



Charon-SSP 5.4 for Linux User's Guide



Contents

| | | |
|----------|--|-----------|
| 1 | <i>About This Guide</i> | 14 |
| 1.1 | Intended Audience | 14 |
| 1.2 | Document Structure | 14 |
| 1.3 | Products Covered in this Guide | 15 |
| 1.4 | Obtaining Documentation | 17 |
| 1.5 | Obtaining Technical Assistance or Product Information | 17 |
| 1.5.1 | Obtaining Technical Assistance..... | 17 |
| 1.5.2 | Obtaining General Product Information | 17 |
| 1.6 | Conventions | 18 |
| 2 | <i>Charon-SSP Licensing Options Overview</i> | 19 |
| 2.1 | Sentinel/Gemalto HASP Licenses | 19 |
| 2.2 | Charon-SSP Automatic Licensing for Cloud Environments | 19 |
| 2.3 | Virtual Environment (VE) Licenses | 19 |
| 3 | <i>Charon-SSP Product Overview</i> | 21 |
| 3.1 | General Information | 21 |
| 3.2 | Supported Virtual Hardware | 23 |
| 3.2.1 | Supported Virtual Hardware in non-Cloud Installations..... | 23 |
| 3.2.2 | Supported Virtual Hardware in Cloud Installations | 24 |
| 3.3 | Charon-SSP Product Variant Comparison | 25 |
| 3.3.1 | Product Variant Overview | 25 |
| 3.3.2 | Product Variant Comparison | 26 |
| 4 | <i>Host System Requirements</i> | 28 |
| 4.1 | Minimum Host System Hardware Requirements | 28 |
| 4.2 | Host System Software Requirements | 30 |
| 4.2.1 | Linux Operating System Prerequisites..... | 30 |
| 4.2.2 | Additional Software Requirements | 31 |
| 4.3 | Guest Operating System Requirements | 31 |
| 4.4 | NetworkManager Considerations | 31 |
| 4.4.1 | NetworkManager and Charon Manager for Linux versions 7.x..... | 32 |
| 4.4.2 | NetworkManager and Charon Manager for Linux versions 8.x..... | 33 |
| 4.5 | Firewall Requirements | 34 |
| 4.5.1 | Frequently used TCP and UDP Ports | 34 |
| 4.5.2 | Linux Firewall and Virtual Bridges | 35 |
| 5 | <i>Charon-SSP Software Installation</i> | 36 |
| 5.1 | General License Information | 36 |
| 5.1.1 | Initial License Installation Overview | 36 |
| 5.1.1.1 | Sentinel/Gemalto Licenses..... | 36 |
| 5.1.1.2 | Charon-SSP VE (Virtual Environment) Licenses | 37 |

| | | |
|------------|---|-----------|
| 5.2 | Charon-SSP RPM Installation | 38 |
| 5.2.1 | Installation Packages Overview | 38 |
| 5.2.1.1 | Components of a Charon-SSP Installation | 38 |
| 5.2.1.2 | Charon-SSP Installation Packages | 39 |
| 5.2.2 | Additional Installation Considerations and Prerequisites | 41 |
| 5.2.2.1 | General Information..... | 41 |
| 5.2.2.2 | License Considerations | 42 |
| 5.2.2.3 | Considerations for Charon-SSP Hosts without Graphics HW..... | 42 |
| 5.2.2.4 | Special Prerequisites for the Charon-SSP Director | 43 |
| 5.2.2.5 | Special Prerequisites for the Charon-SSP Agent..... | 43 |
| 5.2.2.6 | Special Prerequisites for Additional Utilities | 44 |
| 5.2.3 | Installation Commands on Supported Host Operating Systems..... | 45 |
| 5.2.4 | Installing the Charon-SSP Packages | 46 |
| 5.2.4.1 | Installation Example | 46 |
| 5.2.5 | Post-Installation Steps..... | 51 |
| 5.2.5.1 | Post-Installation Tasks for Version 8.x of Supported Linux Distributions..... | 51 |
| 5.2.5.2 | Sentinel HASP Post-Installation Tasks | 52 |
| 5.2.5.2.1 | Setting the Password for the Sentinel HASP GUI | 52 |
| 5.2.5.2.2 | Set more Restrictive File Protections on the hasplm.ini File | 52 |
| 5.2.5.2.3 | Enable access and error logging..... | 52 |
| 5.2.5.3 | Charon-SSP Post-Installation Tasks..... | 53 |
| 5.2.5.3.1 | Setting the PATH variable | 53 |
| 5.2.5.3.2 | Installing the bridge-utils and autossh Packages | 53 |
| 5.2.5.4 | Charon-SSP Manager Post-Installation Tasks on Linux..... | 54 |
| 5.2.5.4.1 | Installing the Xephyr X-Server..... | 54 |
| 5.2.5.4.2 | Creating a Desktop Menu Item for Charon-SSP Manager..... | 55 |
| 5.2.5.5 | Charon-SSP Director Post-Installation Tasks on Linux..... | 57 |
| 5.2.5.5.1 | Creating a Desktop Menu Item for Charon-SSP Director | 57 |
| 5.3 | Charon-SSP Baremetal Installation..... | 58 |
| 5.3.1 | General Information about the Baremetal Distribution | 58 |
| 5.3.2 | Creating Bootable USB media from Baremetal ISO files..... | 59 |
| 5.3.3 | Basic Installation Steps for the Baremetal Distribution..... | 60 |
| 5.3.4 | Charon-SSP Baremetal Post-Installation Tasks..... | 63 |
| 5.3.4.1 | Baremetal Post-Installation Tasks Overview..... | 63 |
| 5.3.4.2 | Customizing the User Environment..... | 63 |
| 5.3.4.3 | Enrolling the Stromasys Public Key for UEFI Secure Boot..... | 63 |
| 6 | Charon Host System Management Overview..... | 65 |
| 6.1 | Host Management for Conventional and Cloud Hosts | 65 |
| 6.1.1 | General System Management Information..... | 65 |
| 6.1.2 | User Accounts in Charon-SSP Cloud Marketplace Images..... | 65 |
| 6.2 | Host Management Charon-SSP Baremetal | 66 |
| 6.2.1 | General System Management Information..... | 66 |
| 6.2.2 | Charon-SSP Baremetal User Accounts..... | 66 |
| 6.2.3 | Charon-SSP Baremetal User Interface..... | 67 |
| 6.2.3.1 | Shutdown, Reboot, Screen Lock..... | 67 |
| 6.2.3.2 | Home Screen Functions | 68 |
| 6.2.3.3 | Charon Screen Functions..... | 69 |
| 6.2.3.4 | Toolbox Screen Functions..... | 69 |
| 6.2.3.4.1 | Host System Settings | 70 |
| 6.2.3.4.2 | Host System Information Display | 71 |
| 6.2.3.4.3 | Update Menu Item | 71 |
| 6.2.3.4.4 | Key Manager | 72 |

| | | |
|-------------|--|-----------|
| 6.2.3.4.5 | Terminal..... | 72 |
| 6.2.3.4.6 | Additional Toolbox Applications | 73 |
| 7 | Configuring and Using the Charon-SSP Software..... | 74 |
| 7.1 | Overview..... | 74 |
| 7.2 | Charon-SSP Directory Structure | 74 |
| 7.3 | Interaction of the Charon-SSP Components..... | 75 |
| 7.4 | Using the Charon-SSP Director | 76 |
| 7.4.1 | Starting the Charon-SSP Director..... | 76 |
| 7.4.2 | Working with the Charon-SSP Director | 76 |
| 7.4.2.1 | Charon-SSP Director Main Menu Bar | 77 |
| 7.4.2.2 | Managing Charon-SSP Director Subgroups | 77 |
| 7.4.2.3 | Charon-SSP Director System Context Menu | 78 |
| 7.4.2.4 | Charon-SSP Director Additional Context Menu | 79 |
| 7.4.2.5 | Charon-SSP Director Keyboard Shortcuts | 79 |
| 7.5 | Using the Charon-SSP Manager | 80 |
| 7.5.1 | Starting the Charon-SSP Manager..... | 81 |
| 7.5.1.1 | Connecting to the Charon-SSP Agent of the Target Host System | 81 |
| 7.5.1.2 | Troubleshooting information..... | 83 |
| 7.5.1.3 | Running the Charon-SSP Manager via SSH X11-Forwarding..... | 84 |
| 7.5.1.4 | Charon-SSP Manager Overview | 85 |
| 7.5.2 | Creating a Virtual Machine | 88 |
| 7.5.3 | Configuring a Virtual Machine | 89 |
| 7.5.3.1 | Hardware Family Configuration (Model) | 91 |
| 7.5.3.2 | CPU Configuration | 93 |
| 7.5.3.3 | DIT Configuration | 95 |
| 7.5.3.3.1 | Client JIT..... | 96 |
| 7.5.3.3.2 | Server JIT | 97 |
| 7.5.3.4 | Memory Configuration | 98 |
| 7.5.3.5 | Graphics Configuration..... | 99 |
| 7.5.3.6 | SCSI Storage Configuration..... | 103 |
| 7.5.3.6.1 | Drivers for Emulated SCSI Devices | 103 |
| 7.5.3.6.2 | SCSI Configuration Window Overview..... | 104 |
| 7.5.3.6.3 | Creating a New Virtual Disk Container File..... | 105 |
| 7.5.3.6.4 | Creating a New Virtual Tape Container File..... | 106 |
| 7.5.3.6.5 | Adding or Editing a Virtual SCSI Device | 107 |
| 7.5.3.6.5.1 | Disks with non-zero LUN ID | 110 |
| 7.5.3.6.6 | Physical Disk Parameters on Charon-SSP | 111 |
| 7.5.3.6.7 | Removing a SCSI Storage Device | 112 |
| 7.5.3.7 | Configuring VDS Storage Devices | 112 |
| 7.5.3.7.1 | Adding or Editing an VDS Device..... | 113 |
| 7.5.3.7.2 | Removing a VDS Storage Device | 115 |
| 7.5.3.8 | Configuring Chassis Identification Parameters | 116 |
| 7.5.3.9 | Configuring a Floppy Drive..... | 117 |
| 7.5.3.10 | Vconsole Configuration (Charon-SSP/4V only) | 118 |
| 7.5.3.10.1 | Vconsole Network Configuration | 119 |
| 7.5.3.10.2 | Vconsole Physical Line Configuration | 120 |
| 7.5.3.11 | TTYA Configuration..... | 120 |
| 7.5.3.11.1 | TTYA Physical Line Configuration..... | 121 |
| 7.5.3.11.2 | TTYA Network Configuration..... | 121 |
| 7.5.3.12 | TTYB Configuration | 122 |
| 7.5.3.13 | TTYX Configuration..... | 123 |

| | | |
|--------------|--|-----|
| 7.5.3.13.1 | Prerequisites..... | 123 |
| 7.5.3.13.2 | TTYX On-Board Mode on Charon-SSP/4U/4V | 124 |
| 7.5.3.13.2.1 | Adding Serial Ports in TTYX On-Board Mode | 125 |
| 7.5.3.13.2.2 | Modifying or Removing TTYX On-Board Mode Ports | 126 |
| 7.5.3.13.2.3 | Managing TTYX On-Board Mode Ports on Solaris | 126 |
| 7.5.3.13.3 | DIGI AccelePort Mode on Charon-SSP/4U (+) | 127 |
| 7.5.3.13.3.1 | Adding Serial Ports in DIGI AccelePort Mode | 127 |
| 7.5.3.13.3.2 | Modifying or Removing Ports in DIGI AccelePort Mode | 127 |
| 7.5.3.13.3.3 | Solaris Driver Installation for DIGI AccelePort Emulation | 128 |
| 7.5.3.13.3.4 | Managing DIGI AccelePort Ports on Solaris | 129 |
| 7.5.3.13.4 | Adding a DIGI PCI Pass-Through Device on Charon-SSP/4U (+)..... | 129 |
| 7.5.3.13.5 | TTYX Ports on Charon-SSP/4M..... | 130 |
| 7.5.3.13.5.1 | Adding Serial Ports in TTYX Mode on SUN-4M..... | 130 |
| 7.5.3.13.5.2 | Modifying or Removing Ports in TTYX Mode on SUN-4M | 130 |
| 7.5.3.13.5.3 | Managing SUN-4M TTYX Ports on Solaris | 130 |
| 7.5.3.14 | GPIO Configuration on Charon-SSP/4U(+)..... | 131 |
| 7.5.3.15 | Parallel Interface Configuration | 132 |
| 7.5.3.16 | Audio Configuration | 133 |
| 7.5.3.17 | USB Configuration..... | 135 |
| 7.5.3.18 | Ethernet Configuration | 137 |
| 7.5.3.18.1 | Supported Adapter Models..... | 137 |
| 7.5.3.18.2 | Charon-SSP Ethernet Configuration Screen Functions | 138 |
| 7.5.3.19 | NVRAM Configuration | 140 |
| 7.5.3.20 | License Settings | 141 |
| 7.5.3.21 | Log Configuration | 143 |
| 7.5.3.21.1 | Viewing the Charon-SSP Log Files..... | 144 |
| 7.5.4 | Starting, Stopping, and Suspending the Emulated System | 145 |
| 7.5.4.1 | Starting the Emulated System..... | 145 |
| 7.5.4.1.1 | Interactive Start | 145 |
| 7.5.4.1.2 | Start with Host System Startup | 147 |
| 7.5.4.2 | Stopping the Emulated System | 147 |
| 7.5.4.3 | Suspending the Emulated System | 148 |
| 7.5.5 | Virtual Machine Context Menu | 149 |
| 7.5.5.1 | Run the Virtual Machine | 149 |
| 7.5.5.2 | Virtual Machine Settings..... | 149 |
| 7.5.5.3 | Remove Machine from the List (non-AL only)..... | 149 |
| 7.5.5.4 | Delete VM from Disk | 150 |
| 7.5.5.5 | Rename VM..... | 150 |
| 7.5.5.6 | Backup VM (Charon-SSP AL Images only) | 150 |
| 7.5.6 | Host System Network Configuration | 151 |
| 7.5.6.1 | General Information..... | 151 |
| 7.5.6.2 | Network Settings Tool – Additional Notes and Caveats | 151 |
| 7.5.6.3 | Network Settings Overview | 153 |
| 7.5.6.4 | Managing Host System Network Interfaces | 154 |
| 7.5.6.5 | Creating a Virtual Network (Virtual Bridge) | 156 |
| 7.5.6.6 | Deleting a Virtual Network | 158 |
| 7.5.6.7 | Resizing a Virtual Network | 158 |
| 7.5.6.8 | Adding a VLAN interface | 159 |
| 7.5.6.9 | Deleting a VLAN Interface | 160 |
| 7.5.7 | Miscellaneous Management Tasks | 161 |
| 7.5.7.1 | Displaying Host Information | 161 |
| 7.5.7.2 | Adding an Existing Configuration to the Charon-SSP Manager | 161 |
| 7.5.7.3 | Determining the Charon-SSP Manager Version | 162 |
| 7.5.7.4 | Modifying the Charon-SSP Agent Preferences..... | 162 |

| | | |
|-------------|---|------------|
| 7.5.7.5 | Setting Console Options..... | 163 |
| 7.5.7.6 | Retrieving a Charon-SSP Core Dump | 164 |
| 7.5.8 | Special Baremetal and Charon-SSP AL Tools | 165 |
| 7.5.8.1 | File Manager..... | 165 |
| 7.5.8.2 | Storage Manager..... | 166 |
| 7.5.8.3 | Setting Time and Date..... | 168 |
| 7.5.8.4 | SFTP Server..... | 168 |
| 7.5.8.5 | Baremetal SSH Public Key Import | 169 |
| 7.6 | The Charon Management Password..... | 170 |
| 7.6.1 | General Information about the Management Password..... | 170 |
| 7.6.2 | Where and How to Set the Charon Management Password..... | 170 |
| 7.6.3 | Resetting a Forgotten Management Password | 171 |
| 7.6.3.1 | Password Reset using the Command-Line..... | 171 |
| 7.6.3.2 | Password Reset on a Baremetal System..... | 171 |
| 7.7 | Using Charon-SSP from the Command-Line..... | 172 |
| 7.7.1 | Emulator Command-Line Options | 172 |
| 7.7.1.1 | Program Name | 172 |
| 7.7.1.2 | Syntax..... | 172 |
| 7.7.1.3 | Description..... | 172 |
| 7.7.1.4 | Exit Status | 174 |
| 7.7.1.5 | Examples..... | 174 |
| 7.7.2 | Charon-SSP CLI..... | 175 |
| 7.7.2.1 | Accessing and Exiting the Charon-SSP CLI | 175 |
| 7.7.2.2 | Charon-SSP CLI Available Commands | 176 |
| 7.7.2.2.1 | Commands Relevant for End-Users..... | 176 |
| 7.7.2.2.1.1 | The <i>help</i> Command | 176 |
| 7.7.2.2.1.2 | The <i>log</i> Command | 176 |
| 7.7.2.2.1.3 | The <i>hasp</i> Command..... | 177 |
| 7.7.2.2.1.4 | The <i>ve</i> Command | 178 |
| 7.7.2.2.2 | Debugging Commands not Relevant for End-Users | 179 |
| 7.8 | Using the Charon-SSP Agent..... | 180 |
| 7.8.1 | Starting the Charon-SSP Agent Service..... | 180 |
| 7.8.2 | Stopping the Charon-SSP Agent Service..... | 180 |
| 7.8.3 | TCP/IP Ports Used by the Charon-SSP Agent..... | 180 |
| 7.9 | User Access to the Virtual SPARC System..... | 181 |
| 7.9.1 | Console Access..... | 181 |
| 7.9.1.1 | Physical Serial Console Access | 181 |
| 7.9.1.2 | TCP/IP-based Serial Console Access via Charon-SSP Manager | 182 |
| 7.9.1.3 | TCP/IP-based Serial Console Access without Charon-SSP Manager..... | 183 |
| 7.9.1.4 | External Serial Console Considerations..... | 184 |
| 7.9.1.4.1 | Serial Console Settings in Charon Manager and Configuration File..... | 184 |
| 7.9.1.4.2 | Accessing the External Emulator Console from a Remote System | 184 |
| 7.9.1.4.3 | Changing the Default PuTTY Configuration for the Serial Console | 185 |
| 7.9.1.4.3.1 | Changing the Default PuTTY Configuration on Windows | 185 |
| 7.9.1.4.3.2 | Changing the Default PuTTY Configuration on a Charon Linux Host..... | 186 |
| 7.9.1.4.4 | Common Problems..... | 186 |
| 7.9.1.5 | Console Access via the Emulated Graphics Device | 187 |
| 7.9.2 | Graphical Interface via X11 Server on Linux and Baremetal..... | 188 |
| 7.9.2.1 | General Information..... | 188 |
| 7.9.2.2 | Enabling XDMCP..... | 189 |
| 7.9.2.2.1 | Enabling XDMCP on Solaris 2.5 to Solaris 9 | 189 |
| 7.9.2.2.2 | Enabling XDMCP on Solaris 10 | 189 |

| | | |
|-------------|--|------------|
| 7.9.2.2.3 | Enabling XDMCP on Solaris 11 | 190 |
| 7.9.2.2.4 | Enabling XDMCP on Older Solaris Versions..... | 190 |
| 7.9.2.3 | Additional Notes for Solaris 11 | 191 |
| 7.9.2.3.1 | Enabling the VNC server on Solaris 11.4..... | 191 |
| 7.9.2.3.2 | Enabling Connections to Remote Xservers..... | 192 |
| 7.9.2.4 | Configuring and Starting the X11 Server in Charon-SSP Manager | 193 |
| 7.9.2.5 | X11 Server Configuration Parameters | 194 |
| 7.9.2.5.1 | Use Cases for the X-Server Additional Options | 194 |
| 7.9.2.6 | Stopping the X11 Server | 196 |
| 7.9.3 | Using the X-Server on Windows..... | 197 |
| 8 | <i>Additional Charon-SSP Tools.....</i> | 199 |
| 8.1 | iSCSI Initiator | 199 |
| 8.1.1 | Prerequisites | 199 |
| 8.1.2 | Adding an iSCSI Target | 200 |
| 8.1.3 | Removing an iSCSI Target..... | 202 |
| 8.2 | NFS Server | 203 |
| 8.2.1 | Prerequisites | 203 |
| 8.2.2 | Adding an NFS Share..... | 203 |
| 8.2.3 | Removing an NFS Share | 204 |
| 8.3 | VNC Server | 205 |
| 8.3.1 | Enabling and Disabling the VNC Server..... | 205 |
| 8.3.2 | Connecting to the Charon-SSP Host via VNC..... | 205 |
| 8.4 | Using a Jumpstart Server..... | 206 |
| 9 | <i>Data Transfer to/from Charon-SSP Host and Guest.....</i> | 207 |
| 9.1 | General Information | 207 |
| 9.1.1 | Direct Data Transfer Over the Network | 207 |
| 9.1.2 | Transferring Backup Archives | 207 |
| 9.1.3 | Cloud-Specific Data Transfer Options | 207 |
| 9.2 | Using VTAPE Container Files | 208 |
| 9.2.1 | Convert Backup Archives to VTAPE Container Files | 208 |
| 9.2.2 | Using UFSdump Backup Archives as VTAPE Container Files | 208 |
| 9.3 | Using NFS for Data Transfer | 208 |
| 9.3.1 | Charon-SSP Host Configured as NFS Server | 209 |
| 9.4 | Using SCP for Data Transfer | 212 |
| 9.5 | Using SFTP for Data Transfer | 213 |
| 10 | <i>SSH VPN – Connecting Charon Host and Guest to Customer Network</i> | 214 |
| 10.1 | Prerequisites..... | 214 |
| 10.1.1 | Creating and Uploading the Public SSH Key..... | 215 |
| 10.1.2 | Adapt SSH Configuration on Charon-SSP Host System | 216 |
| 10.1.3 | PuTTY Considerations | 216 |
| 10.2 | Setting up the VPN Tunnel | 217 |
| 10.2.1 | Steps on the Charon-SSP Host System..... | 218 |
| 10.2.1.1 | Creating a VPN Bridge | 218 |
| 10.2.1.2 | Assigning the Guest Ethernet Interface | 219 |
| 10.2.2 | Steps on the Remote Linux System..... | 220 |
| 10.2.3 | Steps on the Solaris Guest System | 221 |

| | | |
|-------------|---|------------|
| 10.2.3.1 | Routing to/from Solaris Guest | 221 |
| 10.3 | Stopping the SSH Tunnel | 222 |
| 11 | Configuring Charon-SSP Baremetal in Kiosk Mode | 223 |
| 12 | Charon-SSP License Management..... | 224 |
| 12.1 | Licensing Charon-SSP—General Aspects | 224 |
| 12.2 | Sentinel HASP License Management..... | 225 |
| 12.2.1 | Managing Sentinel Licenses with Charon-SSP Manager | 225 |
| 12.2.1.1 | Viewing the License Details | 225 |
| 12.2.1.2 | Gathering Customer-to-Vendor (C2V) Details | 226 |
| 12.2.1.3 | Applying Vendor-to-Customer (V2C) License Updates | 227 |
| 12.2.1.4 | License Manager | 228 |
| 12.2.2 | Managing Sentinel Licenses from the Command-Line | 229 |
| 12.2.2.1 | Viewing the License Details | 229 |
| 12.2.2.2 | Gathering Customer-to-Vendor (C2V) Details | 230 |
| 12.2.2.3 | Applying Vendor-to-Customer (V2C) License Updates | 230 |
| 12.2.3 | Managing Licenses with Sentinel Admin Control Center | 231 |
| 12.2.3.1 | Viewing Licenses..... | 231 |
| 12.2.3.2 | Gathering Customer-to-Vendor (C2V) Details | 231 |
| 12.2.3.3 | Applying Vendor-to-Customer (V2C) License Updates | 232 |
| 12.2.3.4 | Allowing Access to and from Network License Servers | 233 |
| 12.2.3.4.1 | Controlling Access to the License Server on the Client Side | 234 |
| 12.2.3.4.2 | Controlling Access to Network Licenses on the Server Side | 235 |
| 12.2.3.5 | Removing a Software License..... | 236 |
| 12.3 | Virtual Environment (VE) License Management | 237 |
| 12.3.1 | Creating a VE C2V file | 237 |
| 12.3.1.1 | Running esxi_bind before First C2V Creation on VMware | 237 |
| 12.3.1.2 | Creating the VE C2V Fingerprint File | 238 |
| 12.3.2 | Installing a VE V2C File | 239 |
| 12.4 | License Troubleshooting..... | 239 |
| 12.4.1 | Log Files for License Troubleshooting..... | 239 |
| 12.4.1.1 | Log File location | 239 |
| 12.4.1.2 | Log Example: Loss of License during Operation | 240 |
| 12.4.2 | System Processes for Licensing..... | 241 |
| 12.4.2.1 | Sentinel HASP Processes | 241 |
| 12.4.2.2 | VE License Server Process..... | 241 |
| 12.4.3 | Other Problems | 241 |
| 13 | Charon-SSP Software Upgrade | 242 |
| 13.1 | Upgrading via RPM Installation | 242 |
| 13.1.1 | Host Operating System Specifics for Upgrade | 242 |
| 13.1.1.1 | Charon-SSP Installation Packages | 242 |
| 13.1.1.2 | Upgrade Commands on Supported Host Systems | 244 |
| 13.1.2 | Upgrading the Charon-SSP Software Packages | 245 |
| 13.2 | Upgrading the Charon-SSP Baremetal Distribution..... | 247 |
| 13.2.1 | Upgrading Charon-SSP Packages Using the Update App..... | 248 |
| 13.2.2 | Upgrading Charon-SSP Baremetal Using an ISO file or Installation Medium | 249 |
| 13.3 | Upgrading the Charon-SSP Barebone Distribution | 250 |
| 13.4 | Cloud Image Upgrade | 250 |

