa | grep ssf
```

d. Unmount `ssf-update` and reboot the PC.

```
#umount /mnt/ssf-update  
#reboot
```

2.3.7 Starting SSF on External PC

After completion of the installation, the postgres, radius, and SSF services will be started automatically.

This section describes how to start the SSF Core manually.

2.3.7.1 Starting SSF core on External PC

After completion of the installation, the postgres, radius, and SSF services will be started automatically.

Use the following commands to start, stop, status and restart the postgres, radius and SSF services for future use:

1. `systemctl start ssfCore.service`
2. `systemctl stop ssfCore.service`
3. `systemctl status ssfCore.service`
4. `systemctl restart ssfCore.service`

2.4 Upgrading Shelf using System Update

SSF supports MaxCore System Update from release 1.1.0.28 (SP5). Using this feature, you can update a complete shelf with a single command.

SSF for MaxCore release package contains a System Update package, which includes upgrades for all Artesyn Cards (PCIE-7410, PCIE-7210, and PCIE-9205) for MaxCore platform. You can use System Update using CLI and XML interface. The following section describes the procedure in detail.

NOTICE

If any of the applicationHost CPU is booted with network boot, then boot that CPU with latest ramdisk image to avoid any upgrade to be initiated.

To perform System Update, you have to connect to SSF either through CLI or XML interface through Serial access or over IP. For more information, refer to sections [Accessing SSF using XML Interface on page 51](#) and [Accessing SSF using CLI on page 51](#).

By default the credential are,

Username: *Admin* and **Password:** *Admin*

2.4.1 Checking Shelf Status

System Update will upgrade the shelfHost and all participating applicationHost CPUs on the shelf. An applicationHost can only participate in System Update, if upgrade agent (part of SSF agent) is available and connected to the shelfHost. This command lists the applicationHost CPUs which are participating in the System Update. Verify all the applicationHost CPUs that are listed in the output of the command with current version.

To check the current versions of all the components available in the shelf and to view the update status, use the status command as shown below.

```
[root@pcie9205-s1-c1 ~] telnet localhost 11001
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to SSF CLI
Username: Admin
Password:
Access granted
>en
con t
MaxCore(config)system 1
MaxCore(system-1)shelf 22
MaxCore(shelf-1-22)system-upgrade-status
=====
Shelf# 22          status: Upgrade Success
=====
Slot# 1
CPU# 1  status: Upgrade Success
-----
Entity      Status      Curr Ver      Last Updated
```

```

-----
BMC_Application          1.8.000000          Sun Mar 12 12:42:41 IST 2017
BMC_CPLD_Companion      0.03.03000000      Sun Mar 12 12:42:41 IST 2017
BMC_CPLD_Glue           0.03.00000000      Sun Mar 12 12:42:41 IST 2017
BMC_CPLD_I/O_Module     0.06.01000000      Sun Mar 12 12:42:41 IST 2017
Basic_Board_Services    1.5.0              Sun Mar 12 12:42:41 IST 2017
System_Services_Framework Upgrade Success  1.1.0.29 Sun Mar 12 13:42:53 IST 2017
PCIe-9205_BIOS 1.4.00000002          Sun Mar 12 12:42:41 IST 2017
PCIe-9205_CPLD          01.00.00          Sun Mar 12 12:42:41 IST 2017
PCIe-9205_Switch_Management Upgrade Success          1.0.2.24          Sun
Mar 12 13:45:36 IST 2017
ViewCheck Upgrade Success 1.0.2.15 Sun Mar 12 13:48:06 IST 2017

Slot# 2
Slot# 3
Slot# 4
Slot# 5
Slot# 6

CPU# 1 status:  --
-----
Entity          Status          Curr Ver          Last Updated
-----
CPU# 2 status: Upgrade Success
-----
Entity          Status          Curr Ver          Last Updated
-----
Basic_Board_Services 1.5.0          Sun Mar 12 13:29:53 IST 2017
System_Services_Framework Upgrade Success 1.1.0.29          Sun Mar 12 14:07:23
IST 2017
PCIe-7410_BIOS 1.4.00000002          Sun Mar 12 13:29:53 IST 2017
PCIe-7410_CPLD          01.00.00          Sun Mar 12 13:29:53 IST 2017
ViewCheck Upgrade Success 1.1.0.15 Sun Mar 12 14:07:50 IST 2017

```



```

Slot# 7
Slot# 8
Slot# 9
  CPU# 1  status:      --
  -----
  Entity   Status                Curr Ver   Last Updated
  -----
  CPU# 2  status: Upgrade Success
  -----
  Entity   Status                Curr Ver   Last Updated
  -----

Slot# 10
Slot# 11
Slot# 12
Slot# 13
Slot# 14
Slot# 15
  CPU# 1  status:      --
  -----
  Entity   Status                Curr Ver   Last Updated
  -----
  CPU# 2  status:      --
  -----
  Entity   Status                Curr Ver   Last Updated
  -----
  
```

In the above output, you can see that Slot6- CPU1, Slot9-CPU1, Slot9-CPU2 Slot15-CPU1 and Slot15-CPU2 are not participating in System Update but Slot6-CPU2 is participating.

2.4.2 Initiating System Update

To perform System Update, perform the following steps.

1. Download the System Update package from the delivery to the ShelfHost of the MaxCore for which you want to initiate the update.
2. Connect to SSF either through CLI or XML interface through Serial access or over IP.
3. Run the following command.