- 13.4.1 RPM-based Upgrade..... 250
- 13.4.2 Upgrading by Creating a New Cloud Instance 250
- 13.5 Charon Manager Automatic Update..... 251**
- 14 Charon-SSP Software Deinstallation 252**
- 14.1 Software Deinstallation on Conventional RPM Installation 252**
- 14.1.1 Removing the Sentinel HASP Software 252
- 14.1.2 Removing the Charon-SSP Packages on Linux..... 252
- 14.2 Software Deinstallation on Baremetal System 253**
- 14.3 Software Deinstallation on Cloud-Specific Images..... 253**
- 14.3.1 Deinstalling the Charon Manager on Management System 253
- 14.3.2 Terminating the Charon-SSP Cloud Instance 253
- E.1.2.1.1 File-based Configuration Example 310
- E.1.2.1.2 NetworkManager-based Configuration Examples 312

Appendix

- A Appendix – Charon-SSP GUI for Microsoft Windows..... 254**
- A.1 Charon-SSP GUI Installation on Windows 254**
- A.1.1 Prerequisites and General Information 254
- A.1.2 Installing the Charon-SSP Manager for Microsoft Windows..... 255
- A.1.3 Installing the Charon-SSP Director for Microsoft Windows 256
- A.1.4 Using the Charon-SSP GUI on Microsoft Windows 257
- A.1.4.1 Charon Manager on Windows – Serial Console Access 258
- A.1.4.2 Charon Manager on Windows – Graphical Console Access..... 259
- A.1.5 Upgrading the Charon-SSP GUI on Microsoft Windows..... 259
- A.1.6 Removing the Charon-SSP GUI on Microsoft Windows 260
- B Appendix – Configuration File Reference..... 261**
- B.1 Syntax 261**
- B.1.1 Section..... 261
- B.1.2 Properties 261
- B.1.3 Comments 261
- B.1.4 Blank Lines..... 262
- B.2 Reference 262**
- B.2.1 [cpu] Section 262
- B.2.1.1 dit 262
- B.2.1.2 dit_page_size 262
- B.2.1.3 code_page_size 263
- B.2.1.4 fp_boost 263
- B.2.1.5 int_boost..... 263
- B.2.1.6 number 263
- B.2.1.7 idle..... 264
- B.2.1.8 ht 264
- B.2.2 [ethernet] Section 265
- B.2.2.1 interface 265

| | | |
|----------|---|-----|
| B.2.2.2 | mac | 266 |
| B.2.2.3 | model | 266 |
| B.2.3 | [ethernet_n] Section | 266 |
| B.2.4 | [log] Section | 267 |
| B.2.4.1 | destination | 267 |
| B.2.4.2 | path | 268 |
| B.2.4.3 | severity | 268 |
| B.2.4.4 | rotation | 268 |
| B.2.5 | [nvram] Section | 269 |
| B.2.5.1 | disable_autoboot | 269 |
| B.2.5.2 | hostid | 269 |
| B.2.5.3 | path | 269 |
| B.2.6 | [ram] Section | 269 |
| B.2.6.1 | allocator | 270 |
| B.2.6.2 | size | 270 |
| B.2.7 | [scsi_n] Section | 271 |
| B.2.7.1 | lun_X | 272 |
| B.2.7.2 | type | 273 |
| B.2.7.3 | pass_through | 273 |
| B.2.7.4 | removable | 273 |
| B.2.8 | [scsix_n] Section | 274 |
| B.2.9 | [vds_*] sections | 275 |
| B.2.9.1 | path | 275 |
| B.2.9.2 | pass_through | 275 |
| B.2.9.3 | write_through | 276 |
| B.2.10 | [chassis] Section | 276 |
| B.2.11 | [floppy] Section | 276 |
| B.2.11.1 | type | 276 |
| B.2.11.2 | path | 276 |
| B.2.12 | [system] Section | 277 |
| B.2.12.1 | cpu_affinity | 277 |
| B.2.12.2 | io_affinity | 277 |
| B.2.12.3 | io_cpus | 277 |
| B.2.12.4 | machine | 278 |
| B.2.12.5 | model | 278 |
| B.2.13 | [vconsole] Section (Charon-SSP/4V only) | 278 |
| B.2.13.1 | port | 279 |
| B.2.13.2 | start_console | 279 |
| B.2.13.3 | restrict_access | 279 |
| B.2.13.4 | type | 280 |
| B.2.13.5 | log_path | 280 |
| B.2.14 | [ttya] Section | 280 |
| B.2.14.1 | port | 281 |
| B.2.14.2 | restrict_access | 281 |
| B.2.14.3 | start_console | 281 |
| B.2.14.4 | type | 282 |
| B.2.14.5 | log_path | 282 |
| B.2.14.6 | Examples | 282 |
| B.2.15 | [ttyb] Section | 283 |
| B.2.16 | [ttyx] Section | 283 |
| B.2.17 | [digi_ppt_n] Section | 283 |
| B.2.17.1 | path | 283 |
| B.2.18 | [gpib_n] Section | 283 |
| B.2.18.1 | path | 283 |

| | | |
|------------|---|------------|
| B.2.19 | [parallel] Section..... | 284 |
| B.2.19.1 | Printer..... | 284 |
| B.2.20 | [license] Section..... | 284 |
| B.2.20.1 | regular_key_id..... | 284 |
| B.2.20.2 | backup_key_id..... | 285 |
| B.2.20.3 | server..... | 285 |
| B.2.20.4 | backup_server..... | 285 |
| B.2.20.5 | server_key and backup_server_key..... | 285 |
| B.2.21 | [graphics] Section..... | 286 |
| B.2.21.1 | type..... | 286 |
| B.2.21.2 | dual_display..... | 286 |
| B.2.21.3 | remote_display..... | 287 |
| B.2.21.4 | display1 and display2..... | 287 |
| B.2.21.5 | remote_port1 and remote_port2..... | 287 |
| B.2.21.6 | console..... | 287 |
| B.2.21.7 | mouse_port..... | 288 |
| B.2.21.8 | keyboard_port..... | 288 |
| B.2.21.9 | keyboard_layout..... | 288 |
| B.2.21.10 | resolution..... | 289 |
| B.2.21.11 | full_screen..... | 289 |
| B.2.21.12 | refresh_rate..... | 289 |
| B.2.22 | [audio] Section..... | 289 |
| B.2.22.1 | enable..... | 290 |
| B.2.22.2 | server..... | 290 |
| B.2.23 | [usb] Section..... | 290 |
| B.2.23.1 | enable..... | 290 |
| B.2.24 | [obp] Section..... | 290 |
| B.2.24.1 | path..... | 290 |
| C | Appendix – OpenBoot Console..... | 291 |
| C.1 | OpenBoot Console Overview..... | 291 |
| C.2 | OpenBoot Console Command Reference..... | 291 |
| C.2.1 | banner..... | 291 |
| C.2.2 | boot..... | 292 |
| C.2.3 | devalias..... | 292 |
| C.2.4 | help..... | 293 |
| C.2.5 | history..... | 293 |
| C.2.6 | nvalias..... | 294 |
| C.2.7 | nvunalias..... | 294 |
| C.2.8 | printenv..... | 294 |
| C.2.9 | probe-scsi..... | 295 |
| C.2.10 | quit..... | 296 |
| C.2.11 | reset..... | 296 |
| C.2.12 | setenv..... | 297 |
| C.2.13 | show-devs..... | 297 |
| C.2.14 | show-disks..... | 298 |
| D | Appendix – Command-Line Utilities Reference..... | 299 |
| D.1 | Prerequisites..... | 299 |
| D.2 | Disabling the Charon-SSP Agent Service..... | 299 |
| D.3 | Configure the Shell Path..... | 299 |

| | | |
|------------|---|------------|
| D.4 | Tools Reference | 300 |
| D.4.1 | Sentinel HASP License Utilities | 300 |
| D.4.1.1 | hasp_srm_view | 300 |
| D.4.1.2 | hasp_update | 302 |
| D.4.2 | Storage Utilities | 303 |
| D.4.2.1 | mkdskcmd | 303 |
| D.4.2.2 | mtd | 304 |
| D.4.3 | Other Utilities | 305 |
| D.4.3.1 | The create-config Utility | 305 |
| D.4.3.2 | The verify-config Utility | 306 |
| D.4.3.3 | The import-config Utility | 307 |
| E | Appendix – Additional Information for Cloud | 308 |
| E.1 | Dedicated NIC for Guest System | 308 |
| E.1.1 | Basic Concept..... | 308 |
| E.1.2 | Configuration Examples | 310 |
| E.1.2.1 | Step 1: Prepare Second NIC on Host for Use by the Solaris Guest..... | 310 |
| E.1.2.2 | Step 2: Add the Dedicated NIC to the Emulator Configuration..... | 313 |
| E.1.2.3 | Step 3: Configure the Solaris Guest to Use the Private Cloud IP Address | 313 |
| F | Index | 314 |

Document Revisions

| Date | Document version | Comments |
|--------------|------------------|--|
| 03 June 2022 | Version 1 | Initial version of Charon-SSP 5.4 user's guide |
| | | |
| | | |
| | | |
| | | |
| | | |

1 About This Guide

Please note:

- For setting up a **cloud-based Charon host instance**, please refer to the Charon-SSP Version 5 [cloud-specific Getting Started guides](#).
- The sample outputs in this document may show different versions than the one documented in this manual, but they are still representative of what a user will see.

This preface describes conventions and content of this user's guide. It describes the intended audience, how to obtain copies of this guide, related documentation, and further support.

1.1 Intended Audience

This user's guide is intended for anyone who needs to install, configure, or manage the Stromasys Charon-SSP processor/platform virtualization software. A general working knowledge of Linux and its conventions is expected.

1.2 Document Structure

This document contains the following main sections:

- [Charon-SSP Product Overview](#): provides an overview of the emulator concepts and the supported emulated hardware.
- [Charon-SSP Licensing Options Overview](#): provides an overview of the different licensing options for on-premises and cloud-based installations.
- [Host System Requirements](#): the minimum hardware and software requirements that a Charon-SSP host system must fulfill, the guest operating system requirements, and additional considerations regarding network and firewall setup.
- [Charon-SSP Software Installation](#): installation of the individual software packages making up the Charon-SSP product; information about some package-specific additional prerequisites. Installation of the Charon-SSP Barebone distribution.
- [Charon Host System Management Overview](#): notes about the management of the different installation variants.
- [Configuring and Using the Charon-SSP Software](#): introduction to the operation of Charon-SSP Director and Charon-SSP Manager; creating a virtual SPARC system and configuring the virtual hardware; running and accessing a Charon-SSP instance. The section also describes how to use Charon-SSP from the command-line and how to use the commands provided by the Charon-SSP CLI.
- [Additional Charon-SSP Tools](#): configuring supporting tools such as iSCSI initiator, NFS server, and VNC server.
- [Data Transfer to/from Charon-SSP Host and Guest](#): introduction to different methods for data transfer to/from the Charon-SSP host that are intended to support migration situations.
- [Charon-SSP License Management](#): introduction to the Charon-SSP license management tools for Sentinel HASP licenses and a basic overview of license management for VE licenses.
- [SSH VPN – Connecting Charon Host and Guest to Customer Network](#): example of configuring an encrypted tunnel to the Charon-SSP host across a public network.
- [Charon-SSP Software Upgrade and Charon-SSP Software Deinstallation](#)
- Appendices:
 - [Configuration File Reference](#)
 - [OpenBoot Console Reference](#)
 - [Command-Line Utilities Reference](#)
 - [Charon Manager and Director for Microsoft Windows](#)
 - [Additional Information for Cloud Images](#)
 - [Index](#)

1.3 Products Covered in this Guide

This user's guide covers the configuration and management of all Charon-SSP products. For cloud-specific marketplace images, there is a separate, cloud-specific *Getting Started* guide to cover the cloud-specific details

Charon-SSP products discussed in this user's guide:

1. **Conventional product** provided as individual RPM installation packages that make up the overall product:
 - The emulator software itself. This includes the packages for conventional on-premises installations (using HASP licensing) and for virtual environment installations (installation in supported cloud environments, on-premises VMware, or physical server installations) that require a VE license server:
 - Charon-SSP/4M for Linux
 - Charon-SSP/4U(+) for Linux
 - Charon-SSP/4V(+) for Linux
 - Management components of the product:
 - Charon-SSP Agent for Linux
 - Charon-SSP Manager for Linux and Microsoft Windows
 - Charon-SSP Director for Linux and Microsoft Windows
2. **Baremetal appliance** provided as an ISO installation file:

This appliance includes the full Charon-SSP software set and the underlying host operating system. The system is managed mainly via the customized Charon Baremetal GUI. However, the user can access the host operating system if required.
3. **Prepackaged cloud-specific images** that are provided on the different cloud marketplaces. They include the full Charon-SSP software set and the underlying host operating system and may be available on a cloud marketplace in one or both of the following configurations:
 - a. Cloud-specific Charon-SSP Automatic Licensing (AL) image using a public, Stromasys-operated, cloud-specific license server.
 - b. Cloud-specific Charon-SSP Virtual Environment (VE) image using a customer-operated, private VE license server.

These images are provided via the marketplaces of different cloud providers.

At the time of writing, it was planned to provide one or both image types on the AWS, OCI, Azure, and GCP marketplaces. Please always refer to the cloud-specific **Getting Started Guides** of the Charon-SSP products on the [Charon-SSP documentation page](#) to find up-to-date information about marketplace availability.

Important information regarding the Barebone and Baremetal software variants:

Starting with Charon-SSP 4.1.21, the former Barebone image has been merged with the former Baremetal image to form the new Baremetal version. The new Baremetal version is covered by this user's guide. There is no longer a separate user's guide for Baremetal.

If you are currently running a Barebone installation from a previous version, you can upgrade Charon-SSP using the appropriate RPM files. However, there is no longer a Barebone installation ISO.

Charon-SSP product packaging overview:

The following table provides an overview of Charon-SSP packaging, the associated licensing, and the applicable product documentation:

| Basic Packaging | Packaging Variants | | Installation Location ⁵⁾ | Licensing | Documentation |
|---|--|--------------------------|--|---|---|
| Prepackaged Versions including underlying host OS and Charon-SSP software | Cloud marketplace image | Automatic Licensing (AL) | Cloud only | Public, Stromasys-operated license server ¹⁾ or Customer-operated VE license server in AutoVE mode | <ul style="list-style-type: none"> Cloud-specific Getting Started Guide General Charon-SSP User's Guide VE License Server Guide (for AutoVE) |
| | | Virtual Environment (VE) | | Private, customer-operated VE license server ²⁾ | <ul style="list-style-type: none"> Cloud-specific Getting Started Guide General Charon-SSP User's Guide VE License Server Guide |
| | Baremetal ISO | | On-premises (physical HW or VM) | HASP license (local or via license server) | <ul style="list-style-type: none"> General Charon-SSP User's Guide |
| Individual RPMs | VE-enabled emulator | | Cloud or on-premises (physical HW or VMware VM ³⁾) | Private customer-operated VE license server | <ul style="list-style-type: none"> General Charon-SSP User's Guide VE License Server Guide |
| | Non VE-enabled emulator | | On-premises (physical HW or VM) | HASP license (local or via license server) | <ul style="list-style-type: none"> General Charon-SSP User's Guide |
| | Marketplace RPM packages ⁴⁾ | | Cloud only | Public Stromasys-operated license server or Customer-operated VE license server in AutoVE mode | <ul style="list-style-type: none"> Cloud-specific Getting Started Guide General Charon-SSP User's Guide |
| <p>¹⁾ Use of public license server requires Internet access of Charon host. AutoVE option enables an AL image to use a VE license server.</p> <p>²⁾ Typically the VE license server is installed in the same cloud environment - either on the Charon host itself or a separate license server system.</p> <p>³⁾ For the VMware case, there are some restrictions:</p> <ul style="list-style-type: none"> The Charon emulator must be on the same VM as the VE license server, or on a VM running on the same ESXi host as the VE license server (ESXi binding), or on a VM running on an ESXi host managed by the same vCenter Server as the ESXi host on which the VE license server VM runs (vCenter binding). <p>⁴⁾ Only for updating the Charon-SSP emulator software on hosts based on Charon-SSP AL cloud-specific marketplace images.</p> <p>⁵⁾ Non VE-enabled packages can also be installed in cloud environments. However, this is not recommended as HASP licenses are not suitable for clouds.</p> | | | | | |

Figure 1: Charon-SSP variant overview

Please refer to the [Stromasys product documentation for Charon-SSP](#) for additional user's guides and to section [Charon-SSP Product Variant Comparison](#) for important functional differences between the product variants.

1.4 Obtaining Documentation

The latest released version of this manual is available on the Stromasys support website on the [Stromasys Charon-SSP documentation page](#).

1.5 Obtaining Technical Assistance or Product Information

1.5.1 Obtaining Technical Assistance

Several support channels are available to cover the Charon-SSP SPARC virtualization product.

- **If you have a support contract with Stromasys**, please visit <http://www.stromasys.com/support/> for up-to-date support telephone numbers and business hours. Alternatively, the support center is available via email at support@stromasys.com.
- If you purchased the Charon-SSP product through a Value-Added Reseller (VAR), please contact them directly.

1.5.2 Obtaining General Product Information


If you require information in addition to what is available on the Stromasys [Product Documentation and Knowledge Base](#) and on [the Stromasys web site](#) you can contact the Stromasys team using <https://www.stromasys.com/contact/>, or by sending an email to info@stromasys.com.

For further information on purchases and the product best suited to your requirements, you can also contact your regional sales team by phone:

| Region | Phone | Address |
|--------------------------------|-----------------|---|
| Australasia-Pacific | +852 3520 1030 | Room 1113, 11/F, Leighton Centre 77 Leighton Road, Causeway Bay, Hong Kong, China |
| Americas | +1 919 239 8450 | 2840 Plaza Place, Ste 450 Raleigh, NC 27612, USA |
| Europe, Middle-East and Africa | +41 22 794 1070 | Avenue Louis-Casai 84 5th Floor 1216 Cointrin, Switzerland |

1.6 Conventions

Throughout the document, the following conventions apply:

| Notation | Description |
|---|---|
| \$ | The dollar sign indicates a reference to the environment variables for UNIX / Linux variants. |
| # | The number sign represents the super-user prompt for UNIX / Linux. |
| User input | Bold type in interactive examples indicates typed user input. |
| <path> | Bold monospace type enclosed by angle brackets indicates command parameters and parameter values. |
| Output | Monospace type in interactive examples indicates command response output. |
| [] | In syntax definitions, brackets indicate optional items. |
| ... | In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times. |
| <i>disk0</i> | Italic monospace type, in interactive examples, indicates typed, context dependent user input. |
|  | This symbol represents the Enter key without typed user input. Used, for example, to tell the user to select the default value by pressing enter. |
| { <i>version</i> } | Indicates version of Charon-SSP release in the format major.minor.revision for example: 4.4.1 |

The following document specific definitions apply:

| Term | Description |
|---------|---|
| Host | The hardware and the Linux 64-bit-system on which Charon-SSP/4M/4U/4V runs. |
| Guest | The virtual SPARC (or emulated SPARC) created by Charon-SSP/4M/4U/4V on the host. |
| Linux | Any Linux version supported as a host system for Charon-SSP. |
| Windows | All supported versions of Microsoft Windows. |

Emulated SPARC virtual machines are available for the following architectures:

- **Charon-SSP/4M:** 32-bit SPARC V8 Sun-4m architecture
- **Charon-SSP/4U(+):** 64-bit SPARC V9 Sun-4u architecture
- **Charon-SSP/4V(+):** 64-bit SPARC V9 Sun-4v architecture with additional features

The products listed above support almost identical configuration mechanisms, system console, and interfaces. Therefore, options, interfaces, etc. that apply to all emulators will be collectively called Charon-SSP in this document. Platform-specific features will be identified and the relevant platform will be specified.

2 Charon-SSP Licensing Options Overview

As shown above, there are different licensing options applicable to one or more of the different product variants. Please note that this relates to Charon product licensing only. The user is responsible for any Solaris licensing obligations and must provide the appropriate licenses. This section provides an overview of the Charon-SSP product licensing options.

2.1 Sentinel/Gemalto HASP Licenses

Sentinel HASP licenses are the "traditional" licensing method for Charon emulator products. Main characteristics:

- Software and hardware (dongle) licenses.
- Based on third-party vendor solution.
- Require special third-party license driver software (provided as part of the Charon-SSP product kits).
- Installed on Charon-SSP host or separate license server.
- Software licenses are tied to certain characteristics of the physical or virtual hardware.
- Problematic in cloud environments (no USB ports, volatile virtual hardware characteristics).
- USB dongles are a flexible and host-hardware independent solution for on-premises installations.
- The customer is billed for the license by Stromasys.

Please refer to the section [Charon-SSP License Management](#) for details about managing such licenses.

2.2 Charon-SSP Automatic Licensing for Cloud Environments

When installing Charon-SSP AL from a supported cloud marketplace, the cloud instance automatically receives a license at first launch. The license server must be reachable via a cloud-specific public IP address. The license server is operated by Stromasys. The customer is billed by the cloud provider. See also **AutoVE mode** which allows a customer to run their own license server for an instance launched from such a marketplace image.

2.3 Virtual Environment (VE) Licenses

Other license types have drawbacks that prompted the development of VE licenses:

- USB dongles are not suitable for cloud environments. Their use in VMware environments is complex.
- Sentinel software licenses are easy to install in a cloud or VMware environment. However, their ties to real hardware characteristics also makes it easy to inadvertently invalidate them in such environments. Hence, they are not suited for use in cloud environments and difficult to use on VMware. Other Sentinel software license types do not provide the same level of license security.
- Charon-SSP automatic licensing does not allow a customer-specific environment in the cloud without Internet access. This is not suitable for many customers who require a private cloud network environment complying with their own security and management policies.

VE licenses are designed to enable the ease-of-use of software licenses while providing a high level of security for licensing Charon-SSP products in virtualized environments (supported cloud and VMware environments, physical servers). The main characteristics of VE (Virtual Environment) licenses are the following:

- Software licenses only.
- Developed by Stromasys.
- Installed on Charon-SSP host or separate license server.
- Require the Charon-SSP VE license server software.
- Require matching Charon-SSP VE-enabled emulator software.
- Different modes of operation:
 - For **general VE mode**, the customer is billed by Stromasys depending on the number and type of the emulated systems allowed by the installed license(s). The license server software itself is free of charge.
 - **AutoVE mode** is an extension of automatic licensing and introduces metered billing (by the cloud-provider) for VE licenses in cloud environments. It defines how many Charon emulator cloud instances can be run based on the respective license. The number of emulated systems on each host instance is limited by the host resources, not the license. Charon-SSP supports this mode starting with Charon-SSP 5.3.8 (only if provided as part of a Charon-SSP AL marketplace image).
 - The license server for **both modes** is managed by the customer.
- Support at the time of writing:
 - **VE license server availability**: supported clouds (at the time of writing: AWS, OCI, Azure, GCP, IBM, and Nutanix), supported VMware environments, and physical servers.
 - **AutoVE support**: Charon-SSP AL (Automatic Licensing) marketplace images version 5.3.8 or higher on AWS, OCI, GCP, and Azure.
 - **Charon emulator product support**: Charon-SSP VE-capable emulator software (the emulator package names include the string “ve”).

Please refer to the [VE License Server Guide](#) for details about managing licenses in such environments.

3 Charon-SSP Product Overview

3.1 General Information

In 1987, Sun Microsystems released the SPARC V7 processor, a 32-bit RISC processor. The SPARC V8 followed in 1990 – a revision of the original SPARC V7, with the most notable inclusion of hardware divide and multiply instructions. The SPARC V8 processors formed the basis for several servers and workstations such as the SPARCstation 5, 10 and 20. In 1993, the SPARC V8 was followed by the 64-bit SPARC V9 processor. This too became the basis for several servers and workstations, such as the Enterprise 250 and 450.

Due to hardware obsolescence and lack of spare or refurbished parts, software and systems developed for these older SPARC-based workstations and servers have become harder to maintain. To fill the continuous need for certain, end-of-life SPARC-based systems, Stromasys S.A. developed the Charon-SSP line of SPARC emulator products. Charon-SSP products are software-based, virtual machine replacements for the specified native-hardware SPARC systems.

The Charon-SSP virtual machines allow users of Sun and Oracle SPARC-based computers to replace their native hardware in a way that requires little or no change to the original system configuration. This means you can continue to run your applications and data without having to switch or port to another platform. The Charon-SSP software runs on commodity, Intel 64-bit systems ensuring the continued protection of your investment.