```
MaxCore(shelf-1-22)#system-upgrade-initiate filename "<System Update file with Absolute Path>"
```

This will initiate the upgrade on the shelf which include multiple power cycles and reboots

For more information about system update command, refer to *SSF for MaxCore MC3000 Platform XML Interface Guide* and *SSF for MaxCore MC3000 Platform Command Line Interface Guide*.

2.5 Upgrading SSF

This section describes steps to perform SSF software upgrade on PCIE-7410, PCIE-7210, and PCIE-9205 cards individually.

The mc3ksw_update_<7410 | 7210 | 9205>_<os>-<os_ver>_<ver>.iso image contains BBS and SSF repositories that can be used for upgrading SSF on PCIE-7410, PCIE-7210, or PCIE-9205 cards.

The ISO file contains the following repositories for PCIE-7410, PCIE-7210, PCIE-9205, PCIE-7207, and PCIE-8120 cards.

Table 2-5 Software Update Package Contents

ISO Contents	Description
mc3k_<7410 7210 9205>_software_update.sh	PCIE-7410, PCIE-7210, or PCIE-9205 update script.
mc3k_7207_software_update.sh	PCIE-7207 update script.
mc3k_8120_software_update.sh	PCIE-8120 update script.

Table 2-5 Software Update Package Contents (continued)

ISO Contents	Description
ssf<7410 7210 9205>	PCIE-7410, PCIE-7210 or PCIE-9205 yum repository. The PCIE-9205 yum repository includes RRC-TLS.
ssf7207	PCIE-7207 yum repository.
ssf8120	PCIE-8120 yum repository.
ssf<7410 7210 9205>.repo	PCIE-7410, PCIE-7210 or PCIE-9205 yum repository file.
ssf7207.repo	PCIE-7207 yum repository file.
ssf8120.repo	PCIE-8120 yum repository file.
zend-default-vhost-80.conf	Zend server configuration file.
updatePCIE-7207-OS.sh	Ramdisk image creation file.
bbs-release	BBS release file.

NOTICE

If you are running SSF <7410 | 7210 | 9205> 1.1.0.12 you can upgrade to higher version using the mc3ksw_update_<7410 | 7210 | 9205>_centos-7.2_<greater than 12>.iso image. SSF contains a yum repository with new packages.

To upgrade SSF on PCIE-7210, refer [Installing SSF on PCIE-7210 Card on page 27](#).

Upgrading SSF

1. Copy the mc3ksw_update_<7410 | 7210 | 9205>_<os>-<os_ver>_<ver>.iso image to the card.
2. Create a directory for mounting the image.

```
#mkdir -p /mnt/ssf-update
```

3. Mount the image.

```
#mount -o loop mc3ksw_update_<7410 | 7210 | 9205>_<os>-<os_ver>_<ver>.iso  
/mnt/ssf-update
```

4. Upgrade the software:

1. Run `mc3k_<7410 | 7210 | 9205>_software_update.sh` script.

```
#cd /mnt/ssf-update/SSF  
#./mc3k_7410_software_update.sh
```

2. Enter **3** to upgrade the software.
3. Verify the installed RPMs.

```
#rpm -qa | grep ssf
```

Note: If you do not want to install SSF agents for target PCIE cards (PCIE-7207 and PCIE-8210), go to step 7.

5. To upgrade SharpStreamer PCIE-7207:

1. Run `mc3k_7207_software_update.sh` script to upgrade SSF on SharpStreamer PCIE-7207.

```
#cd /mnt/ssf-update  
#./mc3k_7207_software_update.sh
```

2. Enter **3** to upgrade the software.
3. Verify the installed RPMs.

```
#rpm -qa | grep 7207
```

6. To upgrade SharpMedia PCIE-8120:

1. Run the `mc3k_8120_software_update.sh` script to upgrade SSF on PCIE-8120.

```
#cd /mnt/ssf-update  
#./mc3k_8120_software_update.sh
```

2. Enter **3** to upgrade the software.

3. Verify the installed RPMs.

```
#rpm -qa | grep smcTLS
```

7. Unmount `ssf-update` and reboot the card.

```
#umount /mnt/ssf-update  
#reboot
```

While installation / Upgrade if you observe the below issue, execute the commands provided in the resolution section.

Reinitiate the installation / Upgrade once the database is recovered.



For network booted hosts:

1. If a CPU is net (PXE) booted with a ramdisk image of an older version of SSF, it will not get upgraded even if it appears to be upgrading in `system-upgrade-status` as it will boot up from the original image on next reboot. You need to replace the older ramdisk image with the newer version image and boot from it to get the latest version of software.
2. For versions 1.1.0.28 and older, if the current shelf ID is different from the what is present in the configuration in original ramdisk image, `ssfAgent` will not be able to connect to shelf host after it is rebooted during the system upgrade. Hence, it may not show up in `system-upgrade-status`. From version 1.1.0.34, the `ssfAgent` on a network booted CPU will be able to connect to Shelf Host irrespective of the current shelf ID.

SSF Service Configuration

3.1 Starting SSF

SSF starts automatically once you restart the system after its installation and configuration. If you want to start manually, then the following sections describe how to start SSF Core, SSF Agent, Application agent individually, and SSF web interface.

3.1.1 Starting SSF Core

You can stop and start the SSF Services using the `systemd` scripts.

After you start the SSF Core, you can stop, restart, and check the status of the SSF using the following command.

```
#systemctl {stop|restart|status|enable|disable} ssfCore.service
```

3.1.2 Starting SSF Agent

You can stop, restart, and check the status of the SSF Agent using the following command.

```
#systemctl {stop|restart|status|enable|disable} ssfAgent.service
```

3.2 Managing SSF MaxCore Agent

The SSF MaxCore agent service, `mcagent.service` is managed using `systemd`. By default, it will start at OS boot-up.