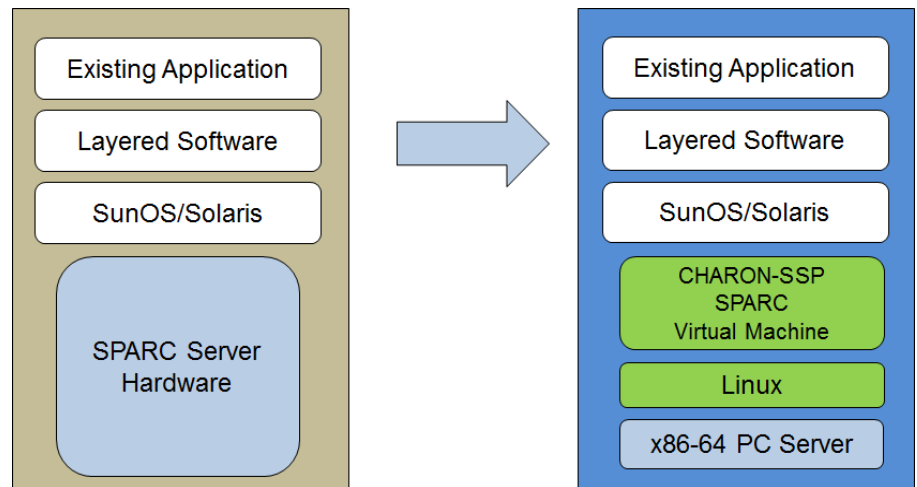


Figure 2: Seamless migration from SPARC hardware to virtual SPARC on x86-64

A general overview of the emulated hardware families is shown below:

Charon-SSP/4M emulates the following SPARC hardware:

Sun-4m family (represented by the Sun SPARCstation 20):

Originally, a multiprocessor Sun-4 variant, based on the MBus processor module bus introduced in the SPARCServer 600MP series. The Sun-4m architecture later also encompassed non-MBus uniprocessor systems such as the SPARCstation 5, utilizing SPARC V8-architecture processors. Supported starting with SunOS 4.1.2 and by Solaris 2.1 to Solaris 9. SPARCServer 600MP support was dropped after Solaris 2.5.1.

Charon-SSP/4U(+) emulates the following SPARC hardware:

Sun-4u family (represented by the Sun Enterprise 450):

(U for UltraSPARC) – this variant introduced the 64-bit SPARC V9 processor architecture and UPA processor interconnect first used in the Sun Ultra series. Supported by 32-bit versions of Solaris starting from version 2.5.1. The first 64-bit Solaris release for Sun-4u was Solaris 7. UltraSPARC I support was dropped after Solaris 9. Solaris 10 supports Sun-4u implementations from UltraSPARC II to UltraSPARC IV.

Charon-SSP/4V(+) emulates the following SPARC hardware:

Sun-4v family (represented by the SPARC T2 and T4):

This variation added hypervisor processor virtualization to the Sun-4u; introduced in the UltraSPARC T1 multicore processor. Selected hardware was supported by Solaris version 10 starting from release 3/05 HW2. Most models - including the hardware emulated by Charon-SSP - require newer versions of Solaris 10. Several Solaris 11 versions are also supported.

For up-to-date information about supported features and versions refer to the sections [Supported Guest Operating Systems](#) and [Supported Virtual Hardware](#) below and to the release notes of your product.

Charon-SSP/4U+ supports the same virtual SPARC platforms as Charon-SSP/4U, and **Charon-SSP/4V+** the same as Charon-SSP/4V. However, the 4U+ and 4V+ versions take advantage of Intel's VT-x/EPT and AMD's AMD-v/NPT hardware assisted virtualization technology in modern CPUs to offer better virtual CPU performance. Charon-SSP/4U+ and Charon-SSP/4V+ require CPUs with VT-x/EPT or AMD-v/NPT support and **must** be installed on a dedicated host system. Running these product variants in a VM (e.g., on VMware) is **not supported**.

Please note: unless otherwise mentioned, the terms Charon-SSP/4U and Charon-SSP/4V also include Charon-SSP/4U+ and Charon-SSP/4V+.

3.2 Supported Virtual Hardware

The different families of Charon-SSP virtual machines support different emulated hardware devices. The tables below describe the device features and maximum numbers supported by the different Charon-SSP virtual machine families.

The available virtual hardware also depends on the installation environment. For example, if the product is installed in a cloud environment, not all the hardware supported by the product on non-cloud servers will be available.

3.2.1 Supported Virtual Hardware in non-Cloud Installations

| Charon-SSP supported virtual hardware (non-cloud installation) | | | |
|--|--|---|--|
| | Charon-SSP/4M | Charon-SSP/4U(+) ⁽¹⁾ | Charon-SSP/4V(+) ⁽¹⁾ |
| SPARC V8 (32-bit) | Y | — | — |
| SPARC V9 (64-bit) | — | Y ⁽²⁾ | Y ⁽⁵⁾ |
| Max. number of CPUs | 4 | 24 | 64 |
| Max. RAM | 64MB to 512MB | 1GB to 128GB | 1GB to 1024GB ⁽⁶⁾ |
| Ethernet controllers | 2 (controller type: le) | 19 (controller types: hme, qfe, bge ⁽⁹⁾) | 8 (controller types: bge, qfe) |
| SCSI controllers | 1 | 2 | 2 |
| SCSI target IDs | 7 ⁽³⁾ | 30 ⁽³⁾ | 30 ⁽³⁾ |
| VDS storage devices | — | — | 120 ⁽⁸⁾ |
| Serial ports | 2 built-in ports and 8 ports as SBus card (STC) emulation | 2 built-in ports, PLUS 14 ports in on-board mode emulation, OR 32 ports as Digi AccelePort 920 emulation, OR PCI pass-through ⁽⁴⁾ with 32 ports (4 x Digi AccelePort 920) or 8 ports (4 x Digi AccelePort C/X) | Vconsole, PLUS 2 built-in ports, PLUS 14 ports in on-board mode emulation |
| Parallel ports | 1 | — | — |
| Floppy drive | 1 | 1 ⁽⁹⁾ | — |
| Graphics controller | 1 (CGTHREE or CGSIX ⁽⁷⁾) | 1 (CGSIX or RAGE XL) | — |
| Audio controllers | 1 (DBRIe) | 1 (DBRIe) | — |
| GPIB controller | — | NI-488.2 GPIB device in PCI pass- through mode ⁽⁴⁾ | — |
| USB ports | — | 1 | 1 |

⁽¹⁾ Charon-SSP/4U+ has same virtual hardware specification as Charon-SSP/4U, Charon-SSP/4V+ the same as Charon-SSP/4V. 4U+ and 4V+ are only supported on physical (baremetal) hardware with Intel VT-x/EPT or AMD-v/NPT support. This hardware must run a Linux kernel supported by Stromasys (see [Host System Software Requirements](#)).

⁽²⁾ SPARC V9 is backward compatible. Hence, Charon-SSP/4U can also support V8 32-bit systems.

⁽³⁾ Each SCSI target ID can have up to 8 LUNs. Therefore, the overall number of SCSI devices can be larger than the number of target IDs. The exact number depends on the emulated hardware, the guest operating system and driver versions, and the SCSI devices used.

⁽⁴⁾ PCI pass-through is only supported when running on supported product variants and Linux kernels (see [Host System Software Requirements](#)).

⁽⁵⁾ Charon-SSP/4V supports one LDom per instance. An LDom virtual disk image can be booted by Charon-SSP directly (system device configuration may have to be adjusted for differences in emulated hardware).

⁽⁶⁾ Actual maximum values are different depending on guest OS: Solaris 10: 1TB, Solaris 11: 512GB.

⁽⁷⁾ CGSIX emulation is not supported for SunOS 4.x guest systems.

⁽⁸⁾ Guest systems starting with Solaris 10-U6 can use VDS and/or SCSI storage devices.

⁽⁹⁾ The BGE controller for Charon-SSP/4U is supported starting with versions 5.2.5 and 5.3.4. Floppy support was added in version 5.3.1).

3.2.2 Supported Virtual Hardware in Cloud Installations

| Charon-SSP supported virtual hardware (cloud installation) | | | |
|--|--------------------------------------|---|-------------------------------------|
| | Charon-SSP/4M | Charon-SSP/4U(+) ⁽¹⁾ | Charon-SSP/4V(+) ⁽¹⁾ |
| SPARC V8 (32-bit) | Y | — | — |
| SPARC V9 (64-bit) | — | Y ⁽²⁾ | Y ⁽⁴⁾ |
| Max. number of CPUs | 4 | 24 | 64 |
| Max. RAM | 64MB to 512MB | 1GB to 128GB | 1GB to 1024GB ⁽⁵⁾ |
| Ethernet controllers | 2 (controller type le) | 19 (controller types: hme, qfe, bge ⁽⁸⁾) | 8 (controller types bge and qfe) |
| SCSI controllers | 1 | 2 | 2 |
| SCSI target IDs | 7 ⁽³⁾ | 30 ⁽³⁾ | 30 ⁽³⁾ |
| VDS storage devices | — | — | 120 ⁽⁷⁾ |
| Serial ports | 2 | 2 | 2 + Vconsole |
| Graphics controller | 1 (CGTHREE or CGSIX ⁽⁶⁾) | 1 (CGSIX or RAGE XL) | — |
| Audio controllers | 1 (DBRIe) | 1 (DBRIe) | — |

⁽¹⁾ Charon-SSP/4U+ has same virtual hardware specification as Charon-SSP/4U, Charon-SSP/4V+ the same as Charon-SSP/4V. 4U+ and 4V+ are only supported on physical (baremetal) hardware with Intel VT-x/EPT or AMD-v/NPT support. This hardware must run a Linux kernel supported by Stromasys (see [Host System Software Requirements](#)).

⁽²⁾ SPARC V9 is backward compatible. Hence, Charon-SSP/4U can also support V8 32-bit systems.

⁽³⁾ Each SCSI target ID can have up to 8 LUNs. Therefore, the overall number of SCSI devices can be larger than the number of target IDs. The exact number depends on the emulated hardware, the guest operating system and driver versions, and the SCSI devices used.

⁽⁴⁾ Charon-SSP/4V supports one LDom per instance. An LDom virtual disk image can be booted by Charon-SSP directly (system device configuration may have to be adjusted for differences in emulated hardware).

⁽⁵⁾ Actual maximum values are different depending on guest OS: Solaris 10: 1TB, Solaris 11: 512GB.

⁽⁶⁾ CGSIX emulation is not supported for SunOS 4.x guest systems.

⁽⁷⁾ Guest systems starting with Solaris 10-U6 can use VDS and/or SCSI storage devices.

⁽⁸⁾ The BGE controller for Charon-SSP/4U is supported starting with versions 5.2.5 and 5.3.4.

3.3 Charon-SSP Product Variant Comparison

When looking at the Charon-SSP product features, one can look at the differences between the different emulated models (as shown above in [Supported Virtual Hardware](#)).

Another comparison is the comparison between the different product variants. This section provides an overview of important differences between some of the product variants.

3.3.1 Product Variant Overview

The basic functionality of Charon-SSP in the different product variants is very similar. However, the product variants also have important differences. This section provides a brief (incomplete) overview.

Conventional RPM installation (Sentinel HASP licenses)

The product is installed as individual RPM packages on a supported Linux distribution and version. For on-premises installations, it offers the greatest flexibility for any customization and for integration into the customers' system management environments.

Please note: Starting with version 4.1.21, the Barebone ISO is no longer available. This functionality has been merged with the Baremetal product. If you are currently running a Barebone installation from a previous version, you can upgrade Charon-SSP using the appropriate RPM files.

Baremetal product (Sentinel HASP licenses)

Charon-SSP Baremetal is a software appliance distributed as an ISO installation file. The main management interface is the customized Baremetal GUI. However, the user can access the underlying host operating system if required. The Baremetal variant provides a fast and easy way to set up Charon-SSP in an on-premises scenario.

Please note: Starting with version 4.1.21, the former products Barebone and Baremetal have been merged to form the new Baremetal product. The table below refers to this new product.

Cloud-specific Charon-SSP Automatic Licensing (AL) versions

Cloud-specific versions of Charon-SSP provided in supported cloud marketplaces provide a Charon-SSP AL image that can be used to easily launch a Charon-SSP host as a cloud instance containing all the necessary software. Licensing is set up automatically at launch, and usage is billed through the cloud service provider. The Charon-SSP host must have access to the Internet for licensing to work.

Charon-SSP for use with VE (Virtual Environment) licenses

Very similar to the non-VE conventional product with respect to installation and management. However, it is mainly intended for installation in cloud and VMware environments. This product variant is provided for **conventional RPM installations** and may also be provided by Stromasys as **cloud-specific images on the marketplaces** of supported cloud-providers.

The main differences when compared to a traditional on-premises installation are the following:

- This product will typically be installed in a cloud or VMware environment. In such environments there are restrictions regarding the supported virtual hardware (e.g., PCI pass-through devices and USB removable storage cannot be used).
- Licensing is provided by a VE license server software package on the emulator host system itself or on a Linux system in a supported cloud or VMware environment (see the *VE license server guide* for details). The customer manages the license server and purchases the license from Stromasys.

Charon-SSP for VE licensing provides a flexible licensing model in a private, customer-specific cloud or VMware environment. The Charon host does not require access to the public Internet for licensing to work. At the same time, this product variant also offers the flexibility and customizability of the conventional RPM-based product variant.

3.3.2 Product Variant Comparison

The following table lists important differences between three of the Charon-SSP product variants (as opposed to the differences between the different emulated architectures):

| Functionality differences | Conventional and VE (RPM) | Baremetal | AL cloud images ⁽⁶⁾ |
|--|---|----------------------|--|
| General differences | | | |
| Linux operating system upgrades from distribution repositories | Restricted (kernel version dependencies for 4U+/4V+ and PCI pass-through) | | |
| Special GUI for host management | N | Y | N |
| Special user accounts for Charon | N ⁽⁷⁾ | Y ⁽⁴⁾ | Y ⁽⁴⁾ |
| Licensing general | Conventional: HASP HL/SL/Network license | | Stromasys-managed public license server, or customer-managed AutoVE license server |
| | VE-enabled emulator versions: customer-managed VE license server (not on Baremetal) | | |
| Changes to number of host CPUs possible | If VE or HASP software license is used, new license may be needed; software and hardware license may have to be updated | | Invalidates license; requires setup of new cloud instance for AL licensing |
| Internet connection required | N | | Y |
| Jumpstart | Y ⁽⁸⁾ | | N |
| Network interface sharing (not recommended) | Y (non-cloud only) | | N |
| Configurable log path | Y | N | N |
| Additional tools | X11, iSCSI, NFS | X11, iSCSI, NFS, VNC | X11 |
| Emulated HW differences | | | |
| 4U+ and 4V+ support | See host Software Requirements | Y | Y ⁽⁵⁾ |
| PCI pass-through devices (Digi and GPIB) | See host Software Requirements | Y ⁽²⁾ | N |
| Digi AccelePort emulation | Y ⁽²⁾ | | N |
| Additional on-board serial lines | Y ⁽⁸⁾ | | N |
| USB devices | Y ⁽²⁾ | | N |
| Parallel port | Y ⁽³⁾ | | N |
| Floppy drive | Y ⁽³⁾ | | N |
| Physical SCSI devices | disk, tape, CD-ROM, generic ⁽⁹⁾ | | disk |
| External serial console via TCP | Y | | N |
| Physical serial ports | Y ⁽⁹⁾ | | Only via terminal server |
| Host HW differences | | | |
| Customer selectable hypervisor support | Y ⁽¹⁾ | | N |

Notes

- (1) Not for Charon-SSP/4U+/4V+ (require VT-x/EPT or AMD-v/NPT support); supported Hypervisors are listed in *Host System Requirements*.
- (2) Only supported on Charon-SSP/4U(+); not supported when installed in cloud environment.
- (3) Charon-SSP/4M/4U only
- (4) User **charon** for GUI operation, SFTP access, and Charon Manager integrated SSH tunnel. User **sshuser** for setting up the general SSH VPN tunnel and for interactive command-line access, **root** access possible.
- (5) Normal cloud instances run on shared hardware; a "baremetal" virtual hardware type must be offered by the cloud provider to run Charon-SSP/4U+/4V+. Please contact Stromasys or your Stromasys VAR if you require this type of emulated SPARC hardware.
- (6) Similar product characteristics are to be expected for all cloud-specific Charon-SSP AL images.
- (7) If provided as a pre-packaged VE image on a cloud marketplace, it has the same user accounts as the AL images.
- (8) Not supported in cloud environment.
- (9) Physical SCSI device and VDS device support in cloud environments: disk only
Physical serial line support in cloud environments: only via terminal server.

4 Host System Requirements

To ensure good performance for the emulated SPARC systems, it is important to follow some guidelines regarding the setup of the Charon-SSP host system as described in the following sections.

4.1 Minimum Host System Hardware Requirements

The sizing guidelines below show the minimum requirements.

Every deployment situation must be reviewed and the host sizing must be adapted to the actual workload.

For example, the number of CPU cores available for I/O must be increased if the guest applications generate a high I/O load. A system with many emulated CPUs is usually able to create a higher I/O load and thus the number of CPU cores available for I/O may have to be increased.

In a hyper-threading environment, the number of CPU cores should be sufficient to avoid high-workload threads sharing one physical CPU core.

Customer-specific workload analysis and emulator host sizing is beyond the scope of this user's guide. Please contact your Stromasys technical representative for guidance.

The CPU allocation for emulated CPUs and the CPU allocation for I/O processing are determined by the configuration. See [CPU Configuration](#) for more information about this and the default CPU allocation for I/O processing.

Minimum requirements for the host system hardware:

- Charon-SSP requires modern x86-64 architecture processors. This could be Intel Servers based on Haswell v3 processors or later, or Desktop Core I7 (CPU frequency at least 3.0 GHz). AMD processors of the same or higher performance are also supported.
- Minimum number of host system CPU cores:
 - At least one CPU core for the host operating system.
 - **For each emulated SPARC system:**
 - One CPU core for each emulated CPU of the instance.
 - At least one additional CPU core for I/O processing (at least two, if server JIT optimization is used). By default, Charon will assign 1/3 (min. 1; rounded down) of the CPUs visible to the Charon host to I/O processing. See [CPU Configuration](#) for configuration options.
- Minimum memory requirements:
 - 4GB or more of RAM for the Linux host operating system. The actual requirements may be higher and will depend on the requirements of the non-emulator services running on the Linux host.
 - **For each emulated SPARC system:**
 - The configured memory of the emulated instance.
 - 2GB of RAM (6GB of RAM if server JIT is used) to allow for DIT optimization, emulator requirements, run-time buffers, SMP and graphics emulation.
- Charon-SSP/4U+ and Charon-SSP/4V+ must run on physical (baremetal) hardware supporting Intel VT-x/EPT or AMD-v/NPT.

- If hyper-threading is enabled on modern x86-64 CPUs, two threads can run on one physical CPU core providing two logical CPUs to the host operating system. If possible, disable hyper-threading for the Charon-SSP host system. However, this is frequently not possible in VMware and cloud environments. The Charon-SSP hyper-threading option enables Charon-SSP to adapt to such environments. See [CPU Configuration](#) for detailed configuration information.
Please note: under sustained high workloads, Charon-SSP threads should not share a physical CPU core – enough physical cores should be available on the host system to satisfy the requirements of the configured emulator(s).
- At least one available USB port (if USB license key is used). More for additional USB devices.
- Single dedicated network adapter for each configured virtual network adapter in Charon-SSP **unless** the virtual network functionality of Charon-SSP is used. In this case, TAP interfaces attached to a virtual bridge on the host system can be used as the basis for Charon-SSP virtual network adapters.
- Free PCI slots to install serial line cards if the Digi PCI pass-through feature is to be used.
- Free PCI or PCIe slots to install GPIB cards if the GPIB PCI pass-through feature is to be used.
- Optical drive if DVD installation media is to be used.
- Enough disk space for the host operating system and any virtual disk/tape container files required by the guest operating system.

Requirements for running the Charon-SSP Linux host under a Hypervisor:

Charon-SSP/4U+ and Charon-SSP/4V+ utilize special hardware functionality to deliver improved performance. Due to the hardware requirements, they can only be used on real hardware providing the correct CPU features. Running Charon-SSP/4U+ or Charon-SSP/4V+ in a VM is not supported. 4U+ and 4V+ installation kits contain also the non-plus versions. Make sure you select the correct variant when configuring your Charon-SSP instance.

The Charon-SSP host system can run on dedicated hardware or as a VM under a Hypervisor.

Please note: if a hypervisor is to be used and if the Charon-SSP emulator software is to be used with a VE license, only VMware is supported. See the *VE License Server User's Guide* for supported versions.

Supported hypervisors for hosting Charon-SSP host systems:

- VMware ESXi 6.5 and above
- Xen
- Microsoft Hyper-V (local HASP dongles are not supported for licensing because they cannot be passed through to the Linux guest system; alternatives are network license, software license, or network-based USB port servers)
- Linux KVM

If a VM is used:

- Configure it to support a Linux x86_64 environment and follow the hardware requirements listed above to configure the virtual hardware with enough capacity for all instances of Charon-SSP that are to run on it.
- Network adapters must support promiscuous mode, or the MAC address of the emulated adapter must be hard-coded to the MAC address of the vNIC presented to the Charon-SSP host (see Ethernet configuration section).
- Guest additions (e.g., VMware Tools) that enhance the usability of the guest system (e.g., video capabilities, mouse integration, shared folders) may be installed, but they are not prerequisites.

4.2 Host System Software Requirements

This section provides an overview of the software prerequisites for running Charon-SSP on **Linux**.

Should you have received Charon Manager and Charon Director kits for **Microsoft Windows**, please refer to [Charon-SSP GUI for Microsoft Windows](#) in the appendix.

4.2.1 Linux Operating System Prerequisites

The Charon-SSP emulator products run on Linux systems. Stromasys supports the following Linux distributions and releases as host environments for Charon-SSP. However, **please read the important restrictions below**

- 64-bit versions 7.0 to 7.9 of Oracle Linux, Red Hat Enterprise Linux, and CentOS
- 64-bit versions 8.1 to 8.5 of Oracle Linux, Red Hat Enterprise Linux, CentOS, and Rocky Linux.
 - **Please note** that as of 1 January 2022 CentOS 8 is EOL. For new deployments, it is recommended to use a non-EOL alternative. For existing installations, the possible negative impacts of staying with an EOL host operating system should be carefully evaluated.
- Baremetal system (host operating system is included for this Charon-SSP product variants)

The Linux distributions **listed above** will be referred to in this guide as “**the supported Linux distributions**”.

Charon-SSP Manager and Director are also available on

- Ubuntu 17 or higher (64 bit)
- Microsoft Windows 7, 8, 10

Important restrictions:

- If the configuration on Linux version 7.x includes a dual emulated graphics display, use Linux version 7.3, or 7.7 and higher.
- **Charon-SSP/4U+ and Charon-SSP/4V+** must run on physical hardware (see [Minimum Host System Hardware Requirements](#)) and are only available for Linux kernels supported by Stromasys, namely:
 - Original kernel version provided by the Baremetal distribution
 - Original kernel versions provided by cloud-specific marketplace images (support depends also on the physical hardware options offered by the cloud provider)
 - Original kernel versions provided by RHEL/CentOS 7.9, 8.2 to 8.4 (starting with Charon-SSP version 5.2.7), and 8.5 (starting with Charon-SSP version 5.3.10). Rocky Linux 8.4 and 8.5 support starting with Charon-SSP 5.3.10.
- **PCI pass-through** is only supported on physical hardware running
 - Charon-SSP Baremetal installations with the kernel version provided by the Baremetal distribution, or
 - a general Linux distribution and version supported by Charon-SSP/4U+ (see above). Please note that in Charon-SSP version 5.2.8, the modules for RHEL/CentOS 8.3 to 8.5 are missing.
- The Charon-SSP VE license server is currently available for supported cloud (at the time of writing AWS, OCI, GCP, Azure, Nutanix, and IBM) and VMware environments, and for physical servers.

Anti-virus Software is not normally needed for well-managed Linux servers and can interfere with the function of the emulator. Therefore, it is recommended, not to use anti-virus software on Charon-SSP host systems. If the customer policy requires that anti-virus software be deployed, it should not run while the emulator is running. If this restriction cannot be satisfied, it absolutely must not be used to scan the vdisk container files. Regular security updates to the host operating system are recommended (but observe the kernel version restrictions mentioned above).

4.2.2 Additional Software Requirements

The **Baremetal** product and the pre-packaged cloud-specific images include all required software and additional utilities for Charon-SSP.

When using the **conventional RPM installation**, some of the Charon-SSP components have additional software prerequisites and require specific operating system packages to be installed. Such prerequisites are described in the installation chapter, or together with the configuration task for which they are required.