```
#systemctl status mcagent.service
```


After installing and configuring SSF, you can access it using the web interface, CLI, and XML to manage the system.

4.1 Accessing SSF using Web Interface

You can access SSF using the web interface for configuring, managing, and monitoring MaxCore platform equipped with multiple resources. You can use any of the following browsers to log on and access the SSF:

- Internet Explorer version 10.0 and later.
- Mozilla Firefox 12.0 and later.
- Google Chrome version 23 and later.

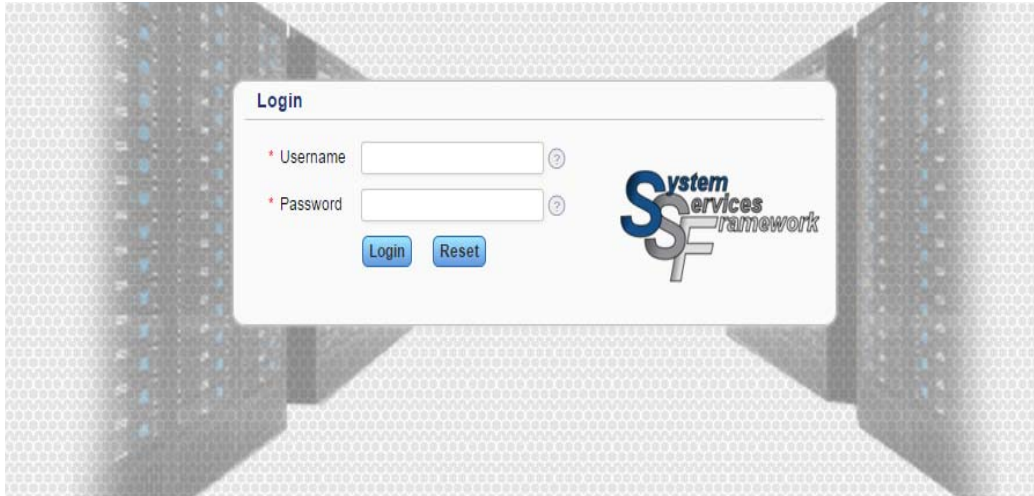
Note: Sometimes after installation or due to some timeout, you may get a blank screen. Cleanup browser cache (CTRL + SHFT+ DEL) and reconnect the SSF using web interface. This is a onetime activity.

To access SSF using Web interface

1. Type the SSF web application URL in the address bar of the web browser and then press **<Enter>**. You can obtain the SSF URL from your administrator. The **SSF Login** dialog box opens, as shown in [Figure 4-1](#).

For example, `https://<IP Address>:<PortNumber>`

Figure 4-1 Login Page



2. In the **Login** dialog box, type your *User name* and *password*.



The default user name and password is “Admin”.

3. Click **Login** for logging into **SSF**. The SSF Home page opens.

For more information on Web interface, refer to Online Help integrated with the SSF application. Click the **Help** icon to access the Online help.

NOTICE

Make sure that the SSF Web Interface is initiated before accessing it. In order to access SSF GUI, the SSF discovery/initialization should be completed.

4.2 Accessing SSF using XML Interface

The XML interface of SSF passes management requests to the SSF framework for processing. It also handles responses and notifications/events from the SSF framework.

The XML interface facilitates access to SSF using an XML-based request protocol. The XML interface is intended for remote configuration of software and scripts. It can also be used via a remote GUI configuration tool. The XML-based requests are sent over a persistent connection to the XML agent, which processes the requests and returns XML-based responses. SSF also sends asynchronous responses/events over the XML interface such as alarm notifications, etc. By default, these events are disabled. For more information about the XML commands, refer to the *System Services Framework for MaxCore™ Platform XML Interface Guide*.

4.3 Accessing SSF using CLI

You can access SSF using the CLI. SSF provides a fully functional CLI.

To access SSF using CLI

1. Establish a secure shell connection to SSF host using SSH.
2. Start the **telnet** connection from an already established secure shell.

```
root@localhost ~]# telnet localhost 11001
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to Maxcore CLI
```

3. Type your *user name* and *password*.

```
Username: Admin
Password:
Access granted

>enable
#configure terminal
Maxcore(config)#
```

NOTICE

By default, the administrator user name and password are “Admin”. To change the existing password, see [Appendix A, Additional Information, on page 57](#).

4.4 Accessing SSF using SNMP

The SNMP is designed to provide a means of managing and monitoring diverse network devices. It has client-server architecture and uses unencrypted text known as community strings for authentication. Communication between the client and server is accomplished using a command. There are four commonly used commands: `snmpget`, `snmpset`, `snmpwalk`, and `snmptrapd` for receiving trap message.

You can access SSF on PCIE card using port number 10165, private enterprise number 26061 and text file `MAX-CORE-MIB` which will be available in `/usr/share/snmp/mibs` directory.

Example for SNMPget

```
#snmpget -v2c -cprivate -m /usr/share/snmp/mibs/MAX-CORE-MIB
snmp_agent_Ip_address:10165 systemName.1
```

```
MAX-CORE-MIB::systemName.1 = STRING: "MaxCore System Framework"
```

Example for SNMP walk,

```
#snmpwalk -m /usr/share/snmp/mibs/MAX-CORE-MIB -v2c -c private
snmp_agent_Ip_address:10165 1.3.6.1.4.1.26061
```

```

MAX-CORE-MIB::systemInfo.1 = STRING: "System Services Framework for
Configuring MaxCore shelves"

MAX-CORE-MIB::systemName.1 = STRING: "MaxCore System Framework"

MAX-CORE-MIB::maxNoEvents.1 = Gauge32: 1000000

MAX-CORE-MIB::eventFilterSeverity.1 = Gauge32: 1

MAX-CORE-MIB::eventFilterType.1 = Gauge32: 32

MAX-CORE-MIB::userConfig.1 = Gauge32: 0

MAX-CORE-MIB::shelfName.1.1 = STRING: "Maxcore (r1.s1.a1.b1)"

MAX-CORE-MIB::shelfAddr.1.1 = STRING: "r1.s1.a1.b1"

MAX-CORE-MIB::shelfInventoryInfo.1.1 = STRING: "Vendor: ARTESYN,
Product: MAXCore"

...

```

Example for SNMP set

```

#snmpset -m /usr/share/snmp/mibs/MAX-CORE-MIB -v2c - cprivate
snmp_agent_ip_address :10165 systemName.1 s "MAXCORE"

MAX-CORE-MIB::systemName.1 = STRING: "MAXCORE"

```

Example for SNMP trap

You can configure the PCIE card using the CLI to send notifications to SNMP managers as traps.

```

#snmp-server host snmp_traphost_ip_address trap version 2c SSF udp-
port 162 Admin

```

To receive traps, start `snmptrapd` application on configured manager that listen at port 162 and notifies the traps.