4.3 Guest Operating System Requirements

The different emulated architectures have different requirements regarding the guest operating systems:

The **Charon-SSP/4M** virtual machines support the following guest operating system releases:

- SunOS 4.1.3 - 4.1.4
- Solaris 2.3 to Solaris 9

The **Charon-SSP/4U(+)** virtual machines support the following guest operating system releases:

- Solaris 2.5.1 to Solaris 10

The **Charon-SSP/4V(+)** virtual machines support the following guest operating system releases:

- Solaris 10 (starting with update 4, 08/07) and Solaris 11.1 to Solaris 11.4

Please note:

- For the emulated SPARC T4, supported Solaris 10 versions are: Oracle Solaris 10 1/13, Oracle Solaris 10 8/11, and Solaris 10 9/10, or Solaris 10 10/09 with Oracle Solaris 10 8/11 patch set.
- The emulated SPARC T4 model is a prerequisite for running Solaris 11.4 in the emulator.

Solaris licensing: the user is responsible for any Solaris licensing obligations and must provide the appropriate licenses.

4.4 NetworkManager Considerations

Important information:

- Network management via the Charon Manager has **changed significantly** between Charon-SSP for versions 7.x of the supported Linux distributions and versions 8.x of these distributions.
- The user interface remains the same, but Charon-SSP uses different operating system methods:
 - The Charon Manager for **versions 7.x** of the supported Linux distributions uses `ifcfg-<interface>` files to store the network configuration **without using the NetworkManager**. This functionality is provided by the **network-scripts** and **bridge-utils** packages.
 - The Charon Manager for **versions 8.x** of the supported Linux distributions uses the **NetworkManager** (via `nmcli` commands). The **network-scripts** and **bridge-utils** packages have been deprecated for these Linux versions and are no longer installed by default. The `ifcfg-<interface>` files still exist and can be edited manually, but every interface that is to be managed by the Charon Manager must be under the control of the NetworkManager.
- **If running Charon-SSP in a cloud environment:** Every cloud environment has specific characteristics that could conflict with interface configurations made manually or via the Charon Manager. Please refer to the documentation provided by the cloud provider and the network-specific sections in the *Getting Started* guides of your product to understand the networking behavior of your cloud instance before you change any interface settings.

The sections below provide some additional information regarding the NetworkManager. Please refer to your host system's manual pages for additional information about the NetworkManager.

4.4.1 NetworkManager and Charon Manager for Linux versions 7.x

As described above, the Charon Manager does not use the NetworkManager to configure host network interfaces if the host operating system is a Linux 7.x system. Interfaces managed by the Charon Manager must be excluded from the control of the NetworkManager.

Please note: the NetworkManager is disabled by default in the Baremetal version of Charon-SSP starting with version 3.1.14 and in the cloud-specific, pre-packaged images provided via the different cloud marketplaces that are based on CentOS 7. **Charon-SSP Baremetal starting with version 5.3.11, and cloud marketplace images starting with SSP version 5.3.12 are based on Rocky Linux 8.5 or higher.** Please refer to the NetworkManager information for Linux 8.x.

The **Charon Manager** offers several functions to manage the host system network interfaces, add virtual bridge interfaces (TAP interfaces), and VLAN interfaces. The Charon Manager depends on **certain naming conventions** regarding interface names and interface configuration file names. Hence, if you manually create interface configuration files, please observe the naming conventions (see [Host System Network Configuration](#)).

In an environment controlled by the NetworkManager, the NetworkManager—if configured accordingly—will create and manage its own interface configuration files. This can sometimes lead to conflicts with the Charon-SSP requirements.

There are several options to prevent such problems:

- If the NetworkManager is not needed for other purposes, you can disable it (commands: `# systemctl stop NetworkManager; systemctl disable NetworkManager`) and create the initial `ifcfg-<interface>` files manually.

Example of a minimal `ifcfg` file:

```
NM_CONTROLLED=no
DEVICE=eth0
NAME=eth0
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
```

- If the NetworkManager is required for other purposes:
 - Make sure it uses the `ifcfg`-files (**plugins=ifcfg-rh** must be enabled in section **[main]** of **/etc/NetworkManager/NetworkManager.conf**),
 - Stop the NetworkManager
`# systemctl stop NetworkManager`
 - Make sure all interfaces required for Charon-SSP have an `ifcfg-<interface>` file following the naming convention described in [Creating a Virtual Network](#), and add the line **NM_CONTROLLED=no** for each interface that should not be controlled by the NetworkManager.
 - Restart the NetworkManager
`# systemctl start NetworkManager`

If no `ifcfg`-file exists for an interface when the network configuration in Charon-SSP Manager is started, the conventional and the Baremetal Charon-SSP product offer to create it using the **Persistence** parameter. This will create the necessary `ifcfg`-file and remove the interface from NetworkManager control. Please refer to chapter [Managing Host System Network Interfaces](#) for more details.

Please note: the **Persistence** parameter is missing from older cloud-specific Charon-SSP images that are based on Linux 7.x. Here, an `ifcfg`-file must be manually created before a new interface can be managed by the Charon Manager.

4.4.2 NetworkManager and Charon Manager for Linux versions 8.x

Starting with version 8.x of the supported Linux distributions, the **network-scripts** and **bridge-utils** packages have been deprecated and are no longer installed by default.

For this reason, the network management part of the Charon Manager for these Linux versions has been changed to use the NetworkManager capabilities via **nmcli** commands. The relevant Charon Manager GUI has not changed, and the naming conventions mentioned above still apply.

The NetworkManager is a prerequisite for using the Charon Manager network management features when running Charon-SSP on a Linux host with version 8.x. **Only interfaces under NetworkManager control are considered as usable interfaces by the Charon Manager.**

If an interface configuration contains the `NM_CONTROLLED=no` statement, it will not be offered for use by the Charon Manager. Such interfaces are listed as *unmanaged* in the output of `nmcli device` and not listed in the Charon Manager Network Settings section.

The NetworkManager also creates ifcfg-files in `/etc/sysconfig/network-scripts` when changing the default configuration with **nmcli**.

The command

```
nmcli -f NAME,DEVICE,FILENAME connection show
```

displays information about where the configuration files are stored. Only change these files manually after informing yourself in detail about the operation of the NetworkManager. Changes in the network configuration that conflict with the operation of the Charon Manager can lead to problems if manual configuration and configuration via the Charon Manager are mixed. Manually changing the configuration files requires a reload of the configuration (**nmcli connection reload**) or a restart of the NetworkManager. Sometimes, for changes to become active, it may also be required to manually delete a connection.

Please note:

- For more information about the options provided by **nmcli**, please refer to the documentation and the manual pages (`man nmcli`) of your host operating system.
- If GUI access is not possible, the **nmtui** utility provides a **text-based** front-end to **nmcli**. After changing an interface in **nmtui**, you must deactivate and reactivate the interface for the changes to become active.
- **Caveat:** If the configuration stored on disk does not match the active configuration on the device, starting or restarting NetworkManager creates an additional in-memory connection that reflects the configuration of the device.

4.5 Firewall Requirements

This section provides an overview of the firewall requirements when running Charon-SSP.

4.5.1 Frequently used TCP and UDP Ports

Please note: The ports used by a Charon-SSP installation will be different depending on the applications running on the host system and on the guest Solaris system. They will also depend on the configured Charon-SSP features. The information in this section is provided for information only and can never be totally complete.

The following table provides an overview of frequently used network ports in a Charon-SSP installation:

| Component | Port(s) | Purpose |
|---|---|--|
| Charon-SSP Agent | 9091 (TCP and UDP) | Communication with Charon Manager and Charon Director |
| | 9101 (UDP) | Communication with Charon-SSP Director |
| Graphics emulation | default: 11001 (TCP) | Mouse event data (default port can be changed; must be unique on host system) |
| | default: 11000 (TCP) | Keyboard event data (default port can be changed; must be unique on host system) |
| Telnet or TCP raw mode serial ports | default: 9000 (TCP) | Port to access emulated serial console or other emulated serial port via TCP. Port must be unique for each emulated port on host system. |
| Xephyr X-server | 6001-6100 (TCP); port specified in X11 server configuration | Determines the X DISPLAY number. For example: 6100 indicates DISPLAY :100. Must be unique on host system. |
| | 7100 (TCP) | Font-server port |
| | 177 (TCP and UDP) | XDMCP server |
| NFS server | 111 (TCP and UDP) | RPC portmapper |
| | ports assigned via portmapper by default | use # rpcinfo -p to determine ports used |
| | static port assignments | For example: setting RPCMOUNTDOPTS="-p port" in /etc/sysconfig/nfs will add "-p port" to the rpc.mountd command. |
| VNC server on host system | 5901-5910 (TCP) | Actual port depends on VNC server configuration. Allow a remote client to access the VNC server on the host system. |
| HASP license manager, license server | 1947 (TCP and UDP) | Access to web-based Sentinel ACC GUI, identification of remote network licenses served by license servers, using remote network licenses. |
| HASP license client | 30000 to 65535 (UDP) | Incoming answers from license servers if broadcast search is used. |
| VE license server | Default: 8083 and 8084 (TCP) | Port 8083 allows clients to access the license on the license server. Port 8084 allows access to an informational web GUI. Ports are configurable. |
| | 443 (TCP) and 902 (TCP, UDP) | Required between VE license server system and ESXi or vCenter Server when running the VE license server in a VMware environment. |
| SSH/SCP/SFTP | 22 (TCP) | Secure login and file transfer |
| PulseAudio server | 4713 (TCP) | Emulated audio device |
| iSCSI target | 3260 (TCP and UDP) | Required for the initiator to access the target. |

4.5.2 Linux Firewall and Virtual Bridges

If firewall rules are to be used for bridged traffic, the kernel can be instructed to apply iptables (also arptables and ip6tables) rules to bridged traffic.

In older versions, this option was included in the bridge functionality itself. Starting with kernel 3.18, the filtering functionality in the form of the `br_netfilter` module was moved into a separate module that can be loaded by the user if required. **If the module is not loaded, no firewall rules are applied to bridge traffic and no further actions are required to pass the bridged traffic through the Linux host system.**

To check if the module is loaded use the command

```
# lsmod | grep netfilter
```

To use the firewall for bridged traffic on newer Linux kernels, the module must be loaded using the command

```
# modprobe br_netfilter
```

or by defining an `iptables` rule that uses the `physdev` module

After the module has been loaded, the system configuration parameters

```
net.bridge.bridge-nf-call-iptables
bridge-nf-call-arptables
bridge-nf-call-ip6tables
```

become available. They are **set to 1 by default** (equivalent to `echo 1 > /proc/sys/net/bridge/bridge-nf-call-iptables`). This value enables iptables rules for bridged traffic.

Setting the parameters to 0 will disable the firewall rules. They can be set permanently via `/etc/sysctl.conf`.

To allow bridged traffic through the enabled firewall, use commands like the following:

```
# firewall-cmd --permanent --direct --add-rule ipv4 filter INPUT 1 \
-m physdev --physdev-is-bridged -j ACCEPT
# firewall-cmd --permanent --direct --add-rule ipv6 filter INPUT 1 \
-m physdev --physdev-is-bridged -j ACCEPT
# firewall-cmd --reload
```

Please refer to the documentation of your host system for more detailed information.

Please note: at the time of writing, this feature is not yet available for `nftables`.

5 Charon-SSP Software Installation

This chapter describes the following installation activities:

- **New installation** of the RPM packages of the conventional Charon-SSP product and the VE-enabled RPM packages on Linux ([Charon-SSP RPM Installation](#)).
- **New installation** of Charon-SSP Baremetal ([Charon-SSP Baremetal Installation](#)).
- Additional prerequisites, and recommended or necessary post-installation tasks.

Please note: This guide covers configuration and management of all Charon variants. However, it does not cover the initial installation of cloud-specific marketplace images (of type AL or VE). Please refer to the cloud-specific Getting Started guides for this information (see the Charon-SSP section on the [Stromasys documentation page](#)).

If you need to upgrade the Charon-SSP software, please refer also to [Charon-SSP Software Upgrade](#).

Should you have received Charon Manager and Charon Director kits for Microsoft Windows, please refer to [Charon-SSP GUI for Microsoft Windows](#) in the appendix.

5.1 General License Information

You need a **product license** to run Charon-SSP emulators. **Upgrading** to this Charon-SSP version from an older version requires a **license update**. If you do not have an appropriate product license yet, please contact your Stromasys representative or your Stromasys VAR.

If you have a **HASP software license, please note:** starting with Charon-SSP 3.0.x, Charon-SSP contains new Sentinel license runtime versions (aksusbd package). The new versions contain important updates. However **older software licenses** (created under runtime version 2.5.1) are not compatible with the new version. Upgrading to new versions of the runtime software in most cases requires the installation of a new software license. Downgrading to an older license runtime version from version 7.63 or later can also cause the invalidation of a software license. Contact your VAR or Stromasys representative to discuss the best way to upgrade.

5.1.1 Initial License Installation Overview

This section provides an overview of basic steps. Please refer to the [Charon-SSP License Management](#) chapter for more information about managing licenses. If you use packages for use with a VE license server (package name includes the string “ve”), please also refer to the **VE license server guide** in the [Licensing Documentation](#) section on the Stromasys documentation page.

5.1.1.1 Sentinel/Gemalto Licenses

After the installation of the Sentinel Runtime Software and the Charon-SSP packages, you can install the Charon-SSP license on the system.

- If you purchased a **hardware license**, you can simply plug the dongle into a free USB port on the system (starting with Charon-SSP 3.0.x, HL-MAX dongles are now also supported).
- If you purchased a **software license**, you must create a fingerprint file in C2V (customer-to-vendor) format containing the system characteristics. Use this file to request a license for your system from Stromasys. Then install the V2C (vendor-to-customer) file that you will receive from Stromasys on your system.
- If you have an **existing license that needs to be updated**, you must create a customer-to-vendor (C2V) file and use this file to request a license update from Stromasys. Then install the V2C (vendor-to-customer) file(s) that you will receive from Stromasys on your system.
- If your license is a **network license served by a license server**, make sure your client system has access to the license server.

The tools to manage **Sentinel HASP licenses** are included in the Charon-SSP installation kits and documented in this user's guide.

5.1.1.2 Charon-SSP VE (Virtual Environment) Licenses

If you use a Charon-SSP VE license server and the matching Charon-SSP emulator packages, the following steps are required:

- Configure the license server address and (optionally) the passphrase in the license section of the emulator configuration.
- If there is no appropriate license on the VE license server:
 - Create a C2V (customer-to-vendor) file on the VE license server. Use this file to request a license from Stromasys.
 - The license received in response (V2C file) must be installed on the VE license server.

Currently, the VE license server is available in supported cloud and VMware environments, and on physical servers. Please refer to the VE license server guide in the [Licensing Documentation section](#) on the Stromasys documentation page for information about how to request and install a license on a VE license server.

5.2 Charon-SSP RPM Installation

5.2.1 Installation Packages Overview

5.2.1.1 Components of a Charon-SSP Installation

Charon-SSP includes the following parts. Unless otherwise mentioned, they are covered in this manual.

Licensing components (only **one of them** is applicable to each specific product variant):

- **aksusbd package** – Sentinel runtime environment required for using HASP licensing in on-premises installations. The installation of the package is covered in this manual. For more information about HASP license management, please refer to [Charon-SSP License Management](#).
- **VE (Virtual Environment) license server**: this license server is typically installed in supported cloud and VMware environments and serves licenses to VE-enabled Charon-SSP emulator instances. **The license server is not part of the Charon-SSP installation kit.** It must be installed separately. Installation and management of the VE license server are covered in the VE license server guide in the [Licensing Documentation section](#) on the Stromasys documentation page. A basic overview is also provided in [Charon-SSP License Management](#). Please refer to the release notes of the VE license server and Charon-SSP for information regarding relevant protocol changes. In some cases, an upgrade to the VE license server may be required to avoid protocol mismatches between license client and server.
- **Stromasys-operated cloud-specific license server** for automatic licensing of Charon-SSP AL images. The installation of such images is covered in separate cloud-specific manuals (please refer to the [Charon-SSP documentation page](#) for details).

Management GUI components:

- **Charon-SSP Manager** – Graphical tool to configure and manage Charon-SSP (local and remote). The Charon-SSP Manager can manage a maximum of 100 emulated SPARC instances on one host system. This number is the *theoretical* maximum number of instances on one host system. The real maximum number is determined by the resources of the host system compared with the resources required by each of the emulated SPARC instances, and the available licenses.
- **Charon-SSP Director** – Graphical tool to manage distributed host systems running multiple emulator instances. The Charon-SSP Director can manage up to 512 host systems.

The Charon Agent:

- **Charon-SSP Agent** – Bridge between the Charon-SSP Manager and the Charon-SSP privileged functions; also enables the Charon-SSP Director to discover Charon-SSP hosts automatically. Handles the automatic start of Charon-SSP virtual machines at system boot (if configured via the Charon Manager).

The Charon-SSP emulator software:

- **Charon-SSP/4U(+)**: 64-bit SPARC V9 Sun-4u architecture
- **Charon-SSP/4V(+)**: 64-bit SPARC V9 Sun-4v architecture
- **Charon-SSP/4M**: 32-bit SPARC V8 Sun-4m architecture

The Open-Source license texts:

- Provided via the **charon-ssp-oss-license** package

A **complete installation** consists of the following:

- One licensing option
- The management GUI
- The Charon Agent
- One or more emulator packages
- Applicable open-source licenses are installed via a separate package (charon-ssp-oss-license-[version].rpm) and stored in **/opt/charon-ssp/oss-license**.¹

The individual components described above are also available in several different **prepackaged installation images**:

- **Charon-SSP Baremetal distribution** – Appliance-type kit provided as an ISO file. It includes everything needed to run a non-cloud-based (on-premises) Charon-SSP host – apart from the product license. The system and the Charon-SSP application are mostly managed using a comprehensive GUI. However, the user can access the underlying host operating system if required.
- **Charon-SSP cloud-specific AL images** and **Charon-SSP cloud-specific VE images** – Available on selected cloud marketplaces. Their installation is described in separate cloud-specific *Getting Started Guides*.

Charon-SSP/4U+ and Charon-SSP/4V+ must run on physical hardware supporting Intel VT-x/EPT or AMD-v/NPT – with a Linux kernel supported by Charon-SSP/4U+/4V+ (see [Host System Requirements](#)).

5.2.1.2 Charon-SSP Installation Packages

The **full set of Charon-SSP packages** is available in RPM format for the supported host operating system distributions and versions (see [Host System Software Requirements](#)).

Charon-SSP Manager and Charon-SSP Director are also available as Ubuntu and Microsoft Windows packages.

The focus of this section is on the RPM package installation. The installation of the Charon Manager and Charon Director for Microsoft Windows are described in [the Microsoft Windows appendix](#).

Obtaining the required packages:

- Stromasys provides a download location for the kits. For official releases, this location **usually contains a ZIP file** with all the required packages for a specific installation situation (e.g., on-premises or cloud). If needed, the individual packages are also accessible in the download location. Typically, they are organized in a directory hierarchy similar to the following:
 - **rpm.common**: contains the packages that are platform independent (e.g., Charon Manager and Charon Agent)
 - **on-prem**: contains the standard RPM packages for Linux 7 and 8, the HASP license drivers, and the Baremetal ISO.
 - **cloud**: contains upgrade packages for cloud-specific AL marketplace images and VE enabled emulator packages—both for Linux 7 and 8.
 - **windows**: contains the Charon Manager and the Charon Director kits for Microsoft Windows.
- More information about the individual packages is provided in the tables below.
- For bootable USB devices (Baremetal product), see [Creating Bootable USB media from Baremetal ISO files](#).
- The Charon Manager packages for all supported platforms are also included in the Charon Agent package. After installing this package, they are in **/opt/charon-agent/ssp-agent/bin/** and can be copied from there to be installed on any supported Linux or Microsoft Windows system.
- Please contact either Stromasys or your Value-Added Reseller (VAR) if you have questions about obtaining the packages.

¹ This product includes third-party code licensed to Stromasys for use and redistribution under open-source licenses. Notwithstanding any of the terms and conditions of your license agreement with Stromasys, the terms of certain open-source licenses may be applicable to your use of this product. Applicable open-source licenses are installed via a separate package (charon-ssp-oss-license-[version].rpm) and stored in **/opt/charon-ssp/oss-license**. This list of open-source licenses may be updated from time to time and is subject to inaccuracies as a result of being compiled with reference to third-party software incorporated into the services as of the date the list was generated. All open-source licensing information is provided "as-is." Stromasys and its suppliers make no representations or warranties, express or implied, with regard to this list or its accuracy or completeness, or with respect to any results to be obtained from use or distribution of the list. By using or distributing this list, you agree that in no event shall Stromasys be liable for any damages whatsoever resulting from any use or distribution of this list, including, without limitation, any special, consequential, incidental, or other direct or indirect damage.

The following tables show the names of the relevant RPM installation packages. In these tables, the placeholders have the following meaning:

- **{version}**
Package version. For example: 5.3.10 for Charon-SSP 5.3.10.
- **{architecture}**
The 4m, 4u (+), or 4v (+) emulated SPARC architectures. The 4U+ and 4V+ packages include 4U and 4V. The 4U+ and 4V+ images in these packages are only supported on real hardware with host operating system kernels supported for Charon-SSP/4U+/4V+.
- **{cloud-id}**
Cloud platform for which the specific Automatic Licensing marketplace image was released (for example **oci** and **aws**). These RPM packages are only available on preinstalled cloud-specific Automatic Licensing images, or they may be provided by Stromasys to update such installations as needed.

Packages **common** to version 7.x and 8.x of the supported Linux distributions:

| Component | Charon-SSP software RPM package names |
|---|---------------------------------------|
| Charon-SSP Manager | charon-manager-ssp-{version}.rpm |
| Charon-SSP Director | charon-director-ssp-{version}.rpm |
| Charon-SSP Agent | charon-agent-ssp-{version}-x86_64.rpm |
| Sentinel runtime environment (for Sentinel HASP licenses) | aksusbd-{version}.x86_64.rpm |
| Open-source software license documents | charon-ssp-oss-license-{version}.rpm |

Packages **specific** to version 7.x of the supported Linux distributions:

| Component | Charon-SSP software RPM package names |
|---|--|
| The emulator software itself: Charon-SSP 4M, 4U, 4U+, 4V, and 4V+ | <u>Emulator packages to be used with Sentinel HASP licenses:</u> charon-ssp-{architecture}-{version}.el7-x86_64.rpm |
| | <u>Emulator packages to be used with a VE license server:</u> charon-ssp-{architecture}-{version}.ve.el7-x86_64.rpm |
| | <u>Emulator packages used to update cloud-specific AL installations:</u> charon-ssp-{architecture}-{version}.{cloud-id}.market.el7-x86_64.rpm |

Packages **specific** to version 8.x of the supported Linux distributions:

| Component | Charon-SSP software RPM package names |
|---|--|
| The emulator software itself: Charon-SSP 4M, 4U, 4U+, 4V, and 4V+ | <u>Emulator packages to be used with Sentinel HASP licenses:</u> charon-ssp-{architecture}-{version}.el8-x86_64.rpm |
| | <u>Emulator packages to be used with a VE license server:</u> charon-ssp-{architecture}-{version}.ve.el8-x86_64.rpm |
| | <u>Emulator packages used to update cloud-specific AL installations:</u> charon-ssp-{architecture}-{version}.{cloud-id}.market.el8-x86_64.rpm |

Ubuntu installation packages:

| Product | Charon-SSP software package names (DEB format for Ubuntu) |
|---------------------|---|
| Charon-SSP Manager | charon-manager-ssp- <i>{version}</i> .deb |
| Charon-SSP Director | charon-director-ssp- <i>{version}</i> .deb |

The installation of these packages and the installation of the Baremetal distribution are discussed in the sections below. The installation of the cloud-specific marketplace images is described in separate user's guides.

Approximate RPM package sizes after installation:

- Charon-SSP Agent: 47 MB (since the Charon-SSP Agent hierarchy is the default location for emulator configuration and log files, the disk space may increase significantly during operation).
- Charon-SSP Manager: 6.5 MB
- Charon-SSP Director: 0.3 MB
- Charon-SSP/4M: 8 MB
- Charon-SSP/4U/4U+: 24/57 MB
- Charon-SSP/4V/4V+: 24/57 MB

Please note:

- The versions of the Charon-SSP packages installed on one system must match. The Charon Manager will show a version mismatch with the agent on the target system and offer an automatic installation of the matching version. The Charon Agent does not check if the emulator packages installed have the same version as the agent. However, it can lead to unpredictable behavior of the product if there is a version mismatch.
- **Exception:** It is possible to have several versions of the Charon-SSP Manager on a system to manage remote systems with versions different from the local system.
- If you want to retain an old version of the Charon-SSP Manager on Linux, copy the content of the directory **/opt/charon-manager** to another directory, e.g., **/opt/charon-manager-<version>** before upgrading. Later, you can use this version to manage remote systems with matching versions. On Windows, older versions are automatically retained when a new version is installed.

Should you have received Charon Manager and Charon Director kits for Microsoft Windows, please refer to [Charon-SSP GUI for Microsoft Windows](#) in the appendix.

5.2.2 Additional Installation Considerations and Prerequisites

In addition to the hardware and operating system prerequisites for the Charon host system, there are additional prerequisites and considerations specific to the individual packages that make up the Charon-SSP product. They are described in this section.

5.2.2.1 General Information

Charon-SSP Agent:

- If you downgrade to an old version (1.4.x or lower), the storage format of the agent password is not compatible with the one used in newer versions. This will cause login-failures. To prevent this, delete the file **/opt/charon-agent/ssp-agent/etc/passwd.conf** before installing the old agent version or before starting the agent (this will re-create the file with default password **stromasys**).
- During the upgrade to 4.0.x or higher, the directory **/opt/charon-agent/ssp-agent/ssp/script** containing the default *init.d* scripts for starting Charon instances at host boot is deleted. The scripts in **/etc/init.d** remain, but are no longer used. Automatic instance start during host system boot is now handled completely by the Charon Agent.

5.2.2.2 License Considerations

The Charon-SSP/4M/4U (+)/4V (+) packages provide the core emulator software. This software requires a license. This means the following:

- For Charon-SSP emulators requiring a **Sentinel HASP license**, the Sentinel HASP software (**aksusbd** package) must be installed on the system (see installation section below).
- For Charon-SSP emulators capable to work with a **VE license server**, an appropriate license server must be available on the same or on a different system. In a VMWare environment, if the license server is not on the same VM as the emulator, it must be on a VM on the same ESXi host (for a license server binding to the ESXi host), or on a VM running on an ESXi host managed by the same vCenter server (for a license server binding to the vCenter server). Please refer to the [VE License Server Guide](#) for more information.
- An appropriate license must be obtained from Stomasys and installed on the Charon host or a license server.

Instances based on Charon-SSP AL (Automatic Licensing) cloud marketplace images are licensed automatically at launch, either using a Stomasys-operated cloud-specific license server or a customer-operated AutoVE server.

Please note: different from previous versions, the Sentinel HASP runtime software (**aksusbd** package) provided starting with Charon-SSP 4.2.5 no longer requires the 32-bit glibc-library to be installed.

5.2.2.3 Considerations for Charon-SSP Hosts without Graphics HW

Charon-SSP contains several features (in particular, graphics device emulation and audio emulation) that require operating system packages typically installed as part of the Linux graphical subsystem. The same applies to the Charon-SSP Manager and the Server JIT feature.

These operating system packages are pre-installed on Charon-SSP Baremetal systems. They are also often already installed on Linux installations with graphics hardware installed and used. However, they are often not installed on a Linux server installation – especially if the server itself has no graphics hardware.

To facilitate easy installation, the Charon-SSP emulator packages in version 4.2.x allow the installation without enforcing the installation of graphics and audio operating system packages required for graphics and audio emulation and the Server JIT feature. This means that some dependencies are not automatically included from the operating system repositories during Charon installation and must be manually installed if the corresponding features are required.

The following table provides an overview of the required packages:

| RPM Package | Graphics and audio emulation | Charon-SSP Manager * | Server JIT feature |
|---|------------------------------|----------------------|--------------------|
| libX11 | X | X | |
| xorg-x11-server-utils | X | X | |
| alsa-plugins-pulseaudio | X | | |
| gtk2 | | X | |
| xorg-x11-xauth (only required for X11-Forwarding) | | X | |
| libc (version 50 for Linux 7.x, version 60 for Linux 8.x) | | | X |

* If you install the Charon Manager with the **yum** or **dnf** command, the packages required by the Manager (except for xorg-x11-xauth) and any dependencies that these packages themselves may have, are resolved automatically if a package repository is available.

If you suspect problems caused by missing packages and the emulator was started via the Charon Manager, **check the emulator crash-log file in addition to the emulator log file**. If starting the emulator from the command-line, review the command-line output.

The packages above have their own dependencies. Install the above packages with the **yum** or the **dnf** command in order to have their dependencies automatically installed. If your server does not have access to the standard operating system repositories, refer to [this document](#) for instructions on setting up a local repositories.

Note:

- The exact list of additionally required packages depends on what is already installed on the server.
- Future versions of Charon-SSP may handle this situation differently and enforce the dependency installation during the installation of the Charon-SSP components.

5.2.2.4 Special Prerequisites for the Charon-SSP Director

The Charon-SSP Director provides a GUI-based management for several distributed host systems each running one or more Charon-SSP emulated SPARC systems. This chapter shows the installation of the Charon-SSP Director.

The Charon-SSP Director requires the Charon-SSP Manager to be installed on the same system.

5.2.2.5 Special Prerequisites for the Charon-SSP Agent

The Charon-SSP Agent provides the connection between the Charon-SSP Manager and the privileged parts of the Charon-SSP emulator software. It also enables the Charon-SSP Director to find Charon-SSP hosts automatically. Starting with version 4.0.x it is also responsible for automatically starting emulator instances at system boot if this configuration option has been selected via the Charon Manager.

Special prerequisites for the Charon-SSP Agent:

- The Charon-SSP Agent must be installed on the same system on which the Charon-SSP emulator software—that is, Charon-SSP/4M/4U (+)/4V (+)—is installed.
- The Charon-SSP Agent uses the **lspci** command for certain *non-critical* functions. This command is part of the **pciutils** package. The package is preinstalled on all prepackaged Charon-SSP distributions (cloud images and Baremetal), but it may not be preinstalled on the individual Linux distribution and version used by the customer. In this case, it is recommended to install the package using the command:
yum install pciutils (version 7.x of the supported Linux distributions)
dnf install pciutils (version 8.x of the supported Linux distributions)

5.2.2.6 Special Prerequisites for Additional Utilities

Depending on the Charon-SSP product, the Charon Manager supports the setup of additional, optional utilities.

The required packages are pre-installed on Baremetal systems. On RPM-based installations, additional packages may be needed for certain Charon Manager utilities if any of these utilities are to be used.

| Utility in Charon Manager | Prerequisite packages |
|--|---|
| Network Settings features (host interface configuration, virtual bridges, VLAN interfaces) | <p>Linux versions 8.x: The <i>NetworkManager</i> is required for the Charon Manager host network management functionality. Normally, this package is pre-installed even on basic server installations.</p> <p>Linux versions 7.x: The <i>bridge-utils</i> package is required for virtual bridge management (see post-installation tasks)</p> |
| X11 Server configuration | <i>Xephyr</i> and the <i>ifconfig</i> command (see post-installation tasks) |
| SSH VPN tunnel example | Remote tunnel endpoint: <i>autossh</i> (see post-installation tasks) |
| iSCSI | <i>iscsi-initiator-utils</i> |
| NFS server | <i>nfs-utils</i> |
| VNC server | Required software (<i>tigervnc-server-minimal</i> , <i>tigervnc-license</i> , <i>tigervnc-server</i>). The VNC server must be configured manually on non-Baremetal systems. The remote system requires a VNC client package (e.g., <i>vinagre</i>). |

5.2.3 Installation Commands on Supported Host Operating Systems

The following tables provides an overview of the installation commands for the supported host operating systems. For details, please refer to the relevant man-pages on Linux. The table only lists command-line installation options for Linux. There are also graphical installation tools. To describe all of them is outside the scope of this document.

| | RPM package installation |
|---|--|
| Package manager (uses repositories, takes care of dependencies, etc.) | <p>Linux 7.x of the supported Linux distributions:</p> <pre># yum install [<package-name> <path-to-package>]</pre> <p>Linux 8.x of the supported Linux distributions:</p> <pre># dnf install [<package-name> <path-to-package>]</pre> <ul style="list-style-type: none"> • If the path to an RPM file is provided, yum and dnf use a local package installation. If only the package name is provided, yum and dnf will try to use the configured repositories. • The dnf command was introduced in versions 8.x of the supported Linux distributions. Its syntax is very similar to yum – and the yum command can still be used. |
| Command to install individual local packages | <pre># rpm -i <path-to-package></pre> |

| | DEB package installation (Ubuntu) |
|--|---|
| Command to install individual local packages | <pre># dpkg -i <package-name></pre> |

All installation steps on Linux must be performed from a privileged account as denoted by the '#' prompt.

5.2.4 Installing the Charon-SSP Packages

This section describes the installation of the different Charon-SSP packages that make up the overall Charon-SSP product. While it mentions the commands to install Debian-format packages on Ubuntu, this chapter concentrates on the installation of the RPM based product set on the relevant host operating systems.

You can install the Charon-SSP packages individually or in one step. The below steps show how to install the Sentinel license drivers first and then the Charon-SSP packages in one step.

| Description of step | Command |
|--|--|
| Copy the installation packages to the Charon host system. | The commands used depend on the setup of the host system. Often SFTP or SCP are used. |
| Log into the Charon host system as the root user. | |
| Go to the directory containing the installation packages | <code># cd <directory-containing-packages></code> |
| Install the Sentinel HASP Runtime . This step is not required if you use a VE License Server (see the <i>VE License Server Guide</i> for details). | Versions 7.x of the supported Linux distributions: <code># yum install aksusbd-7.103-1.x86_64.rpm</code> Versions 8.x of the supported Linux distributions: <code># dnf install aksusbd-7.103-1.x86_64.rpm</code> |
| Install the Charon-SSP packages . The packages can be installed individually or in one step. | Versions 7.x of the supported Linux distributions: <code># yum install charon*.rpm</code> Versions 8.x of the supported Linux distributions: <code># dnf install charon*.rpm</code> |

5.2.4.1 Installation Example

Installing the license driver package:

```
# yum install aksusbd-7.103-1.x86_64.rpm
Loaded plugins: fastestmirror
Examining aksusbd-7.103-1.x86_64.rpm: aksusbd-7.103-1.x86_64
Marking aksusbd-7.103-1.x86_64.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package aksusbd.x86_64 0:7.103-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch           Version         Repository      Size
=====
Installing:
  aksusbd         x86_64        7.103-1        /aksusbd-7.103-1.x86_64 13 M
```

Transaction Summary

```
=====
Install 1 Package
```

```
Total size: 13 M
```

```
Installed size: 13 M
```

```
Is this ok [y/d/N]: y
```

```
Downloading packages:
```

```
Running transaction check
```

```
Running transaction test
```

```
Transaction test succeeded
```

```
Running transaction
```

```
  Installing   : aksusbd-7.103-1.x86_64                      1/1
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/aksusbd.service to
/etc/systemd/system/aksusbd.service.
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/hasplmd.service to
/etc/systemd/system/hasplmd.service.
```

```
  Verifying    : aksusbd-7.103-1.x86_64                      1/1
```

```
Installed:
```

```
  aksusbd.x86_64 0:7.103-1
```

```
Complete!
```

Installing the Charon-SSP Agent package:

```
# dnf install charon-agent-ssp-5.4.2-x86_64.rpm
```

```
Last metadata expiration check: 0:07:16 ago on Di 10 Mai 2022 08:59:11 EDT.
```

```
Dependencies resolved.
```

```
=====
Package                Architecture Version      Repository      Size
=====
Installing:
charon-agent-ssp       x86_64      5.4.2-1     @commandline    42 M
```

Transaction Summary

```
=====
Install 1 Package
```

```
Total size: 42 M
```

```
Installed size: 46 M
```

```
Is this ok [y/N]: y
```

```
Downloading Packages:
```

```
Running transaction check
```

```
Transaction check succeeded.
```

```
Running transaction test
```

```
Transaction test succeeded.
```

```
Running transaction
```

```
  Preparing    :                                             1/1
```

```
  Running scriptlet: charon-agent-ssp-5.4.2-1.x86_64         1/1
```

```
  Installing    : charon-agent-ssp-5.4.2-1.x86_64         1/1
```

```
  Running scriptlet: charon-agent-ssp-5.4.2-1.x86_64         1/1
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/ssp-agentd.service →
/etc/systemd/system/ssp-agentd.service.
```

```

Verifying      : charon-agent-ssp-5.4.2-1.x86_64          1/1
Installed:
  charon-agent-ssp-5.4.2-1.x86_64
Complete!

```

Installing the Charon-SSP emulator packages:

```

# dnf install charon-ssp-4?-*.*.rpm
Last metadata expiration check: 0:05:42 ago on Di 10 Mai 2022 08:59:11 EDT.
Dependencies resolved.
=====
Package           Architecture Version           Repository         Size
=====
Installing:
charon-ssp-4m     x86_64           5.4.2.el8-1      @commandline      2.6 M
charon-ssp-4u     x86_64           5.4.2.el8-1      @commandline      7.5 M
charon-ssp-4v     x86_64           5.4.2.el8-1      @commandline      7.6 M

Transaction Summary
=====
Install 3 Packages

Total size: 18 M
Installed size: 58 M
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                               1/1
  Running scriptlet: charon-ssp-4v-5.4.2.el8-1.x86_64 1/3
  Installing      : charon-ssp-4v-5.4.2.el8-1.x86_64 1/3
  Running scriptlet: charon-ssp-4v-5.4.2.el8-1.x86_64 1/3
  Running scriptlet: charon-ssp-4u-5.4.2.el8-1.x86_64 2/3
  Installing      : charon-ssp-4u-5.4.2.el8-1.x86_64 2/3
  Running scriptlet: charon-ssp-4u-5.4.2.el8-1.x86_64 2/3
  Installing      : charon-ssp-4m-5.4.2.el8-1.x86_64 3/3
  Running scriptlet: charon-ssp-4m-5.4.2.el8-1.x86_64 3/3
  Verifying      : charon-ssp-4m-5.4.2.el8-1.x86_64 1/3
  Verifying      : charon-ssp-4u-5.4.2.el8-1.x86_64 2/3
  Verifying      : charon-ssp-4v-5.4.2.el8-1.x86_64 3/3

Installed:
  charon-ssp-4m-5.4.2.el8-1.x86_64      charon-ssp-4u-5.4.2.el8-1.x86_64
  charon-ssp-4v-5.4.2.el8-1.x86_64
Complete!

```

Installing the Charon-SSP Manager and Director packages including the automatic installation of dependencies:

```
# dnf install charon-manager*.rpm
Last metadata expiration check: 0:08:06 ago on Di 10 Mai 2022 08:59:11 EDT.
Dependencies resolved.
=====
Package                Arch      Version      Repository      Size
=====
Installing:
charon-manager-ssp     x86_64    5.4.2-1      @commandline    2.4 M
Installing dependencies:
atk                    x86_64    2.28.1-1.e18 appstream        270 k
<lines removed>

Transaction Summary
=====
Install 43 Packages

Total size: 12 M
Total download size: 9.2 M
Installed size: 38 M
Is this ok [y/N]: y
Downloading Packages:

<lines removed>

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction

<lines removed>

Installed:
  atk-2.28.1-1.e18.x86_64
  cairo-1.15.12-3.e18.x86_64
  charon-manager-ssp-5.4.2-1.x86_64
<lines removed>

Complete!

# dnf install charon-director-ssp-5.4.2.rpm
Last metadata expiration check: 0:31:40 ago on Di 10 Mai 2022 08:59:11 EDT.
Dependencies resolved.
=====
Package                Architecture Version      Repository      Size
=====
Installing:
charon-director-ssp     x86_64     5.4.2-1      @commandline    113 k

Transaction Summary
=====
Install 1 Package
```

```
Total size: 113 k
Installed size: 287 k
Is this ok [y/N]: y
Downloading Packages:
```

<lines removed>

```
Installed:
  charon-director-ssp-5.4.2-1.x86_64
```

```
Complete!
```

Installing the open-source software license package (the text documents are stored under /opt/charon-ssp/oss-license):

```
# dnf install charon-ssp-oss-license-5.4.2.rpm
Last metadata expiration check: 0:06:33 ago on Di 10 Mai 2022 08:59:11 EDT.
Dependencies resolved.
=====
Package                Arch      Version      Repository      Size
=====
Installing:
  charon-ssp-oss-license  x86_64    5.4.2-1      @commandline    21 k

Transaction Summary
=====
Install 1 Package

Total size: 21 k
Installed size: 95 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                1/1
  Installing     : charon-ssp-oss-license-5.4.2-1.x86_64 1/1
  Running scriptlet: charon-ssp-oss-license-5.4.2-1.x86_64 1/1
  Verifying      : charon-ssp-oss-license-5.4.2-1.x86_64 1/1

Installed:
  charon-ssp-oss-license-5.4.2-1.x86_64

Complete!
```

5.2.5 Post-Installation Steps

5.2.5.1 Post-Installation Tasks for Version 8.x of Supported Linux Distributions

Please note: starting with Charon-SSP 5.2.7, Charon-SSP/4U+/4V+ will set this parameter automatically.

For Charon-SSP/4U+/4V+ running on Linux version 8.x, a kernel parameter must be configured that will be set when the system boots. This will avoid unexpected problems during Charon-SSP operation.

To set this parameter, perform the following steps as the **root user**:

- Open the file **/etc/default/grub** in a text editor.
- Add the following parameter to the definition of **GRUB_CMDLINE_LINUX**.
transparent_hugepage=never
- Backup the current **grub.cfg** file. Depending on whether the system boot mode is legacy BIOS or EFI, this file is in **/boot/grub2/** or in **/boot/efi/EFI/<linux-distribution>/**.
- Create a new grub.conf with the additional parameter:
grub2-mkconfig > /tmp/grub.cfg
- Copy or move the new grub.cfg file to the correct location (see above).
- Reboot.
- Check if the parameter was set correctly:
cat /proc/cmdline

5.2.5.2 Sentinel HASP Post-Installation Tasks

The following post-installation tasks are highly recommended to improve security and trouble-shooting options.

5.2.5.2.1 Setting the Password for the Sentinel HASP GUI

| Step | Description |
|------|---|
| 1 | <p>Open a web browser and navigate to http://localhost:1947.</p> <p>If the server is not GUI based, you can access the interface from a remote host by replacing localhost with the name of the server running Charon-SSP. Prerequisite: remote access to the Sentinel ACC GUI must be enabled by editing <i>/etc/hasplm/hasplm.ini</i> and changing the value of the parameter ACCremote from 0 (access disabled) to 1 (access enabled)). A template file can be found in <i>/opt/charon-agent/ssp-agent/etc/hasplm.ini</i>.</p> |
| 2 | <p>Click on Configuration in the left-hand menu pane.</p> |
| 3 | <p>Click on the Basic Settings tab.</p> |
| 4 | <p>Under Password Protection, click the Change Password button.</p> |
| 5 | <p>At the Change Password window:</p> <ul style="list-style-type: none"> • Leave the Current Admin Password field blank (there is no password set by default). • Enter the desired password into the New Admin Password field. • Repeat the desired password in the Re-enter new Admin Password field. • Click the Submit button. <p>With these settings, you may be prompted for a username and password when you connect to the HASP GUI from a remote system. Just enter the password and leave the username field empty.</p> |
| 6 | <p>Back at the Basic Settings tab:</p> <ul style="list-style-type: none"> • Under the section Password Protection, select the All ACC Pages radio button. • Click the Submit button to save this change. |
| 7 | <p>If desired, you can allow remote access to the Sentinel HASP GUI:</p> <ul style="list-style-type: none"> • Go to the Basic Settings tab. • Select the Allow Remote Access to ACC check box. • Click the Submit button. <p>These steps have the same effect as editing <i>hasplm.ini</i> the way described in step 1.</p> |

If you permit remote access to the Sentinel software via port 1947, **make sure to allow only authorized systems** to access this port by employing appropriate firewall configurations.

5.2.5.2.2 Set more Restrictive File Protections on the *hasplm.ini* File

To prevent unprivileged access to the Sentinel HASP configuration file on Linux, enter the following commands:

```
# chmod 0700 /etc/hasplm
# chmod 0600 /etc/hasplm/*
```

5.2.5.2.3 Enable access and error logging

To enable access and error logging for the license driver, see [Enabling logging in Sentinel Admin Control Center](#). This is strongly recommended in order to facilitate trouble-shooting.

5.2.5.3 Charon-SSP Post-Installation Tasks

The following post-installation tasks are recommended to improve the usability of the software:

- Add the Charon-SSP for Linux Software to the PATH variable of the shell.
- When running versions 7.x of the supported Linux distributions, install the *bridge-utils* package if you plan to use the virtual network feature of the Charon Manager.
- Install *autossh* on the remote tunnel endpoint in the customer network if you plan to use the SSH VPN tunnel to connect customer network and Charon host (as described later in this document).

5.2.5.3.1 Setting the PATH variable

The following table shows the steps to append the Charon-SSP paths to the PATH variable of the shell:

| Shell | Step | Description |
|--|------|---|
| C Shell | 1 | Open the file <code>.login</code> in your home directory or a file for systemwide settings (e.g., <code>/etc/profile.d/charon-ssp.csh</code>) with a text editor. |
| | 2 | Add the line <pre>setenv PATH \$PATH:/opt/charon-ssp/ssp-4m:/opt/charon-ssp/ssp-4u: /opt/charon-ssp/ssp-4v</pre> to the end of the file. |
| Bourne Shell (e.g., <i>bash</i> or <i>sh</i>) | 1 | Open the appropriate file in your home directory (<code>.profile</code> , <code>.bash_profile</code> , or <code>.bashrc</code>) or a file for systemwide settings (e.g., <code>/etc/profile.d/charon-ssp.sh</code>) with a text editor. |
| | 2 | Add the lines <pre>PATH=\$PATH:/opt/charon-ssp/ssp-4m:/opt/charon-ssp/ssp-4u: /opt/charon-ssp/ssp-4v export PATH</pre> to the end of the file. |

5.2.5.3.2 Installing the bridge-utils and autossh Packages

Please note: the *bridge-utils* package installation is only **applicable if running versions 7.x** of the supported Linux distributions. Linux 8.x uses **nmcli** to set up virtual bridges.

If you plan to use the virtual network configuration of Charon-SSP Manager on Linux 7.x, you must install the *bridge-utils* package on Linux if it is not already installed.

The following table lists the **installation steps for the *bridge-utils* package:**

| Description | Command |
|--|---------------------------------------|
| Install <i>bridge-utils</i> (if not already installed; dependencies are installed automatically). Note that Red Hat needs to be registered to have access to the repositories. | <pre># yum install bridge-utils</pre> |

Please note: it is possible to configure a virtual bridge without the **bridge-utils** package, for example, by using the appropriate **ip** commands. However, the *bridge-utils* package is required for the configuration via the Charon-Manager functions.

To configure the SSH VPN tunnel following the example in [SSH VPN – Connecting Charon Host and Guest to Customer Network](#), you also need *autossh* on the remote Linux endpoint of the tunnel.

The following table lists the **installation steps for the *autossh* package** (if it has not already been installed):

| Description | Command |
|--|---|
| Install the EPEL repository Note that Red Hat needs to be registered to have access to the repositories. Up-to-date information can be found on: https://fedoraproject.org/wiki/EPEL | Older Red Hat / CentOS 7 versions: <pre># yum --enablerepo=extras install epel-release</pre> Current Linux versions : <pre># yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm (EPEL for Linux version 7.x)</pre> or <pre># dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm (EPEL for Linux version 8.x)</pre> Recommended on Red Hat 7.x (as EPEL packages may depend on the extras repositories): <pre># subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-extras-rpms" --enable "rhel-ha-for-rhel-*-server-rpms"</pre> |
| Install the <i>autossh</i> package. | <pre># yum install autossh</pre> |

5.2.5.4 Charon-SSP Manager Post-Installation Tasks on Linux

The following post-installation tasks are highly recommended to improve usability and security of the product:

- Installing the **Xephyr** X-Server and the **ifconfig** command (if not already installed).
- Creating a Charon-SSP Manager menu Item
- Upon first login with the Charon Manager, you will be asked to change the default management password. If you need to change the password later, please refer to [Modifying the Charon-SSP Agent Preferences](#).

5.2.5.4.1 Installing the Xephyr X-Server

To use an X-Server on Linux in conjunction with the Charon-SSP Manager (described in [Graphical Interface via X11 Server for Linux](#)), it is necessary to install the Xephyr X-server.

Use the following commands to install the relevant Xephyr package (the example uses the **yum** command of Linux 7.x; on Linux 8.x, you can use the **dnf** command):

| Linux System | Installation commands |
|---|--|
| RPM-based installation on the supported Linux distributions | <pre># yum install xorg-x11-server-Xephyr</pre> |
| Ubuntu | <pre># apt-get update # apt-get install xserver-xephyr</pre> |

Use the following commands to install the package containing the **ifconfig** command:

| Linux System | Installation commands |
|---|---|
| RPM-based installation on the supported Linux distributions | <pre># yum install net-tools</pre> |
| Ubuntu | <pre># apt-get update # apt-get install net-tools</pre> |

5.2.5.4.2 Creating a Desktop Menu Item for Charon-SSP Manager

Use the following steps to create a desktop menu item for Charon-SSP Manager (if required).

Different distributions and different desktop environments have different ways to add desktop launchers and menu items. This section describes some commonly used approaches, but there may be differences in how this is handled in your desktop environment.

There are two steps to be performed:

1. Create an application shortcut (launcher).
2. Add the launcher to the desktop menu.

Create an application shortcut:

Many desktop environments allow a user to create a new desktop shortcut by right-clicking on the desktop. However, newer Linux distributions, especially the ones using GNOME 3 as the basis for their desktop environments, no longer have this option. They also do not always have a menu visible on the desktop by default.

On such systems, you can create the application shortcut manually as described below, or use one of the optional menu administration packages, such as **alacarte**, which provides a GUI to create such shortcuts.

Bear in mind that alacarte creates the required **.desktop** file as a *user-specific* file (e.g., **\$HOME/.local/share/applications/alacarte-made.desktop**).