```

#snmptrapd -f -Lo -m MAX-CORE-MIB

```


Configuration Management

SSF provides a set of predefined configuration candidates. The changes you perform on configuration candidates are stored in database and not applied to the real hardware. This is called emulator mode. In this mode, you can work on more than one MaxCore configuration, called configuration candidate. By default, no configuration is available in these candidates. You can store configurations in these candidates and use them whenever you require. You can apply any of these configurations candidates to the real hardware.

You can perform the following tasks under Configuration Management:

- **Copy Configuration:** To copy a configuration either from the current running hardware configuration or from configuration candidates or from external configuration files.
- **Switch to emulator candidate:** To view and modify the configuration in a candidate, you have to switch to the emulator mode by selecting the desired candidate.
- **Apply active candidate to hardware:** To apply the selected configuration to hardware.
- **Clear configuration:** To remove the configuration of a candidate.
- **Save a copy of configuration to a file:** To save the configuration of a candidate to a file.
- **Reload the saved Configuration from a file:** To load a configuration file on to the hardware or to a candidate.

For more information, refer to *SSF for MaxCore MC3000 Platform GUI Help*.

5.1 Adding or Remove Emulator Mode Candidate Names

The following are the list of predefined configuration candidates provided by SSF:

- ssfConfig1
- ssfConfig2
- ssfConfig3
- ssfConfig4
- startup

SSF allows you add new configuration candidates or remove existing configuration candidates.

5.1.1 Adding a Configuration Candidate

To add a configuration candidate:

1. Enter the candidate name in `/opt/ssf/etc/config/main/ssf_config.ini` file in the following format.

```
Configuration<number> = <CandidateName>
```

For example, `Configuration6 = ssfconfig1`

The left side text must start with term `Configuration` followed by a number.

For example, `Configuration6`. The number should be continuous in an incremental order.

2. Update `MaxConfig` value based on number of candidates available in the system. Make sure that the value of `MaxConfig` is greater or equal to number of configured candidates.
3. Restart SSF Core service. A new configuration candidate is added with . This will not affect configuration in existing candidates.

5.1.2 Removing a Configuration Candidate

To remove a configuration candidate:

1. Remove the line containing respective configuration candidate from the `/opt/ssf/etc/config/main/ssf_config.ini`
2. Restart SSF Core service.

NOTICE

This will not affect configuration in existing candidates, candidate is not shown by SSF, but it will still exist in the database. To delete a candidate from database, remove `/opt/ssf/data/` and then restart SSF Core Service.

Additional Information

A.1 Resetting Administrator Password

To reset the administrator password:

1. Log on to the card on which SSF and PostgreSQL are installed.
2. Log on as a PostgreSQL user with the `# su -l postgres` command.
3. Connect to the PostgreSQL database using the below command

```
#!/usr/local/postgres/bin/psql SSF
```

 The following output is displayed.

```
psql (9.1.3)
Type "help" for help.
SSF=#
```
4. List the available users, using the `SSF=# select * from "user";` command. The list of available users along with the passwords are displayed as shown below.

```
user | password | hash
-----+-----+-----
Admin | Admin    |
(1 row)

SSF=#
```
5. Reset the administrator password using the following command.

```
SSF=# update "user" set password = 'test' where "user" = 'Admin';
UPDATE 1
SSF=#
```

After changing the password of the administrator, you can check whether the new password is reflected or not by listing the available users using the `SSF=# select * from "user";` command. The following output is displayed.

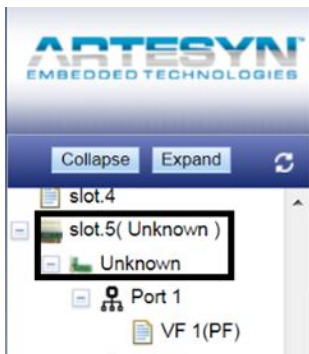
```
SSF=#
SSF=# select * from "user";
user | password | hash
-----+-----+-----
Admin | test     |
(1 row)

SSF=#
```

A.2 Configuring a Non-Artesynt card for SSF

If a non-Artesynt card is inserted into the MaxCore chassis, the card information will be shown as **Unknown** in the SSF GUI. The following figure shows a sample SSF GUI view when a non-Artesynt card is inserted into a MaxCore system, for example, assume that a non Artesynt card is inserted in slot 5.

Figure A-1 Non-Artesynt Card in MaxCore system - Before Configuration



To view the card information in the SSF GUI, you need to perform the following steps:

1. Obtain the vendor and device details of the card using **mccs_tool.py** tool. For this you need to run the following command on ShelfHost.

```
#mccs_tool.py --method=list-devices
```

Note: Only the output of the card in slot 5 is shown below for quick reference.

```
slot: 5 device: 1
  func: 1
    vendor      : 0x8086
    device      : 0x15a4
    class       : 0x20000
  pci location@mcpu : 2a:00.0
  plx switch     : 1
  plx station    : 4
  plx port      : 16
```

2. Add the card details in `/opt/ssf/etc/config/main/mxc_supported_cardinfo.ini` file in the following format.

```
<CardName>::<Vendor>::<Device>::<DeviceNumber>::<DeviceName>::<S_Device (optional)>
```

For example, `XYZ::0x8086::0x15a4::1::XYZ-1::2023`

Make sure to add details of all the devices available on the card.

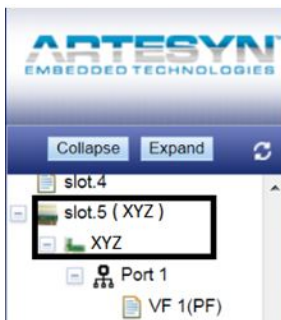
Note: The card output shows the vendor and device details as `0x8086` and `0x15a4` respectively.

3. Restart the SSF hardware agent.

```
#/opt/ssf/etc/config/s99mcagent.sh restart
```

After completion of the process, the name of the card with its details are shown in SSF GUI as shown in the following figure.

Figure A-2 Non-Artesyn Card in MaxCore system - After Configuration



A.3 Updating Shelf Address

SSF allows you update the MaxCore shelf address through GUI. Shelf address is represented with the following convention.