If you need a *systemwide* .desktop file, use the manual method for creating a shortcut as described below.

Creating an application shortcut using **alacarte**:

Start **alacarte** from the command-line and select **New Item**. This opens the following window where you can enter the information for the Charon-SSP Manager as shown below.

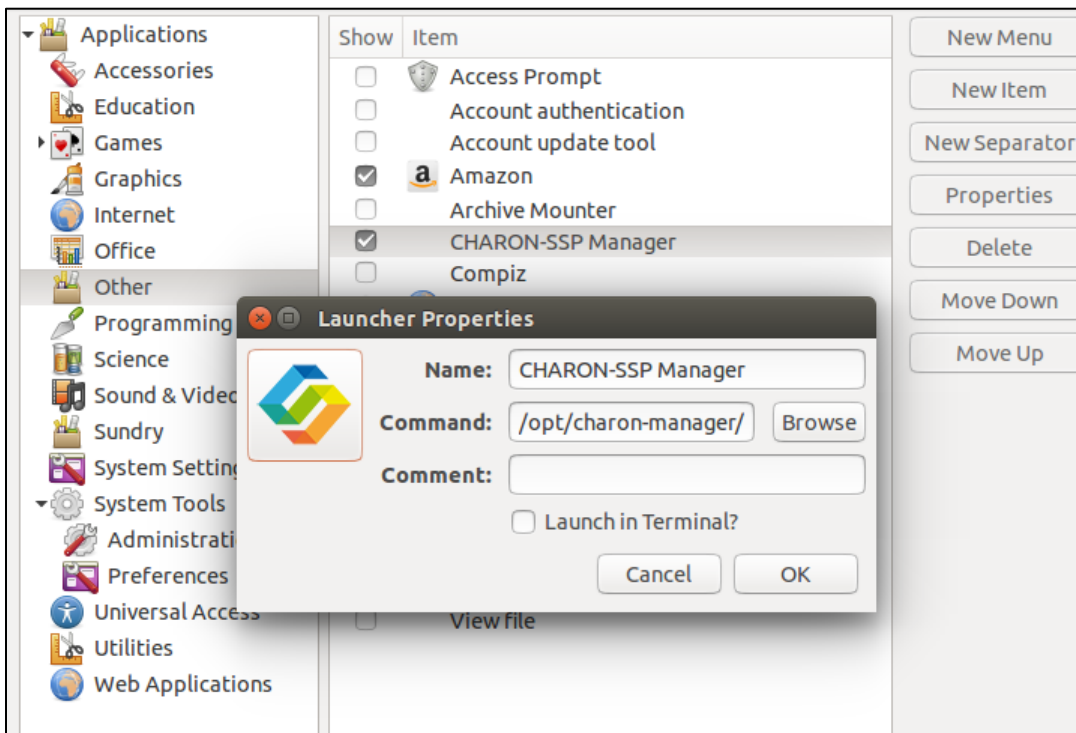


Figure 3: Using alacarte to add menu items

The data needed (path of executable and icon name) is listed in the manual example below. Clicking on **OK** creates a Charon-SSP Manager item in the category **Other** in this example. You can select any other category to create your shortcut. To select an icon, click on the “picture frame” on the left of the **Properties** window.

Note that **alacarte** does not store the category in the **.desktop** file it creates. Instead, it integrates the information in the desktop menu application configuration, e.g., *gnome-application-menu*. If your desktop menu system requires these settings, you must edit the created file. In such cases, it may be faster to use the manual method described below.

Creating an application shortcut manually:

| Step | Description |
|------|---|
| 1 | Using the root account, create the file /usr/local/share/applications/charon-ssp-manager.desktop . |
| 2 | Add the following text to the file created in step 1 and save it. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre>[Desktop Entry] Version=<add appropriate version here> Name=Charon-SSP Manager Exec=/opt/charon-manager/ssp-manager/ssp-manager Icon=/opt/charon-manager/ssp-manager/resource/charon.png Terminal=false Type=Application StartupNotify=true Categories=System;</pre> </div> |
| 3 | Set the file protections and ownership appropriately: <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre># chmod 0644 /usr/local/share/applications/charon-ssp-manager.desktop # chown root:root /usr/local/share/applications/charon-ssp-manager.desktop</pre> </div> |

Add the application shortcut to the desktop menu:

If your desktop environment provides a desktop menu, you will find the Charon-SSP Manager entry in the category you configured.

An example is shown below:



Figure 4: Sample desktop menu item

On Systems that do not provide such a menu on the desktop, there are other ways to access the applications. For example, on plain GNOME 3 systems click on the link **Activities** in the upper left corner, search for the Charon-SSP Manager, and add it to the favorites by right-clicking on it.

5.2.5.5 Charon-SSP Director Post-Installation Tasks on Linux

The following post-installation task is not necessary, but recommended to improve usability of the product:

- Create a Charon-SSP Director Menu Item

5.2.5.5.1 Creating a Desktop Menu Item for Charon-SSP Director

The steps required for creating a desktop icon are the same as described for the Charon-SSP Manager (section [Creating a Desktop Menu Item for Charon-SSP Manager](#)).

Differences with respect to the required parameters:

- The path of the executable is **`/opt/charon-director/ssp-director/ssp-director`**.
- The path of the icon is **`/opt/charon-director/ssp-director/resource/director.png`**.

5.3 Charon-SSP Baremetal Installation

5.3.1 General Information about the Baremetal Distribution

The Charon-SSP Baremetal distribution is an ISO file (**charon-baremetal-ssp-*<version>*.iso**) containing the Charon-SSP virtual machine software and the underlying host operating system with additional utilities. This software is distributed either as an ISO image or on read-only optical media (for bootable USB devices see section [Creating Bootable USB media from Baremetal ISO files](#)). If you do not have a copy of the Charon-SSP Baremetal distribution, please contact either Stromasys or your Value-Added Reseller for further help.

Charon-SSP Baremetal can run on dedicated hardware or in a virtual machine. See [Host System Requirements](#).

Important general information regarding the Baremetal installation:

The Charon-SSP Baremetal distribution contains Charon-SSP/4M, Charon-SSP/4U (+), and Charon-SSP/4V (+). Charon-SSP/4U+ and Charon-SSP/4V+ utilize special hardware functionality to deliver improved performance. Due to the hardware requirements, they can only be used on real hardware providing the required CPU features. **Running Charon-SSP/4U+ or Charon-SSP/4V+ in a VM is not supported.** When configuring an emulated SPARC system, the Charon-SSP Manager provides an option to select whether the host system runs on hardware supported by Charon-SSP/4U+/4V+ or not.

Important additional information regarding Baremetal upgrades:

- The Charon-SSP Baremetal installation procedure described in this section, is also used for **major upgrades** (or downgrades) to the system. This happens automatically **if the existing system disk is selected**.
- **If a system upgrade instead of an installation is performed**, the host system files and host applications as well as the Charon-SSP Baremetal-specific applications are upgraded while important Charon-SSP configuration data and container files are preserved. However, note the following:
 - Starting with version 4.1.29, a new Baremetal installation creates a **root (“/”) partition with a size of 15GiB**. Baremetal systems that were originally installed from earlier versions have a root partition size of **5GiB**. The partition size is not changed during an upgrade. Hence, the user must carefully evaluate if the available space on the root partition is still sufficient. Starting with version 5.3.11, Charon-SSP Baremetal is based on Rocky Linux 8.5. The change from Linux 7 to Linux 8 caused an increase in the space required on the root partition (at least 4.1GiB) leaving very little free space on a 5GiB root partition. **In such cases, a new installation should be considered if possible (after backing up all relevant Charon and user data)**.
 - If performing a system upgrade, the host system must be shut down to boot from the installation CD. **Make sure to cleanly shutdown any guest systems running in Charon-SSP instances**, and to power off the instances before shutting down the host system.
 - After shutting down any running guest system and stopping the emulator, **and before performing an upgrade, back up all your configuration data, container files, and other customer-specific data**. In the case of a downgrade, bear in mind that an older software version may not understand all parameters in a configuration created with the new software version.
 - Any **customer-specific configurations, applications, and additional Linux packages** added to the original Baremetal installation are lost during an upgrade using the ISO image (comparable to re-installing a normal Linux system). Hence, such additions must be added again. Also, the post-installation tasks must be reviewed and, if necessary, repeated.
 - If upgrading from a Charon-SSP version before 5.3.11, the underlying operating system will be upgraded **from Centos 7 to Rocky Linux 8**. This major version change will cause some significant system management changes. For example, the network configuration and user configurations may be reset.
 - At the time of writing, desktop settings made for the login screen (e.g., keyboard and display settings) via the Settings app are lost during an upgrade using the ISO image. The default settings are restored (including the US keyboard layout).

5.3.2 Creating Bootable USB media from Baremetal ISO files

Be very careful to use the correct device name for the USB device—otherwise you can severely damage your installed system!

A Charon-SSP Baremetal ISO file can be used to create a bootable USB device, for example, a USB stick.

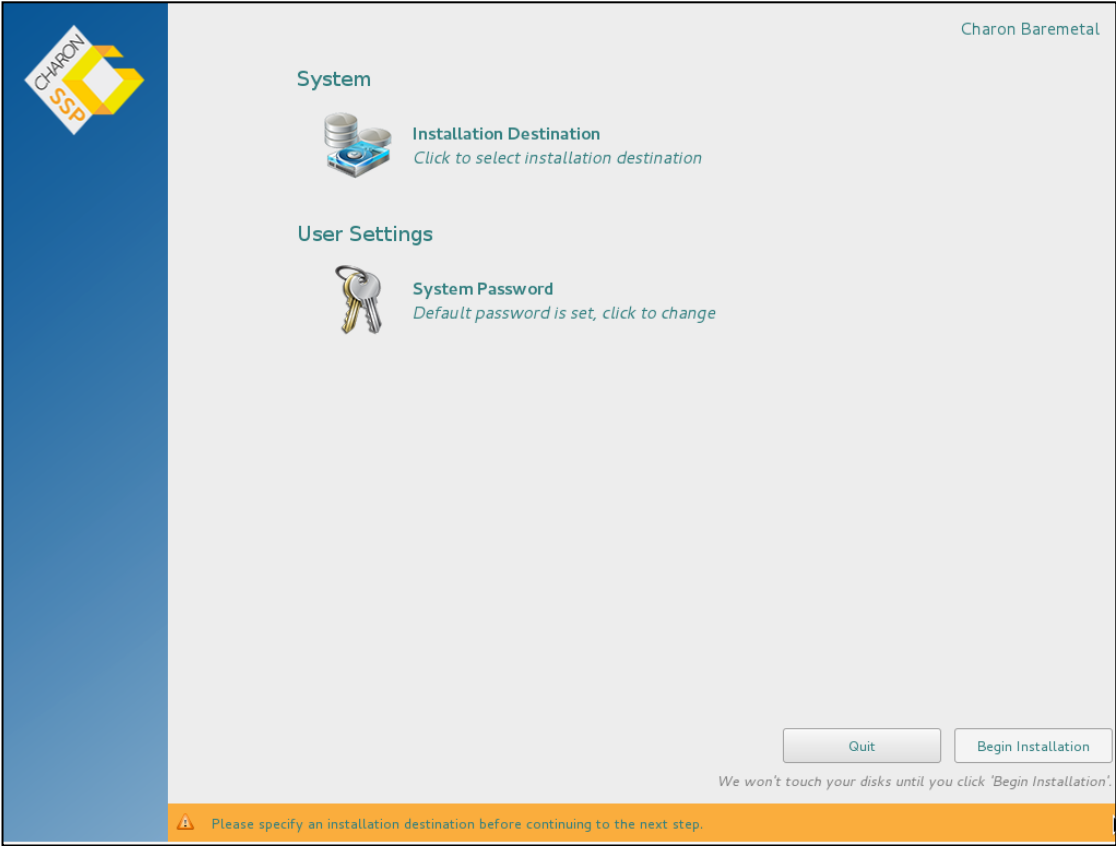
- To create a bootable stick that supports both BIOS and UEFI booting for Baremetal distributions, use the free tool [Rufus](#) on Microsoft Windows.
- A bootable USB device can also be created using the **gnome-disks** utility on Linux, or the following Linux command (*X* stands for the device letter of the USB device, for example, sdc or sdh):

```
# dd if=<charon-baremetal-name>.iso of=/dev/sdX bs=4M
```

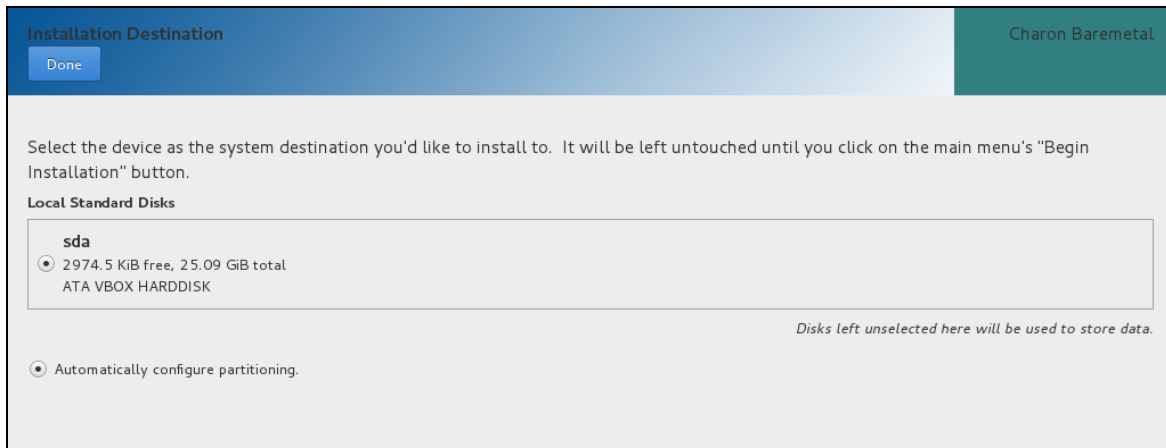
Please note: this method reliably creates USB devices bootable using the legacy BIOS method, but the devices may not always be bootable via UEFI (especially for older BIOS versions). If this problem occurs, use Rufus as described above.

5.3.3 Basic Installation Steps for the Baremetal Distribution

The following table shows the basic installation steps for the Charon-SSP Baremetal distribution:

| Step | Description |
|------|---|
| 1 | <p>If the host system is a virtual machine (not supported for Charon-SSP/4U+/4V+):</p> <ul style="list-style-type: none"> • Configure the virtual machine for a Linux x86-64 environment. • Use the requirements specified in the section Hardware Requirements to configure the virtual hardware with sufficient capacity for all Charon-SSP instances that are to run on it. • Attach the Charon-SSP Baremetal distribution ISO or physical installation medium to the VM. • Power up the VM and boot from the installation medium. <p>If the host system is a physical machine:</p> <ul style="list-style-type: none"> • Verify that your system meets the hardware requirements. • Load the Charon-SSP installation medium. • Boot the system from the installation medium. |
| 2 | <p>Upon successful boot, you should see the following summary screen:</p> <p>This screen indicates on the orange bar at the bottom of the window that the installation destination (storage) has not yet been configured. In addition, the default management password (stromasys) set by the installation can be changed to a user-defined value.</p>  <ol style="list-style-type: none"> 1. Click on the Installation Destination icon to enter the storage configuration. This step is mandatory. 2. Optionally, click on the System Password icon to set a new management password (default = stromasys). This password is used to access all system-created, password-protected, user-visible functions and accounts of the Charon-SSP Baremetal system. After the installation, you can change it via the Charon-SSP Manager. If the password is not changed here, it must be set at first login of the Charon Manager. Note that the installation uses a US keyboard layout. <p>To return from any configuration screen to this summary screen, press Done in the upper left corner of the respective configuration screen.</p> |

3 The installation options for the **destination storage** are shown on the following screen:



Select the disk(s) you want to use as the system device(s). The disk will be partitioned automatically by Charon-SSP.

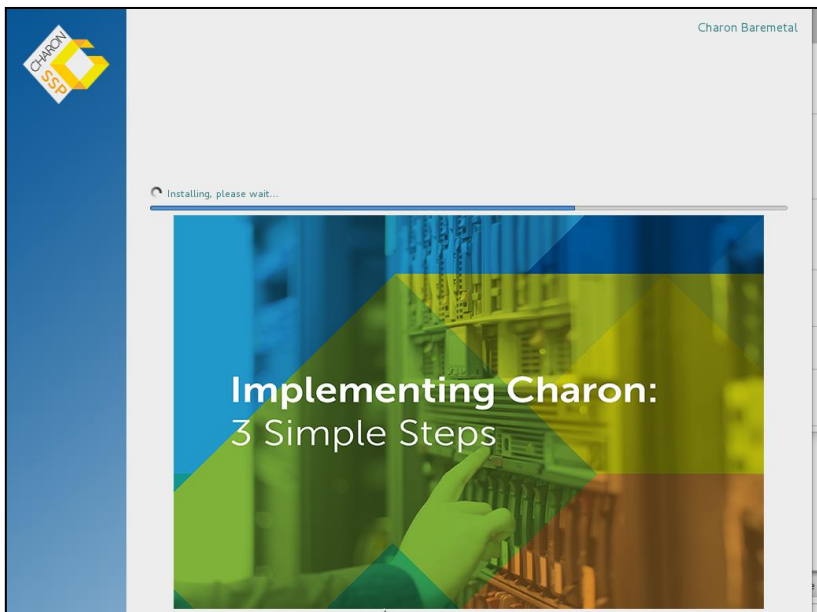
Please note:

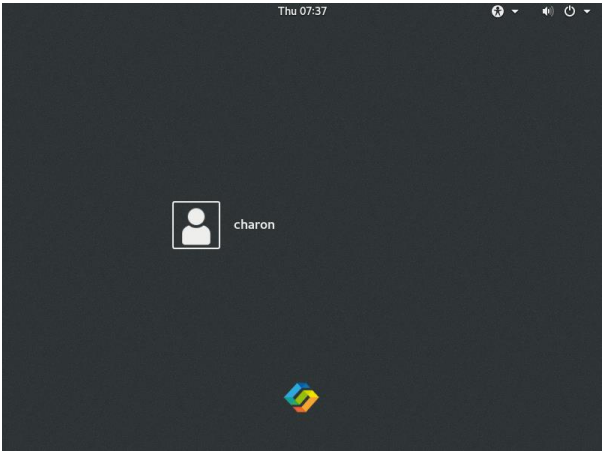

- If you use the **current system disk** of an **existing Baremetal system**, an upgrade (or downgrade) procedure will be performed instead of a new installation.
- Any disks not selected as system disks will not be changed by the installation. They can later be added as data disks using the Charon-SSP Baremetal storage manager.

Press **Done** in the upper left corner to finish your disk selection and get back to the **Installation Summary** page.

4 Once you have completed the storage configuration and, optionally, the password configuration, press **Begin installation** on the **Installation Summary** page.

This leads to a series of installation progress pages displaying a progress bar and some text about the progress of the installation.



| | | | | | | | | | | | | | | | | |
|---------------|---|--|-----|-------------------------------------|-------------|-------|---------|---------|------------|---|---------------|--------|----------|-------------------------|----------|------------------------------|
| <p>5</p> | <p>Upon completion, the system will automatically reboot. The installation media is ejected automatically to prevent booting the Charon-SSP Baremetal installer again.</p> <p>After the reboot, the Charon-SSP login screen will be displayed (use the password set during the installation or the default password <i>stromasys</i>).</p> <p>Remember that the default keyboard layout is the US layout. You can change it after logging in.</p> |  <p>The image shows a dark-themed login screen for Charon-SSP. At the top right, it displays 'Thu 07:37'. In the center, there is a user icon labeled 'charon'. At the bottom right, there is a colorful logo consisting of three interlocking shapes in blue, green, and yellow.</p> | | | | | | | | | | | | | | |
| <p>6</p> | <p>Following a successful login, the Charon-SSP Baremetal overview screen will be displayed:</p> <p>The installation of the Charon-SSP Baremetal distribution is now complete.</p> |  <p>The image shows the 'Welcome to Charon Baremetal SSP' overview screen. It features a sidebar on the left with icons for 'Home', 'Charon', and 'Toolbox'. The main content area is titled 'System Information' and lists the following details:</p> <table border="1"> <tr> <td>OS:</td> <td>Charon Baremetal SSP v4.3.4 build-1</td> </tr> <tr> <td>Charon-SSP:</td> <td>4.3.4</td> </tr> <tr> <td>Memory:</td> <td>3789 MB</td> </tr> <tr> <td>Processor:</td> <td>Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz x 2</td> </tr> <tr> <td>User Storage:</td> <td>8.5 GB</td> </tr> <tr> <td>Host IP:</td> <td>10.0.0.1, 192.168.2.106</td> </tr> <tr> <td>Up Time:</td> <td>Tue Jan 12 17:18:19 UTC 2021</td> </tr> </table> <p>Below the system information is a 'Quick Action' section with two buttons: 'Charon Manager' (with the colorful logo) and 'Terminal' (with a terminal icon). The bottom of the screen shows a system tray with a volume icon and the time '12:22'.</p> | OS: | Charon Baremetal SSP v4.3.4 build-1 | Charon-SSP: | 4.3.4 | Memory: | 3789 MB | Processor: | Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz x 2 | User Storage: | 8.5 GB | Host IP: | 10.0.0.1, 192.168.2.106 | Up Time: | Tue Jan 12 17:18:19 UTC 2021 |
| OS: | Charon Baremetal SSP v4.3.4 build-1 | | | | | | | | | | | | | | | |
| Charon-SSP: | 4.3.4 | | | | | | | | | | | | | | | |
| Memory: | 3789 MB | | | | | | | | | | | | | | | |
| Processor: | Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz x 2 | | | | | | | | | | | | | | | |
| User Storage: | 8.5 GB | | | | | | | | | | | | | | | |
| Host IP: | 10.0.0.1, 192.168.2.106 | | | | | | | | | | | | | | | |
| Up Time: | Tue Jan 12 17:18:19 UTC 2021 | | | | | | | | | | | | | | | |

5.3.4 Charon-SSP Baremetal Post-Installation Tasks

5.3.4.1 Baremetal Post-Installation Tasks Overview

Please note: in case of an upgrade using the Baremetal ISO file, the post-installation tasks must be reviewed and, if required, repeated.

The following post-installation tasks are required or highly recommended for improved security and usability:

- Unless you changed the management password during the installation, you will be asked to change the management password upon first connecting to the Charon-SSP host system with the Charon Manager. See [Using the Charon-SSP Manager](#).
- Customizing the user environment.
- Enrolling the Stromasys public key for UEFI Secure Boot
- Set a password for the Sentinel HASP web interface (see [Sentinel HASP Post-Installation Tasks](#))

5.3.4.2 Customizing the User Environment

The following items can be configured to adapt the user environment:

- Keyboard settings: the behavior and layout of the keyboard can be adapted to the user's preferences.
- Display settings: the settings can be adapted to the characteristics of the host system.

Both configuration options are accessible from the Charon-SSP **Toolbox** (see [Toolbox Screen Functions](#)).


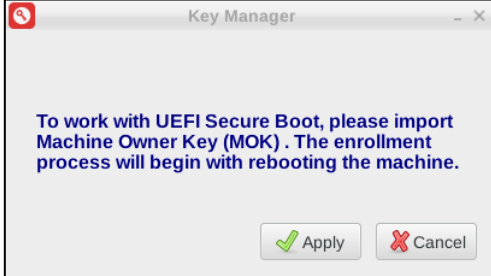
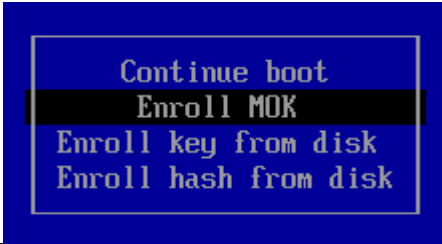
5.3.4.3 Enrolling the Stromasys Public Key for UEFI Secure Boot

Charon-SSP Baremetal and the previously available Charon-SSP Barebone include Linux kernel modules for PCI pass-through (Charon-SSP/4U and 4U+) and accelerated CPU virtualization using VT-x/EPT or AMD-v/NPT (Charon-SSP/4U+/4V+). These kernel modules are loaded on demand when the Charon-SSP instance starts.

Since Charon-SSP version 2, the Baremetal and Barebone products have supported UEFI boot (in addition to the traditional BIOS boot). If UEFI secure boot is used, every kernel module that is to be loaded into the kernel must be digitally signed. Otherwise, it cannot be loaded successfully.

Charon-SSP Baremetal provides a graphical interface to simplify the key enrollment process.

Perform the following steps to import the Stromasys public key:

| Step | Description |
|------|--|
| 1 | Open the Toolbox tab in the Charon-SSP Baremetal GUI. Click on the Key Manager icon to open the key manager.  |
| 2 | If UEFI Secure Boot is enabled on the system, you will see a message like the one displayed in the screenshot. If UEFI Secure Boot is not enabled, you will receive a message that the import function is not available. To initiate the key import, click on Apply .  |
| 3 | You will be prompted for the required password. Enter the Charon-SSP Management password and click on OK . The key manager will inform you that a reboot is required to complete the procedure. |
| 4 | Reboot the system. |
| 5 | The blue screen <i>Shim UEFI key management</i> will be displayed. Press any key to start the MOK (machine owner keys) management function. If you do not press a key before the timeout expires, the boot process will continue. Important: if this screen times out and normal boot continues, the key will not be enrolled and you must start with step 1 again to import the key. |
| 6 | After pressing a key, a small menu will be displayed. Select Enroll MOK from the menu. Confirm this and the following selections with the Enter key.  |
| 7 | The next screen offers the option to view the key before continuing with the enrollment. Select Continue to carry on with the next step. |
| 8 | Confirm that you want to enroll the key by selecting Yes on the next screen. |
| 9 | You will be prompted for a password (use the password you entered in the key manager window). |
| 10 | Confirm that the system should be rebooted by selecting OK . |
| 11 | After the reboot, you can verify the success of the operation by opening a terminal window and entering the command: <code># mokutil --list-enrolled</code> |

The enrollment of the key is persistent across re-installations of the Charon-SSP host operating system.

To enroll the key on a **Barebone** system use the following steps:

- As the root user, execute the import_key script:
`# /opt/charon-ssp/[ssp-4u | ssp-4v]/import-key/import_key.sh`
- Follow the instructions above starting with **Step 4**.

6 Charon Host System Management Overview

The underlying Linux host operating system of a Charon-SSP installation requires a certain amount of system management as any Linux system used to run important services for a business. Depending on the type of Charon-SSP installation, the approaches to system management differ. They are described below.

6.1 Host Management for Conventional and Cloud Hosts

6.1.1 General System Management Information

For RPM-based installations (cloud and on-premises) of Charon-SSP and Charon-SSP cloud-specific images (type AL and VE), the Linux host system is managed with the tools provided by the host operating system or by third-party vendors.

The cloud-specific Charon-SSP AL image provides some additional support via the Charon Manager, e.g., for storage and file management.

General Linux system management knowledge is required to perform such tasks. Please refer to the documentation of your host operating systems for details.

The pre-packaged cloud images are configured by default to support the Charon-SSP kernel modules required for Charon-SSP/4U+/4V+ (if the Charon host runs on a physical/baremetal instance in the cloud). Updating to a kernel not provided by Stromasys will invalidate this support. Other restrictions may also apply. If you are unsure if a planned change to the system will cause problems with the operation of Charon-SSP/4U+/4V+, please contact your Stromasys partner or Stromasys support.

6.1.2 User Accounts in Charon-SSP Cloud Marketplace Images

During the launch of a Charon-SSP cloud (AL or VE) system, three user accounts are created.

The **charon** user:

This account is used by Charon-SSP cloud installations mainly for SFTP access for file transfer to and from the Charon host system (under /charon/storage). It can also be used for setting up the integrated Charon manager SSH tunnel when running the Charon Manager on a remote system.

Direct interactive login via the network is not possible. The **password** for this account is the general management password set at first login of the Charon Manager (default before being set: **stromasys**). However, for most of the activities will take place using the key-pair installed during instance launch.

The **sshuser** user:

This is the primary account for remote interactive login and for setting up the SSH VPN tunnel. It has access to the root account via the **sudo** command. The **password** for this account is the general management password set at first login of the Charon Manager (default before being set: **stromasys**). However, for most of the activities (especially remote login) will take place using the key-pair installed during instance launch.

The **root** user:

Privileged administrator user. Accessible via **sudo** from the **sshuser** account.

6.2 Host Management Charon-SSP Baremetal

6.2.1 General System Management Information

Please note the following points with respect to Linux shell access:

- Charon-SSP Baremetal allows user access to the underlying host operating system. If this option is used to manage the host system, general Linux system management knowledge is required. Please refer to the documentation of your host operating system for details.
- The Baremetal system is configured by default to support the Charon-SSP kernel modules required for Charon-SSP/4U+/4V+ and the PCI pass-through features. Updating to a kernel not provided by Stromasys will invalidate this support. Other restrictions may also apply. If you are unsure if a planned change of the system will cause problems with the operation of Charon-SSP, please contact your Stromasys partner or Stromasys support.

The **primary system management interface** of a Charon-SSP Baremetal system is the customized Charon Baremetal GUI. This section provides a brief introduction to this GUI.

6.2.2 Charon-SSP Baremetal User Accounts

The management password for the user accounts and the Charon Agent can be set during installation or via the Charon-SSP Manager as described in [Modifying the Charon-SSP Agent Preferences](#). In case of a forgotten management password, please refer to the section [Resetting a Forgotten Management Password](#).

During the installation of a Charon-SSP Baremetal system, three user accounts are created.

The **charon** user:

This is the main user-visible account used by Charon-SSP Baremetal. It provides among other things

- the GUI that provides interactive access to the system,
- the terminal window that allows access to the host operating system shell (**sudo** access to root possible),
- remote connection to the system via VNC,
- integrated Charon manager SSH tunnel when running the Charon Manager on a remote system, and
- SFTP access for file transfer to and from the Charon Baremetal host system.

Direct interactive login via the network is not possible. The **password** for this account is the general management password set during installation or at first login of the Charon Manager (default before being set: **stromasys**). After installing a public key via the Charon Manager or manually, connecting with the associated private SSH key is also possible.

The **sshuser** user:

This is the primary account for remote interactive login and for setting up the SSH VPN tunnel. It has access to the root account via the **su** command. The **password** for this account is the general management password set during installation or at first login of the Charon Manager (default before being set: **stromasys**).

After installing a public key via the Charon Manager or manually, connecting with the associated private SSH key is also possible.

The **root** user:

Privileged administrator user. The **password** for this account is the general management password set during installation or at first login of the Charon Manager (default before being set: **stromasys**). SSH key-based login can be configured manually.

6.2.3 Charon-SSP Baremetal User Interface

The Charon-SSP Baremetal GUI-based user interface provides comprehensive management tools for all required tasks. This section describes the interface.

6.2.3.1 Shutdown, Reboot, Screen Lock

Once logged in as the user **charon**, there is no log out function to avoid unintentional closing of open applications. Instead, the console can be protected by locking the screen. This and other functions are provided via the symbolic power button at the bottom left of the screen, as illustrated below:

Clicking on the power button symbol in the bottom left corner, opens a small panel with the following options:

- Lock the screen
- Restart the system (green symbol)
- Shut down the system (red symbol)
- **Cancel** the operation

Please note: before a reboot or shutdown all running guest-systems must be cleanly shut down and the emulator should be stopped. Normally, the **Restart** and the **Shutdown** icons are inactive if there is a running emulator on the system.

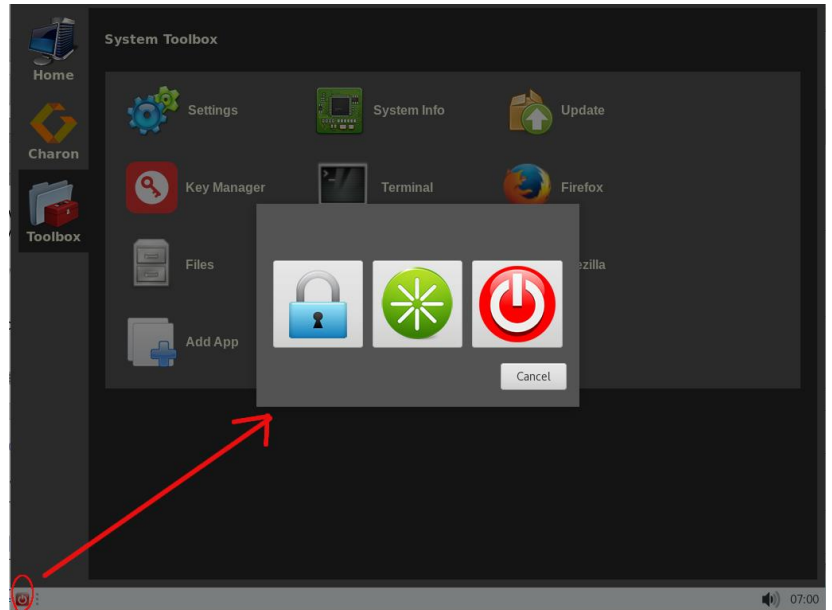


Figure 5: Baremetal power button symbol

6.2.3.2 Home Screen Functions

After the installation or a reboot, the **Home** screen is normally the first screen displayed by the Charon-SSP Baremetal system (unless Kiosk mode is enabled as described later in this document). The following image shows an example:



Figure 6: Charon-SSP Baremetal home screen

This screen provides a quick overview of the Charon-SSP host system in **System Information**. It shows

- Operating system version
- Charon-SSP package version
- Host system RAM and CPU configuration
- Disk space
- IP addresses of the host system
- System uptime
- The **Quick Action** providing access to frequently used activities
- The red **Power** button at the bottom left of the screen allows you to lock, shut down, or reboot the system

The left-hand bar shows the other available tabs. These are described below.

6.2.3.3 Charon Screen Functions

The **Charon** screen provides access to the Charon-SSP specific management tools as shown in the example below:

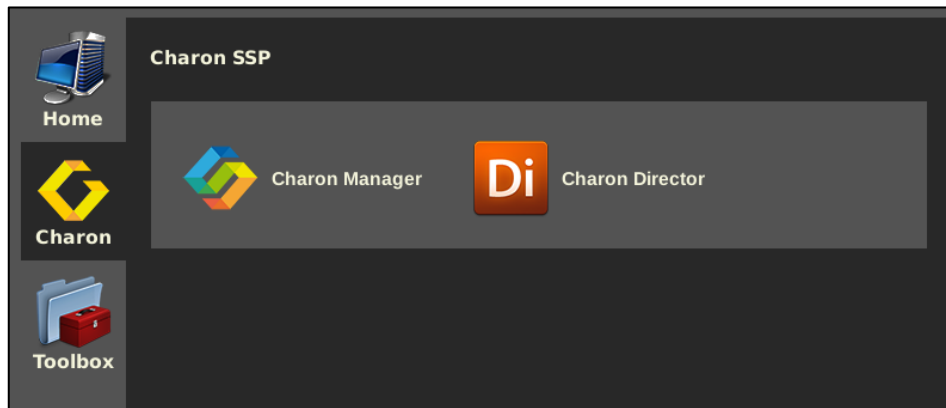


Figure 7: Charon-SSP Charon screen

On this screen, you can start the **Charon-SSP Manager** and the **Charon-SSP Director**. These are the central management tools for the Charon-SSP product. Their use is described in detail in the chapter [Configuring and Using the Charon-SSP Software](#).

6.2.3.4 Toolbox Screen Functions

The toolbox screen provides system management functions relevant to the host system. A sample is provided by the image below:

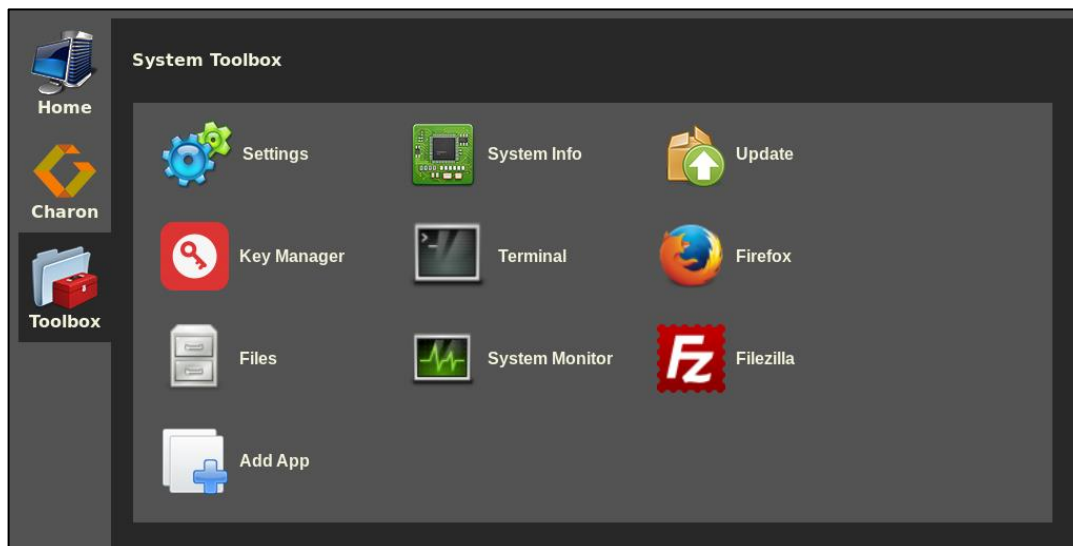


Figure 8: Charon-SSP Baremetal System Toolbox

The functions available in the **System Toolbox** are described below.

6.2.3.4.1 Host System Settings

Clicking on the **Settings** menu opens the Gnome settings window. It allows the user to manage many settings. Three of them are mentioned here as examples.

Language and keyboard settings:

The language and keyboard settings can be changed under **Region & Language**. The language settings do not apply to the Charon-specific GUI – only to the standard Linux components. However, the required keyboard can be set under **Input Sources**: select the required layout and delete the default layout. To enable the local keyboard selection on the **login screen**, click on **Login Screen** and set as appropriate.

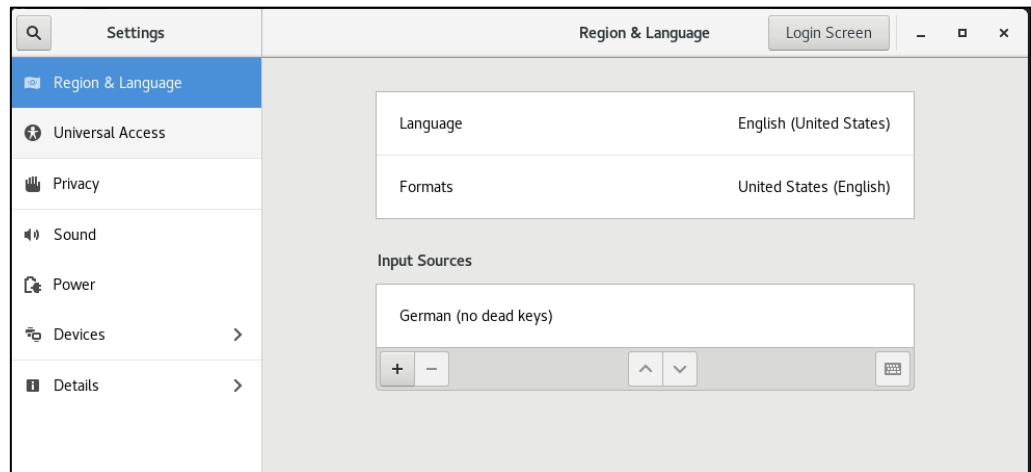


Figure 9: Baremetal local keyboard settings

Screen resolution:

The screen resolution can be set under **Devices > Displays**.

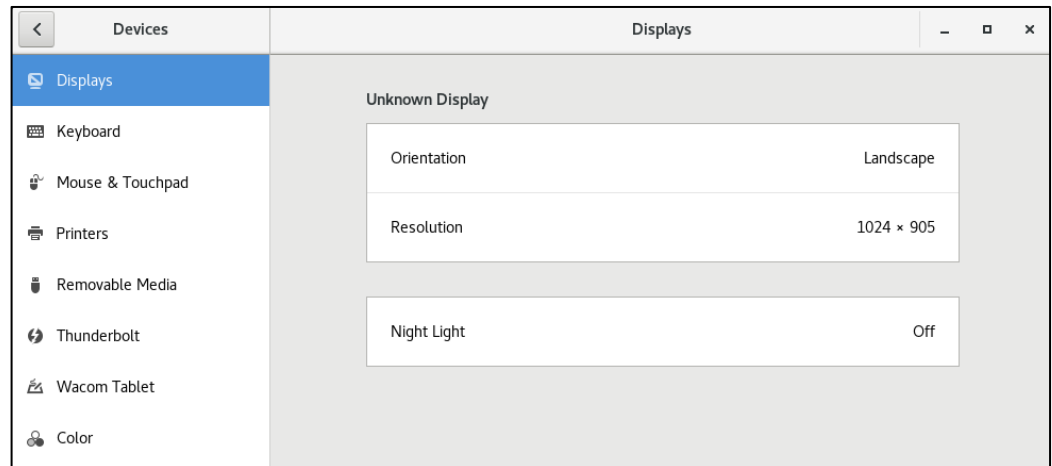


Figure 10: Baremetal screen resolution settings

Screen Saver and Screen Lock:

To enable a screen saver with automatic screen lock, you must enable the screen saver and the locking in **Settings > Privacy > Screen Lock** AND enable screen blanking in **Settings > Power**.

6.2.3.4.2 Host System Information Display

Clicking on the **System info** menu item displays detailed information about the CPU, the RAM, and the network interfaces of the host system.

This information may be needed when reporting a problem to Stromasys support.

It also provides some information about the CPU capabilities and the current resource utilization. See the following example:

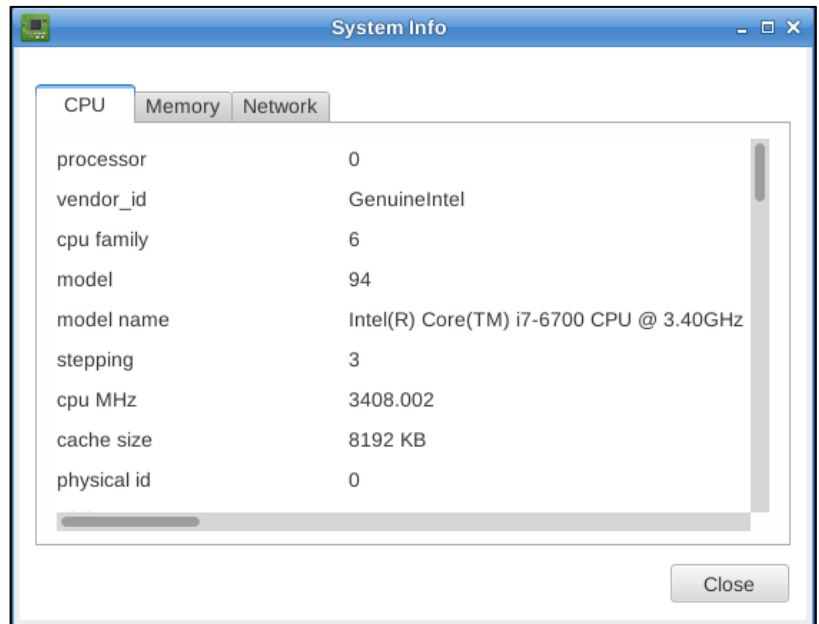


Figure 11: Charon-SSP Baremetal system information window

6.2.3.4.3 Update Menu Item

Clicking on **Update** opens a software update window like the one below. Users must enter the management password (initially set during installation or at first login) to perform updates.

This window enables the user to update the Charon-SSP product packages. For a detailed description of the update process, refer to [Upgrading the Charon-SSP Baremetal Distribution](#).

For larger upgrades involving the host operating system and the Baremetal-specific utilities, an ISO may be provided instead. The steps involved in installing such an ISO file are described in [Charon-SSP Baremetal Installation](#).

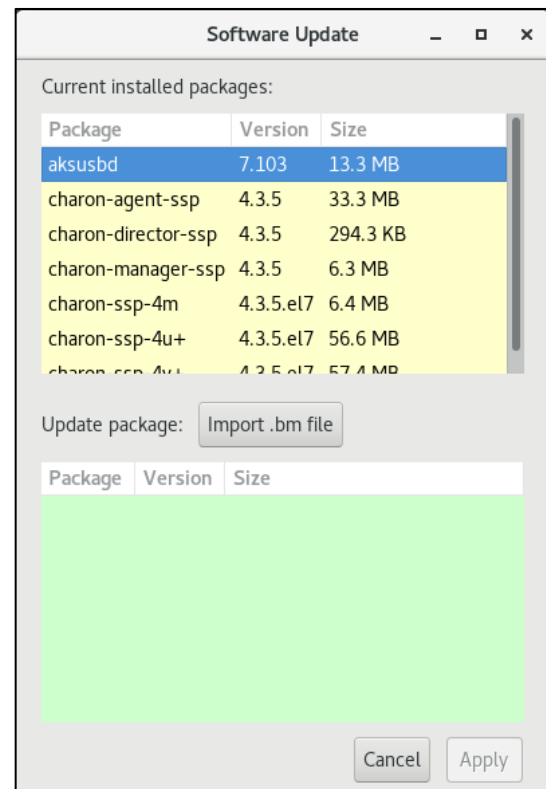


Figure 12: Charon-SSP Baremetal software update window

6.2.3.4.4 Key Manager

The Key Manager allows the user to enroll the Stomasys Public Key for UEFI secure boot. Charon-SSP Baremetal includes Linux kernel modules for PCI pass-through (Charon-SSP/4U and 4U+) and accelerated CPU virtualization using VT-x/EPT and AMD-v/NPT (Charon-SSP/4U+/4V+). These kernel modules are loaded on demand when the Charon-SSP instance starts.

Since Charon-SSP version 2, the Baremetal product has supported UEFI boot (in addition to the traditional BIOS boot). If UEFI secure boot is used, every kernel module that is to be loaded into the kernel must be digitally signed. Otherwise, it cannot be loaded successfully.

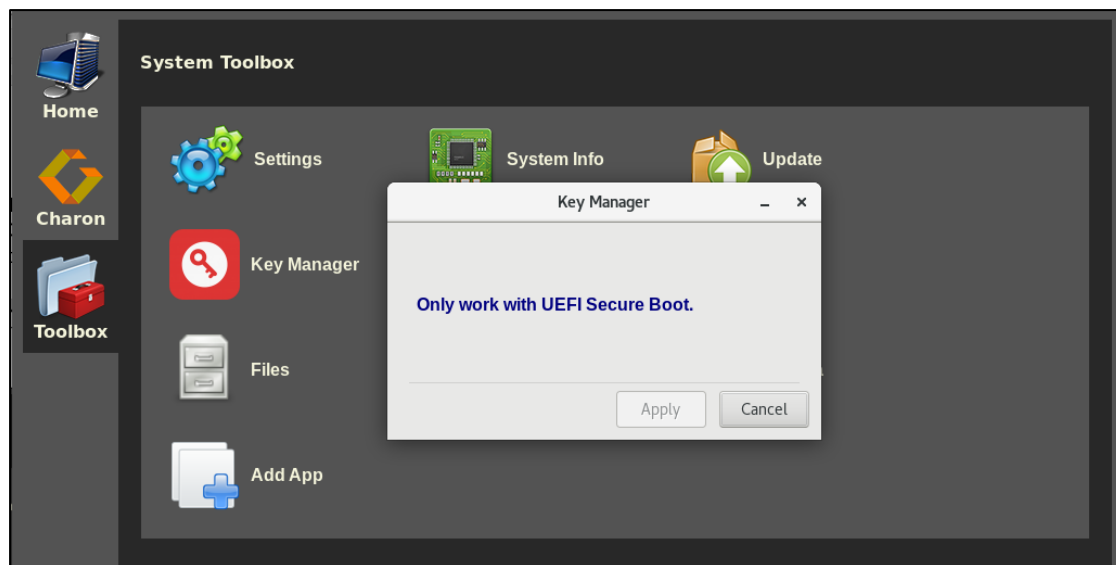


Figure 13: Key Manager on non-UEFI secure boot system

The required key can be enrolled using the **Key Manager**. Please refer to [Enrolling the Stomasys Public Key for UEFI Secure Boot](#) for more information.

6.2.3.4.5 Terminal

The **Terminal** icon opens a terminal window in the account of the **charon** user.

Access to the root account is possible using the **sudo** command.

Please note:

The Baremetal system is configured by default to support the Charon-SSP kernel modules required for Charon-SSP/4U+/4V+ and the PCI pass-through features. Updating to a kernel not provided by Stomasys will invalidate this support. Other restrictions may also apply. If you are unsure if a planned change of the system will cause problems with the operation of Charon-SSP, please contact your Stomasys partner or Stomasys support.

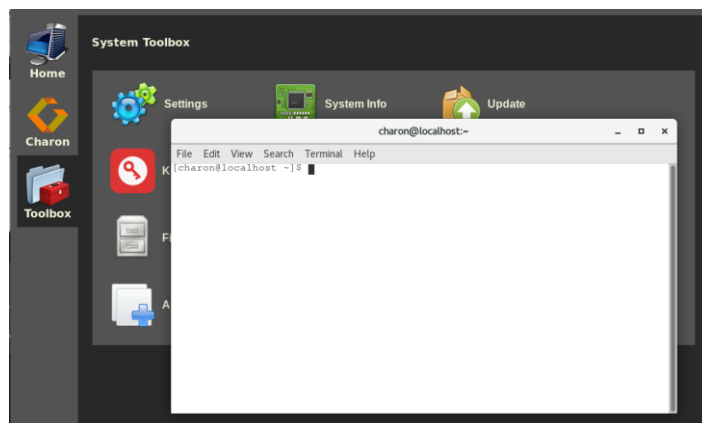


Figure 14: Baremetal terminal window

6.2.3.4.6 Additional Toolbox Applications

By default, the Toolbox contains additional standard Linux applications, in particular:

- The Firefox web-browser
- The FileZilla file transfer program
- The System Monitor (Performance data)
- The system file browser

The toolbox also contains the icon **Add App**. This icon enables the user to add additional desktop applications to the Toolbox.