```
r1.s<ID>.<Location/Address>
```

`r1` denotes the rack number. Currently, this is not user configurable and is kept for future use. `ID` denotes the shelf number and `Location/Address` denotes a user-configurable text. You are allowed to modify and update these details for unique identification of your shelf.

You can update either Shelf ID or Location/Address or both.

NOTICE

Shelf ID change in **Shelf Address change will trigger a Shelf Host reboot. The booting of Shelf Host results to power reset of Application Hosts. It is STRONGLY SUGGESTED to shut down all Application Hosts gracefully before proceeding. You can use "Power Off All App Hosts" button on the System > MaxCore > Overview screen in GUI to perform this operation.**

For more information on how to update shelf address, refer to SSF for MaxCore MC3000 Platform GUI Online Help.

A.4 Network Boot Configuration

SSF allows you to enable or disable network boot configuration of applicationHost (s). The Network Boot Configuration dialog box shows the status of availability of images (Ramdisk and Kernel) of CPU for that specific card (PCIE-7410, PCIE-9205, and PCIE-7210). If any of these images are not available or you want to replace the available image with a new image, you need to upload the required image.

For more details, see *Network Boot Configuration* Section in *SSF for MaxCore MC3000 GUI help*.

A.5 Setting Default Configuration

It allows you to set the default assignments, ETH3 and ETH4 virtual functions, to applicationHosts. The pre-defined virtual functions are automatically assigned to the available applicationHosts and thus bringing applicationHosts into internal networking.

NOTICE

You do not need to assign these virtual functions to applicationHost manually. This operation will not disturb any other assignments made from other endpoint devices.

A.6 Enabling Network Time Protocol

Using this feature you can enable or disable SSF Configured NTP (Network Time Protocol) Service. SSF provides this feature at the CPU level of a MaxCore system. This feature allows you to synchronize the date and time of application hosts with the shelfHost.

To enable or disable this feature, click Off/On control button in line with SSF Configured NTP Service label. This is a toggle button. When this service is enabled (ON), the time stamp of all the application hosts running in the MaxCore gets aligned with shelfHost running time and when this service is disabled (OFF), the respective application hosts will be running independently irrespective of shelfHost running time.

NOTICE

Disable *SSF Configured NTP Service*, if you want to synchronize any of the applicationHosts date and time with an external source other than shelfHost.

A.7 Configuring Multiple Shelves on MaxCore

To configure multiple shelves on MaxCore, follow the procedure below.

NOTICE

Assuming shelf Ids/chassis numbers for the shelf are already configured as mentioned in [Appendix A, Updating Shelf Address, on page 59](#).

On System Host (that hosts all the MaxCore shelves)

1. Check the current shelf's configuration using CLI or XML Command.
MaxCore(config-HardwarePlatformManager)#**listShelves**

```
Shelf: Shelf
      rackID: 1
      ShelfId: 1
      Name: Shelf
      shelfHostIpAddr: 127.0.0.1
```

```
isMaster: true;
```

Here, the shelfHost IP address is localhost (127.0.0.1) and the shelf id is 1.

2. Add a new Shelf using CLI or XML command.
The Shelf's ID and shelfHost IP Address should be different from that of earlier configured shelves, and shelfHost IP Address should be reachable to SSF core.

Syntax

```
MaxCore(config-HardwarePlatformManager)#addShelf shelfID  
<Shelf_Id> shelfHostIpAddr <Shelf_IP_Address> master false  
shelfName <Shelf_Name>
```

Example

```
MaxCore(config-HardwarePlatformManager)#addShelf shelfID 2  
shelfHostIpAddr 172.27.2.2 master false shelfName Shelf2
```

For more information about add shelf command, refer to *SSF for MaxCore MC3000 Platform XML Interface Guide* and *SSF for MaxCore MC3000 Platform Command Line Interface Guide*.

On Shelf Hosts (other than System Host)

1. Disable loop-back for ETH3 while connecting multiple shelves to a hub to be on default base network.

```
# mccs_tool.py --method=set-loopback --mode=off --func=16,2
```
2. Start and Stop following services.

```
#systemctl stop ssfCore.service  
#systemctl disable ssfCore.service  
#systemctl start mcagent.service  
#systemctl enable mcagent.service
```

On both System Hosts and Shelf Hosts

1. Restart all SSF services.
For more information on restarting SSF services, see sections [Starting SSF on page 47](#) and [Managing SSF MaxCore Agent on page 47](#).

A.8 Configuring firewalld to Allow SSF Communication

NOTICE

Assuming that the following configurations are present:

- **firewalld** is running both on SSF-Core and App-host.
- **firewalld** is already started while executing the commands. (Command: `systemctl start firewalld`)

Follow the below procedure to configure `firewalld` to allow SSF internal communication:

On App-host

1. Identify the ports for which firewall rule needs to be added. Run the below command.

```
# cat /opt/ssf/etc/config/bcsim/ssfApi.conf
```

Output:

```
# Transport type (one of: tcp, uds)
#transport=tcp
#listening address
LocalTcpAddress=0.0.0.0:21215  <-  firewall rule to be added for
this on App Host side
# embeddedMIND process location
emindTcpAddress=192.168.201.100:21212  <-  firewall rule to be
added for this on SSF Core side
#emindUdsPath=/tmp/emind-tl-uds
# Link health-check period (in seconds)
healthcheckPeriod=10.0
# Log settings
#logEnabled=yes
#logLevel=error # one of: error, info, debug
#logFile=eMindApi.log
[root@localhost ~]#
```

2. Add firewall rule on App Host. (port 21215 in this case). It allows traffic from SSF-Core to App-host.

```
#firewall-cmd --zone=public --add-port=21215/tcp
```

Change the port number in the above command as per your `ssfApi.conf` file.

On SSF Core-host

- Add firewall rule on the SSF Core host. All app-hosts usually connect to port 21212. So, this needs to be added for firewall exception.

```
#firewall-cmd --zone=public --add-port=21212/tcp
```

A.9 Verifying SSF and BBS Versions Installed on the System

For verifying SSF version execute below command.

```
#cat /etc/ssf-release
```

For verifying BBS version execute below command.

```
#cat /etc/blade-release
```

A.10 Changing Logging Configuration

By default, the log level configured is "info".

To change it to lower levels in order to avoid excessive logging, perform the following steps:

1. Login to SSF CLI and enter the config mode.

```
# telnet localhost 11001
Trying: 1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome to SSF CLI
Username: Admin
Password:
Access granted
>enable
```



```
#configure terminal
```

```
MaxCore(config)#
```

2. Goto `logfilter` class and select "syslog" instance and then enter `show` to get the current logging configurations.

```
MaxCore(config)#logfilter syslog
```

```
MaxCore(logfilter-syslog)#show
```

```
logfilter, syslog
```

```
type                = priority
```

```
priority            = info
```

```
modules             =
```

3. Enter priority from one of the following.

```
MaxCore(logfilter-syslog)#priority ?
```

```
critical  Filter priority (applicable for priority log filter)
```

```
[debug]
```

```
debug     Filter priority (applicable for priority log filter)
```

```
[debug]
```

```
error     Filter priority (applicable for priority log filter)
```

```
[debug]
```

```
info      Filter priority (applicable for priority log filter)
```

```
[debug]
```

```
warning   Filter priority (applicable for priority log filter)
```

```
[debug]
```

A.11 SSF Core Configuration

SSF core is a management and configuration interface between hardware and software. SSF core consists of two sets of configuration files. One for SSF-Core (`ssfMxcd`) and other for SSF-CSIM (`ssfcsimd`). All SSF-Core (`ssfMxcd`) executable configuration files are stored in `/opt/ssf/etc/config/main/`. You can edit the `ssf.ini`, `maxcore.conf`, and `tl.ini` files stored at this location to configure the required parameters.

All CSIM (`ssfcsimd`) executable configuration files are stored in `/opt/ssf/etc/config/csim/`. You can edit the `ssfApi.conf` file stored at this location to configure the required parameters. [Table A-1](#) describes the configuration options.



	 CAUTION
	<p>Misconfiguring any of the following may lead to an unusable system.</p>

Table A-1 SSF Core Configuration Files

File Name	Configuration Options	Description
<code>ssf.ini</code> Directory: <code>/opt/ssf/etc/config/main</code>	<code>MaxSessions</code>	The default value for <code>MaxSessions</code> is 100. You can edit it to any required value.
	<code>SessionTimeout</code>	The default value for <code>SessionTimeout</code> is 1800 seconds. You can edit it to any required value.
<code>maxcore.conf</code> Directory: <code>/opt/ssf/etc/config/main</code>	<code>log_level</code>	Log level INFO - Notifications and important information. DEBUG - Verbose Default is INFO.
	<code>domain</code>	MaxCore ID in the system. This may become obsolete.
	<code>rack</code>	Rack ID in the system.
	<code>shelf</code>	Shelf ID or chassis number to uniquely identify the MaxCore.
	<code>name</code>	Name of the MaxCore. This is optional.
	<code>mcpu_ipaddr</code>	Shelf Host IP address. The default IP is 127.0.0.1. You can edit it to any Shelf Host IP reachable by the system host.
	<code>mcpu_port</code>	TCP/IP port. The default value is 8888.
	<code>mcpu_evt_port</code>	TCP/IP port for events. The default value is 8890.
	Note: To access multiple MaxCores simultaneously, you can configure more than one domain in the same file.	

Table A-1 SSF Core Configuration Files (continued)

File Name	Configuration Options	Description
tl.ini Directory: /opt/ssf/etc/ config/main	transport	The default Transport is set as TCP. You can also change it to Unix Domain Sockets (UDS), if needed. SSF Server (SSF Host) listens on both TCP and UDS sockets for connection from the SSF agent.
	emindTcpAddress	By default, the emindTcpAddress is set to localhost. You can replace with the IP address of the SSF Core.
ssfApi.conf Directory: /opt/ssf/etc/co nfig/csim/	transport	The default Transport is set as TCP. You can also change it to UDS, if required. SSF Server (SSF Host) listens on both TCP and UDS sockets for connection from the SSF agent.
	emindTcpAddress	By default, the emindTcpAddress is set to localhost. You can edit with the IP address of the SSF Core.
	healthcheckPeriod	By default, the healthcheckPeriod is set to 1.0 seconds. In every 1.0 second the system checks the health link between CSIM core and SSF. You can modify this value to required duration.
	logEnabled	By default, the logEnabled field is enabled for log collection. You can modify it to disable log collection.
	logLevel	By default, the logLevel is set to error. You can modify to either error, info or debug.
	logFile	By default, logFile name is eMindApi.log file.

A.12 SSF Agent Configuration

Server SSF core consists of SSF-BCSIM (`ssfbcsimd`) executable configuration files. These files are stored in `/opt/ssf/etc/config/bcsim/`. You can edit the `ssfApi.conf` file stored at this location to configure the required parameters. The following table provides the list of configuration options of `ssfApi.conf` file.

Table A-2 SSF Server Configuration Files

File Name	Configuration Options	Description
ssfApi.conf	transport	The default transport is set as TCP. You can also change it to UDP if needed.
	emindTcpAddress	By default, the emindTcpAddress is 172.27.1.2. You can edit with the IP address of SSF Core.
	healthcheckPeriod	By default, the healthcheckPeriod is to 1.0 second. In every 1.0 second, the system checks the health link between CSIM core and SSF. You can edit this value to required duration.
	logEnabled	By default, the logEnabled field is enabled for log collection. You can edit it to disable log collection.
	logLevel	By default, the logLevel is set to error. You can edit to either error, info or debug.
	logFile	By default, the logFile name is eMindApi.log file.
	ShelfHostIPAddresses	By default, the ShelfHostIPAddresses is set to 172.27.1.2. You can edit with the IP address of the shelf host of the shelf in which the SSF agent is to be considered. Note: This option is only applicable to SSF Agent and no other TL servers.
	localTcpAddress	By default, the localTcpAddress is set to 0.0.0.0:21215. You can edit with the IP address and port on which the SSF agent needs to be configured to listen to incoming request from SSF Core.

A.12.1 Service Manager Configuration

In the Service Manager (SMAN) configuration file `SMAN.conf`, you can add additional user defined services. The `SMAN.conf` file is stored in `/opt/ssf/etc/config/bcsim/`.

Use the following guidelines to add a service:

- Service name should be less than 20 characters.
- Service name should be the same as that of Linux daemon service, if there exists a Linux daemon.
- Description of the service should be equal to or less than 128 characters.
- Binary file path should be equal to or less than 128 characters.
- Tabs should be given before the file list and space should be used between path and filename.

Syntax

```
## SYSLOG-NG configuration #####

#service syslog-ng

{
    enable=y;
    desc="syslog-ng system logger application";
    binFilePath=/opt/ssf/etc/config/bcsim/etc/init.d/syslog-ng;
    numberOfConfigFiles=2;
    filename
    {
        /etc/syslog-ng/ syslog-ng.conf;
        /etc/syslog-ng/ scl.conf;
    }
}
```

```
}
```

NOTICE

To disable a particular service parsing, add '#' before the service.

A.12.2 Service Manager Configuration INI File

Service Manager (SMAN) configuration INI file (*SMAN.ini*) supports two modes:

- **ConfigMode:** Supports commit-config of configured applications. The services under SSF control should be separated by comma (.). For example, `[ConfigMode]:pcieBsnet,dhcpd,tftp,ntpd,syslog-ng,syslcu`
- **NonConfigMode:** Does not support commit-config of configured applications. For example, `[NonConfigMode]:syslog-ng,tftp`

NOTICE

Add an application under "configMode" in case you choose to perform 'Commit Config' for the application. Otherwise add it to "NonConfigMode".

A.13 Hardware Agent Configuration

Hardware Agent RPM consists of MaxCore Agent executable configuration files (*mxccagent.conf*). Configuration files are stored in `/opt/ssf/etc/config/agent`. You can edit the *mxccagent.conf* file stored at this location to configure the required parameters. The following table provides the list of configuration options of *mxccagent.conf* file.

Table A-3 Hardware Agent Configuration

File Name	Configuration Options	Description
mxccagent.conf Directory: /opt/ssf/etc/ config/agent	log_level	Log level INFO - Notifications and important information. DEBUG - Verbose Default is INFO.
	domain	MaxCore ID in the system. This may become obsolete.
	master	Identifies whether SSF core runs on this Shelf Host. The default type is true.
	rack	Rack ID in the system.
	shelf	Shelf ID or chassis number to uniquely identify the MaxCore.
	name	Name of the MaxCore. This is optional.
	mcpu_ipaddr	Shelf Host IP address. The default IP is 127.0.0.1. You can edit it to any Shelf Host IP reachable by the system host.
	agent_listening_port	This is listening TCP/IP port to make a connection for SSF core. The default value is 8888.
	agent_evt_listening_port	This is listening TCP/IP port for event. The default value is 8890.
	con_type	Connection type to BMC. The default value is smi.
	ipmi_con_tmout	IPMI connection timeout. The default value is 5000 msec.
	Note: The below commands is not applicable if the connection type is (con_type) smi. But, it is applicable only if it is LAN type.	
	bmc_ipaddr	BMC IP address. The default IP is 192.168.201.9. You can edit it to any BMC IP reachable by the shelf host.
port	RMCP port. The default value is 623.	
auth_type	RMCP authentication type. The default type is md5.	

Table A-3 Hardware Agent Configuration (continued)

File Name	Configuration Options	Description
	privilege	RMCP privilege. The default privilege is admin.
	username	RMCP username. The default username is admin.
	password	RMCP password. The default password is admin.

A.14 Ramdisk Creation for PCIE-7210

Follow the below steps to create Ramdisk for PCIE-7210.

1. Mount the SSF MaxCore update ISO on /mnt/ssf-update directory

```
mkdir /mnt/ssf-update
mount -o loop mc3ksw_update_7210_centos-<OS Ver>_<SSF
Version>.iso /mnt/ssf-update
```
2. Create a directory and copy the CentOS7.2 Distribution ISO and SSF update ISO images to the directory.

```
cd /home
mkdir ramdisk
cp <CentOS 7.2 Distribution iso> /home/ramdisk
cp mc3ksw_update_7210_centos-<OS Ver>_<SSF Version>.iso
/home/ramdisk
```
3. Execute the create_updated_iso.sh script to generate the ramfs.xz image.

```
/mnt/ssf-update/SSF/create_updated_iso.sh <CentOS7.2
Distribution iso> mc3ksw_update_7210_centos-<OS Ver>_<SSF
Version>.iso xz
```

For example,

```
/mnt/ssf-update/SSF/create_updated_iso.sh
MC3K_7210_OS_CENTOS72_DIST_1_0_2.iso mc3ksw_update_7210_centos-
7.2_1.1.0.39.iso xz
```

NOTICE

If any packages are missing which are required on the machine, please download the package and install it to create the ramfs image.

4. Once the updated ISO is created mount the ISO and copy the `ramfs` image.

```
mkdir /tmp/ramdisk
mount -o loop MC3K_7210_OS_CENTOS72_DIST_1_0_2_updated.iso
/tmp/ramdisk
cp /tmp/ramdisk/images/ramfs.xz <net boot directory>
```


Troubleshooting and FAQ

B.1 Overview

This section provides FAQs and their usual solutions on SSF for MaxCore.

B.2 System Discover or Active Configuration is Empty

Problem Description

After login tree is not populated and prints the following: “System discovery may be in progress or Active configuration is empty.”

Root Cause and Solution

- Shelf id is 0
- `Pciemgmt.service` is not started, in turn `mcagent` is not started or
- Discovery is in progress
- If shelf is not connected to an emulator candidate, then the message means Shelf host discovery just started.
- If shelf is connected to an emulator candidate, empty left with the message above means, the emulator candidate has no configuration.

B.3 SSF Core Failure

Problem Description

Why SSF core fails to start.

Root Cause and Solution

- PAM-postgres-SSF communication error. If so, wait for a while and retry starting SSF core.
- `Radiusd` service is not running.
- User table in SSF got corrupted

- Insufficient persistence memory - Make sure sufficient physical disk space available for database transactions being done by SSF.
- Meta.txt not is sync with the compiled SSF binaries - Make sure that `/opt/ssf/etc/config/main/meta.txt` is proper and not corrupted.

B.4 Host OS Not Displayed

Problem Description

Why is host OS not seen under CPU in navigation pane.

Root Cause and Solution

- The CPU is not in the internal (base) network. May because virtual functions from ETH3 (default base network) are not assigned to the CPU. At least one VF from ETH3 should be assigned to the CPU.
- ssfAgent on that specific CPU is not running.
- ssfAgent did not get shelf host IP address through mcparams.
 - Verify `/opt/boardinfo/params` on that specific CPU to see if shelfhost IP address is populated.
- If firewall is enabled on shelf host and or application host, then proper IP table rules should be added.
- SSF discovery is in progress

B.5 Login Failure

Problem Description

Why do login fail.

Root Cause and Solution

- Login fails if wrong credentials are provided as input.
- SSF failed to start or User table in SSF got corrupted.

B.6 Switch Management Tab is Hidden

Problem Description

Why is switch management tab not seen.

Root Cause and Solution

See the reasons in [Appendix B, Host OS Not Displayed](#).

B.7 Reload Failure

Problem Description

Why reload fails.

Root Cause and Solution

- If shelf ID of the shelf is different to what is present in the configuration.
- If any of IPMI or PEX commands fails.
- Configuration file is corrupted.
- If the PCIE card is PCIE-9205, then if any of RRC configurations set is failed, then reload will fail.
- If configuration is captured on a shelf which has different hardware configuration to the one on which it is applied.
- If configuration is captured at different level (for example captured at Shelf) to the one on which it is applied (at system level).
- Even if the configuration is captured and applied at same level, reload may fail if it applied on different MOID. For example configuration captured at HOST OS of slot 1, CPU1 fails if applied to Slot15, CPU1 HOST OS.

B.8 GUI Access and Logging Issues

Problem Description

User logged in after a long interval and Tree is not getting loaded or not able to access GUI. It returns Error 500 or any http error.

Root Cause and Solution

- Linux may be responding slowly.
- File system is corrupted.

B.9 PCIE-9205 Switch Management is Not Populated in GUI

Problem Description

In case PCIE-9205 is placed as Network CPU and if PCIE-9205 Switch Management is not populated in GUI.

Root Cause and Solution

Perform the following steps.

1. Check if `172.27.<SHELF ID>.2` is reachable from PCIE-9205.
2. Check if PEP4 (ex. `enp6s0`) of PCIE-9205 is having DHCP IP on network `172.27.x.x`. To find device name of PEP4, use the below commands. In this case, `enp6s0` is the PEP4 interface.

```
# lspci -vv | grep "FM10000\|VP"
06:00.0 Ethernet controller: Intel Corporation Ethernet Switch
FM10000 Host Interface
Product Name: FM10000
[VP] Vendor specific: 4
0a:00.0 Ethernet controller: Intel Corporation Ethernet Switch
FM10000 Host Interface
Product Name: FM10000
[VP] Vendor specific: 8
```

```
# systool -c net
Class = "net"
Class Device = "enp0s20u2"
Device = "3-2:1.0"
Class Device = "enp10s0"
Device = "0000:0a:00.0"
Class Device = "enp4s0f0"
Device = "0000:04:00.0"
Class Device = "enp4s0f1"
Device = "0000:04:00.1"
Class Device = "enp6s0"
Device = "0000:06:00.0"
```

3. Check if DHCP client is running on PEP4 interface. `emindTcpAddress` and `ShelfHostIPAddress` in configuration file(s) of SSF BCSIM and SSF RRC TLS is to be set with the IP address of br0 on System Host. It would be of the format `172.27.<SHELFID>.2`.
For example,
`emindTcpAddress=172.27.44.2:21212 => This is to be modified in both rrc/ssfApi.conf and bcsim/ssfApi.conf`
`ShelfHostIPAddress=172.27.44.2 => This is to be modified in bcsim/ssfApi.conf`
Note: Reboot PCIE-9205, if you perform any changes on these files.
4. If an old configuration file is not loaded properly, ensure the below:
 - Size of Port, VLAN, and Pool descriptions is less than 24 characters.
 - In the old configuration file, replace the trailing spaces at the end of each line using the below command in vim:
`%s/\s\+$/ /`

B.10 Incorrect Device Id to RRC PEP Port Mapping

Problem Description

If listed VF ports in the configuration interface is not matching as per mapping listed by `/opt/switch_sw/etc/pcie9205_getpep.sh` script. This script lists the mapping between PEP devices and RRC port, then there is a correction required in the mapping configuration file.

Root Cause and Solution

By default, all the four PEP devices are mapped in reverse to sw1p2* ports of RRC with EEPROM v10. However, if there is any change in the mapping or to confirm the mapping, copy “/opt/switch_sw/etc/pcie9205_getpep.sh” from PCIE-9205 to management CPU and run the script. This script will be listing the mapping between PEP devices and RRC ports.

If there is a different mapping between PEP devices and RRC ports, the correct mapping need to be updated in “/opt/switch_sw/etc/pep_info.conf” as below:

```
<DEVICE ID> <RRC PORT>
```

For example,

1. sw1p20
2. sw1p21
3. sw1p22
4. sw1p23

B.11 How to check whether SSF Services are running fine

Below are the services which need to be checked in order to confirm SSF running properly.

1. pciemgmt.service - on shelf host of SSF configured shelves.
2. mcagent.service – on shelf host of SSF configured shelves.
3. ssfCore.service – on system host.
4. ssfAgent.service – on all SSF configured hosts.
5. dhcpd.service – on shelf host of SSF configured shelves.
6. network.service – on all SSF configured hosts.
7. ssfRRCAgent.service – on all SSF configured network hosts.
8. zend-server.service – on system host.
9. vsftpd.service – on system host.

Related Documentation

C.1 Artesyn Embedded Technologies - Embedded Computing Documentation

The publications listed below are referenced in this manual. You can obtain electronic copies of Artesyn Embedded Technologies - Embedded Computing publications by contacting your local Artesyn sales office. For released products, you can also visit our Web site for the latest copies of our product documentation.

1. Go to www.artesyn.com/computing/support/product/technical-documentation.php.
2. Under FILTER OPTIONS, click the Document types drop-down list box to select the type of document you are looking for.
3. In the **Search** text box, type the product name or manual name and click Filter.

Table C-1 Artesyn Embedded Technologies - Embedded Computing Publications

Document Title	Publication Number
SSF for MaxCore™ MC3000 Platform XML Interface Guide	6806800T71
SSF for MaxCore™ MC3000 Platform Command Line Interface Guide	6806800T87
MaxCore™ MC3000 Platform Installation and Use	6806800T88
MaxCore™ MC3000 Platform Quick Start Guide	6806800T89
MaxCore™ MC3000 Platform Safety Notes Summary	6806800T90
ViewCheck on PCIE Card User Guide	6806800T92
MaxCore™ MC3000 Platform Networking Application Note	6806800T97
Getting Started with MaxCore™ MC3000 Application Note	6806800T98
SharpStreamer™Pro PCIE-7210 Installation and Use	6806800U29



Artesyn Embedded Technologies, Artesyn and the Artesyn Embedded Technologies logo are trademarks and service marks of Artesyn Embedded Technologies, Inc. All other product or service names are the property of their respective owners.

© 2017 Artesyn Embedded Technologies, Inc.