The example shown here illustrates how to add the **gedit** editor to the toolbox:

This function replaces the customer-specific Baremetal applets of older Baremetal versions. The customer can now manually install applications and add a shortcut to the Toolbox, using the **Add App** icon.

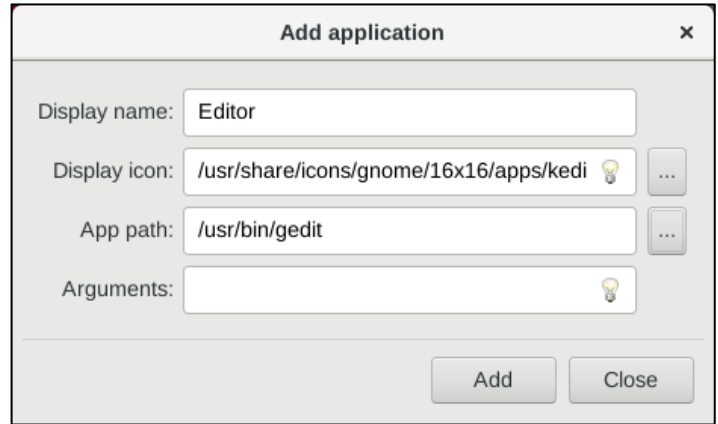


Figure 15: Baremetal Toolbox Add App

7 Configuring and Using the Charon-SSP Software

7.1 Overview

The following sections provide a detailed description of the individual parts of Charon-SSP for Linux and their use.

For the Charon-SSP environment, the most relevant documentation is the section [Using the Charon-SSP Manager](#). It describes the primary interface used to manage the hosted emulated SPARC systems. However, the Charon-SSP products also provide command-line access. For this, please refer to the sections [Using Charon-SSP from the Command-Line](#) and [Using the Charon-SSP Agent](#) in this user's guide.

Unless otherwise specified, the terms Charon-SSP/4U and Charon-SSP/4V also include Charon-SSP/4U+ and Charon-SSP/4V+.

7.2 Charon-SSP Directory Structure

The Charon-SSP Software is installed under the `/opt` directory of the host system. Below, is a short summary of the content of the individual directories when all Charon-SSP packages are installed. It focuses on items that may be of interest from a user's perspective. **The following is an overview, not a complete inventory.**

| | |
|-----------------------------|---|
| /opt/charon-agent | |
| └─ ssp-agent | |
| ├─ agent-log | Agent log files. |
| ├─ bin | The ssp-agent executable. |
| ├─ etc | Additional configuration files (e.g., hasplm.ini template, passwd file, etc.) |
| ├─ ssp | vm.dat containing information about Charon instance know by the agent (not user editable) |
| ├─ sun-4m | Default location (corresponding to architecture) for virtual SPARC system configuration files, emulator and console log files under a directory with a name corresponding to the virtual SPARC name. |
| ├─ sun-4u | |
| ├─ sun-4v | |
| ├─ utils | charon-passwd program to set the management password from the command-line |
| ├─ license | License utilities, e.g., hasp_srm_view, hasp_update. |
| ├─ mkdisk | Makedisk utility. |
| └─ mktape | Maketape utility. |
| /opt/charon-director | |
| └─ ssp-director | |
| ├─ bin | The ssp-director executable. |
| ├─ config | Charon-SSP Director configuration data. |
| └─ resource | Charon-SSP Director internal resources. |
| /opt/charon-manager | |
| └─ ssp-manager | |
| ├─ bin | The ssp-manager executable and utilities used by the Manager. |
| ├─ config | Charon-SSP Manager configuration data, e.g., X11 server configuration. |
| ├─ help | Currently not used. |
| ├─ resource | Charon-SSP Manger internal resources. |
| └─ ssp | Temporary files (e.g., cached Charon instance configurations and built-in console cache) |
| /opt/charon-ssp | |
| ├─ ssp-4m | Charon-SSP executables (by architecture), default configuration files (ssp4v.cfg, ssp4u.cfg, ssp4m.cfg), image to run the graphical console locally, etc. The open-source software license documents are stored in /opt/charon-ssp/oss-license. |
| ├─ ssp-4u | |
| └─ ssp-4v | |
| /opt/baremetal | |
| | Baremetal specific utilities |

7.3 Interaction of the Charon-SSP Components

The Charon-SSP software components together provide the environment for running and managing virtual SPARC systems. The management components can run on the same or on different systems as the virtual SPARC machines. The following image provides a first overview of the interaction between the Charon-SSP components:

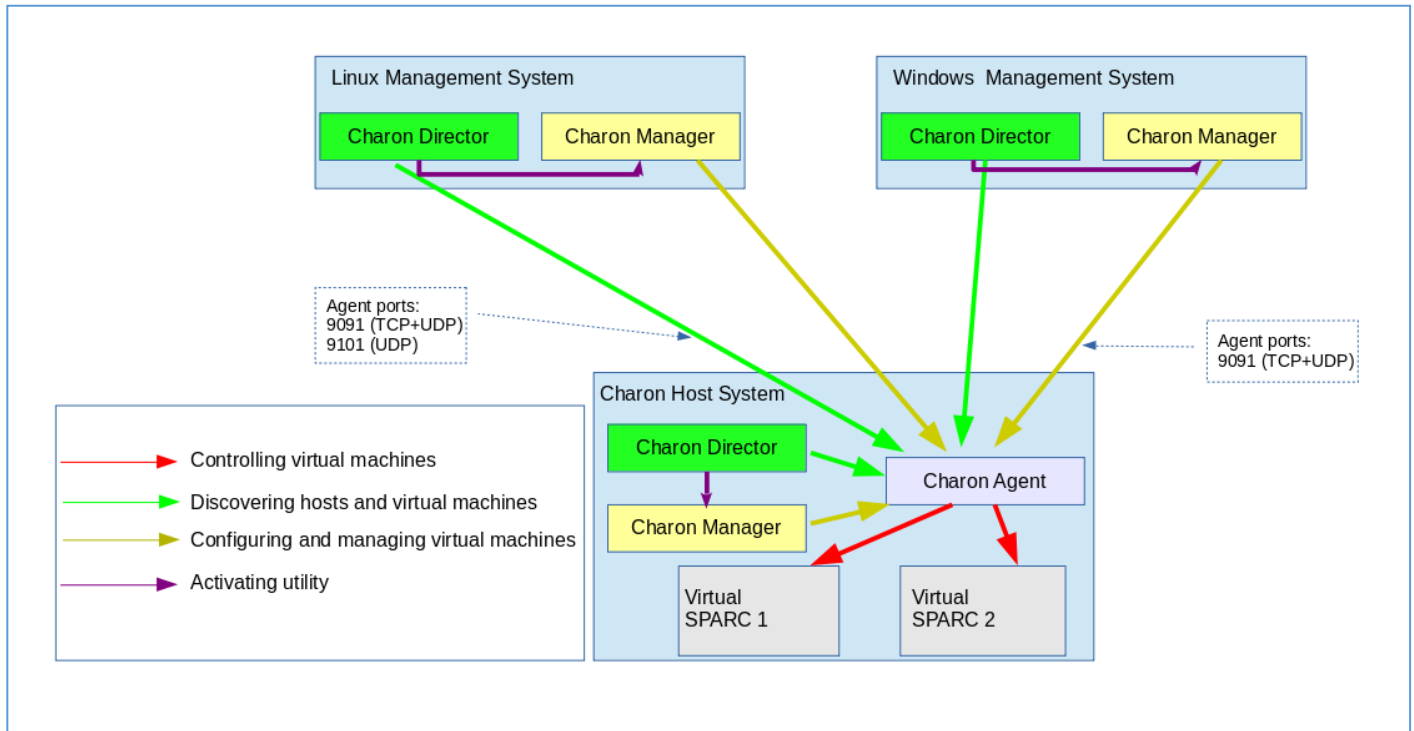


Figure 16: Charon-SSP components and their interaction

The Charon-SSP Agent uses port 9091 (TCP and UDP) to communicate with the Charon-SSP Manager and Director. For the communication with the Charon-SSP Director, port 9101 (UDP) is also required. These components are described in the following sections.

7.4 Using the Charon-SSP Director

The Charon-SSP Director is a GUI-based system for managing multiple host systems that each run one or more Charon-SSP guests. It can run on the Charon host system or on another supported Linux, Microsoft Windows or Baremetal system and requires the Charon-SSP Manager to be installed on the same system. The Charon-SSP Director can detect host systems on the same subnet automatically. Other systems can be added manually. This software component is especially useful in environments with several host systems.

7.4.1 Starting the Charon-SSP Director

On **non-Baremetal Charon-SSP installations**, the Charon-SSP Director can be started by **clicking on the associated icon**, or **via the command-line**.

- Use the following command to start the Charon-SSP Director from the command-line on Linux:

```
$ /opt/charon-director/ssp-director/ssp-director
```
- On **Charon-SSP Baremetal installations**, the Charon-SSP Director can be started by clicking on the **Charon Director** icon on the **Charon** screen.

Please note: When starting the Charon-SSP Director on a system with a non-operational Charon-SSP Manager path configured, it will prompt the user to select the desired Charon-SSP Manager image (select the correct version and the *ssp-manager* executable from the respective *bin* folder). This can happen, for example, if the Charon-SSP Manager is not installed at the default location in order to keep several versions of the Manager on the system.

7.4.2 Working with the Charon-SSP Director

Starting the Charon-SSP Director opens the main screen with automatically detected and manually added systems:

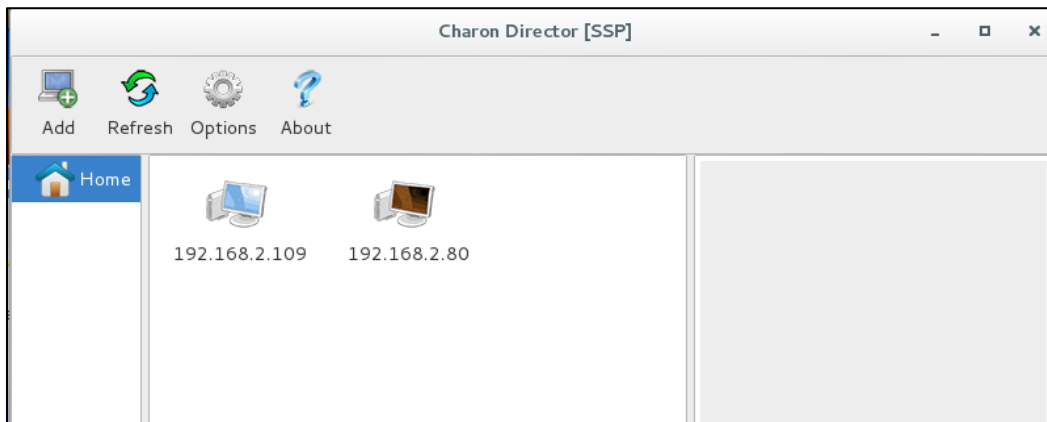
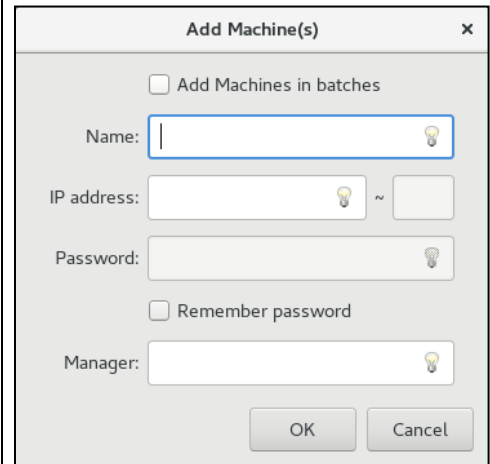


Figure 17: Charon-SSP Director – Main window

7.4.2.1 Charon-SSP Director Main Menu Bar

The main menu bar of the Charon-SSP Director offers the following options:

| Menu item | Description |
|----------------|---|
| Add | <p>If systems have not been detected automatically, Charon-SSP host systems can be added manually to the Charon Director:</p> <ol style="list-style-type: none"> 1. Click on the Add symbol. A new window opens. 2. If required, select the batch option. 3. For a single system, add Name, IP Address, and optionally the password required to connect to the Charon-SSP Agent of this system. <ol style="list-style-type: none"> a. To add the password enable Remember Password. 4. When adding a group of systems, specify the starting and ending IP Address. 5. Optionally use the field Manager to add the path to a non-default Charon-SSP Manager. This can be helpful if the default Charon-SSP Manager has a different version than the one installed on the target system. 6. Click OK to return to the main page. |
| Refresh | To refresh the host system status via the automatic discovery process, click on the Refresh symbol. |
| Options | <p>Additional Charon-SSP Director options:</p> <ol style="list-style-type: none"> 1. Click on the Options symbol. This opens a new window. 2. Optional: activate the refresh interval and set it to your preferred value. 3. Optional: you can choose to remove unresponsive systems automatically. 4. If the Charon-SSP Manager has been installed in a non-default location, you can specify the location of the default Charon-SSP Manager here. 5. Click OK to return to the main page. |
| About | Displays the version of the Charon-SSP Director. |



7.4.2.2 Managing Charon-SSP Director Subgroups

The Charon-SSP Director allows grouping systems into subgroups. Subgroups are managed from the context menu in the left pane of the window. Right-click on an **empty space** under **Home** to open the subgroup context menu. Opening the context menu **when an existing subgroup is selected** will add a new subgroup under the existing group.

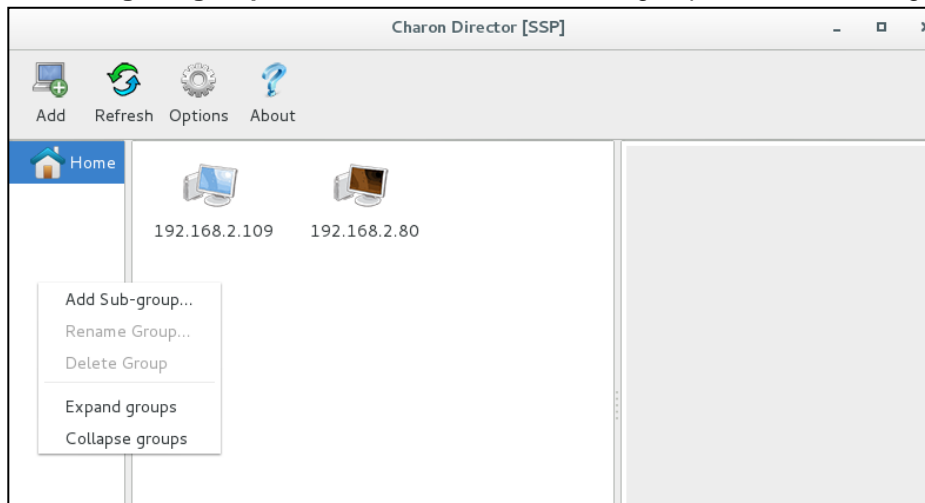
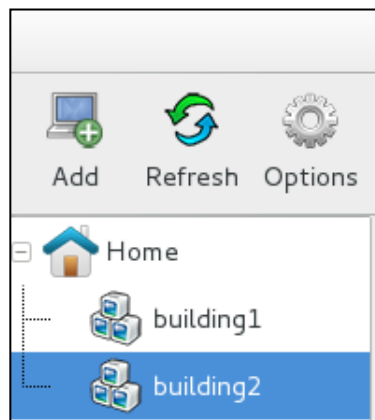


Figure 18:Charon-SSP Director sub-group menu

The Charon-SSP Director subgroup menu has the following items:

| Menu item | Description |
|------------------------|--|
| Add Sub-group | Clicking on this option opens a window to specify the name of the new subgroup. After confirmation, the subgroup will be shown under Home or an existing subgroup (depending on where the context menu was opened). |
| Rename Group | Prompts for the new name of the selected group. |
| Delete Group | Deletes the selected subgroup and groups below it, but not the systems within the groups. |
| Expand groups | Expands the list of subgroups under the selected position (Home or a specific group). |
| Collapse groups | Collapses the list of subgroups under the selected position (Home or a specific group). |



The parent group always shows all systems of the subordinated groups. The **Home** screen shows all systems.

7.4.2.3 Charon-SSP Director System Context Menu

When **selecting one of the displayed systems** in Charon-SSP Director, information about the system is displayed in the right-hand pane, including the number of Charon-SSP instances on the system—overall and the number of active Charon-SSP instances.

A **right-click** on the **selected** system opens a context menu with the following options:

| Menu item | Description |
|---------------------------|--|
| Connect to machine | Open the Charon-SSP Manager to connect to the Charon-SSP Agent of the selected system. If you did not store the agent password for the target system, this will open the Charon-SSP Manager's login window: <ol style="list-style-type: none"> 1. Enter the password for the Charon-SSP Agent of the target machine. 2. If required, enable the Charon Manager integrated SSH tunnel. 3. Click on Connect. 4. The Charon-SSP Manager opens with the target machine information. Continue with the section <i>Using the Charon-SSP Manager</i> below to learn about the functions of the Charon-SSP Manager. |
| Edit machine | Change the parameters of the system. You can change the same parameters as when adding a host system manually. |
| Update | Refresh the status display for the machine. |
| Copy | Copy the machine definition. It can be pasted by right-clicking in an empty space in the machine list window (e.g., of a different subgroup) and selecting Paste from the context menu. |
| Delete | Delete the selected system. This will delete the system from the current group and all subgroups of the group. If used from Home , the system will be deleted from Charon-SSP Director. |
| Select all | Select all systems in the system display section. |

7.4.2.4 Charon-SSP Director Additional Context Menu

When **clicking** on an empty space in the system list area, a small context menu opens with the following functions:

| Menu item | Description |
|--------------------|--|
| Add machine | Add one or more systems to the current group. See the description in the <i>Main Menu Bar</i> section above for more detail. |
| Paste | Insert previously copied machine definitions. |
| Select all | Select all systems in the system display section. |

7.4.2.5 Charon-SSP Director Keyboard Shortcuts

The following keyboard shortcuts are available in Charon-SSP Director:

| Action | Shortcut |
|-------------------|----------|
| Copy | Ctrl+C |
| Paste | Ctrl+V |
| Select all | Ctrl+A |
| Delete | DEL |

7.5 Using the Charon-SSP Manager

The Charon-SSP Manager is the graphical management interface for Charon-SSP. With the Charon-SSP Manager, you can manage multiple virtual SPARC systems, Sentinel HASP licenses, and virtual networks on the local host and on remote hosts.

The Charon-SSP Manager can run on the same machine as the Charon-SSP emulator or on a remote system. It can run on a Baremetal system or on a supported Linux system.

The following sections describe how to use the Charon-SSP Manager for the different aspects of managing Charon-SSP.

Please note:

- In the Charon-SSP emulator software, the term **virtual machine** denotes an emulated SPARC system. It is not to be confused with virtual machines running in a hypervisor, such as VMware.
- The screenshots in this section are mostly from conventional or Baremetal systems. The Charon Manager of a cloud-specific image will only show the features supported by this image.
- If you must provision a large number of emulator instances, the command-line utilities to create, verify, and import emulator configurations may be helpful. They are available starting with Charon-SSP version 5.2.6 (see the [command-line utilities in the appendix](#)).

The main Charon-SSP Manager topics in the following sections are:

- [Creating a Virtual Machine](#)
- [Configuring a Virtual Machine](#)
 - Model configuration
 - CPU and optimization configuration
 - Memory configuration
 - Graphics configuration
 - Storage configuration
 - Terminal device configuration
 - Ethernet configuration
 - Audio configuration
 - GPIB device configuration
 - USB configuration
 - License settings
 - Log configuration
- [Virtual Machine Context Menu](#)
 - Running a virtual machine
 - Accessing the virtual machine settings
 - Renaming a VM
 - Removing a VM from the Charon-SSP Manager list
 - Deleting a VM from disk
- [Host System Network Configuration](#)
- [Miscellaneous Management Tasks](#)

Editing the configuration of an emulator instance after an upgrade from an older version of Charon-SSP may apply changes to the configuration of an emulated system that will make the configuration incompatible with the older version of Charon-SSP.

It is therefore **strongly recommended to save a backup copy of the emulator configurations** before using a new version of the Charon-SSP Manager or manually adding new features.

The default configuration file location when the Charon Manager is used to configure emulated SPARC systems:

```
/opt/charon-agent/ssp-agent/ssp/<sparc-family>/<vm-name>
```

If the Charon-SSP Manager is not used, the location can be selected by the user at his/her convenience. To save the files, you can, for example, just copy the files to a safe location or save them as a TAR or ZIP archive to an external device.

7.5.1 Starting the Charon-SSP Manager

On **non-Baremetal Charon-SSP installations**, the Charon-SSP Manager can be started by **clicking on the associated icon**, or **via the command-line**. Use the following command to start the Charon-SSP Manager from the command-line:

```
$ /opt/charon-manager/ssp-manager/ssp-manager
```

On **Charon-SSP Baremetal installations**, the Charon-SSP Manager can be started by clicking on the **Charon Manager** icon on the **Charon** screen.

7.5.1.1 Connecting to the Charon-SSP Agent of the Target Host System

Starting the Charon-SSP Manager opens a login window with two tabs like the sample shown below:

Login window: Login tab

On this tab perform the following steps:

- Enter IP address or hostname of your Charon-SSP instance into the **IP address** field (**localhost** for local system).
- Enter the Charon-SSP management password into the **Password** field. **At first login**, leave the field empty. You will be prompted to set a new password (*Baremetal only*: the password can also be set during the installation and applies to all password protected functions of the Baremetal system). Until it is changed, this management password is then valid for all subsequent Charon Manager logins, and logins of the users *charon* and *sshuser* (if applicable).

Set **SSH tunnel** to **ON** if the connection runs over a public network and traffic must be encrypted. It should be set to **OFF** when managing the local system (a tunnel to localhost will not work).

Login window: SSH tab

If using the Manager across a public network without existing VPN, perform the following steps:

- Conventional or Baremetal installation: if you have not installed the required keypair yet, please refer to [Creating and Uploading the Public SSH Key](#). Cloud-image installations: use the keypair associated with the instance at launch.
- Enter the Charon-SSP user into the **Username** field (**sshuser** or **charon** for Baremetal and cloud-specific systems; user with matching public key installed for other product versions).
- Enter the path to the private key file into the **Private key** field (click on the three dots to open a file browser). The corresponding public key must be in the file `~/.ssh/authorized_keys` of the user entered above.
- Enter the **Passphrase** for the private key if required.
- Adjust the SSH **Server port** (default 22) if required.

After entering the required information, click on **Connect**. This opens the main screen of the Charon-SSP Manager as described in the next section, or, **at first login**, prompts you for the new management password before continuing with the login.

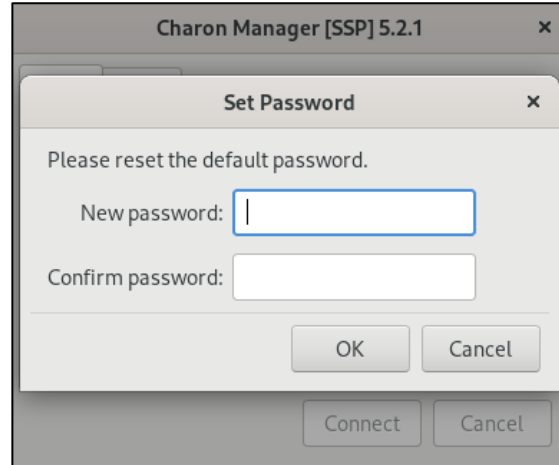
Initial password prompt:

If this is the first login, you will be asked to set a new password and to confirm it. Until it is changed, this management password is then valid for all subsequent Charon Manager logins, and logins of the users *charon* and *sshuser* (if applicable). It is not removed if Charon-SSP is reinstalled. Note that **older versions** of the product will not prompt for the password at first login but will use a default password (**stromasys**).

Click on **OK** to confirm your input.

When entering the password on a Baremetal system, bear in mind that the default keyboard layout is US English

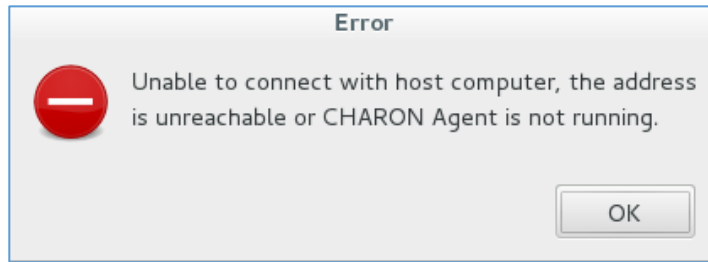
After completing the initial password change, the normal login process continues.

**Additional information:**

- On cloud instances, the initial password change requires that the embedded SSH tunnel of the Charon Manager be enabled, unless the Charon Manager runs on the cloud host itself and is displayed via SSH X11-forwarding (i.e., the Charon Manager is connected to localhost).
- The embedded SSH tunnel of the Charon Manager requires that **AllowTcpForwarding** be set to **yes** in */etc/ssh/sshd_config* of the target SSP host system (this should be the default).
- The Charon-SSP Agent listens on port **9091**. To connect directly (without the embedded SSH tunnel) to a remote Charon-SSP agent, make sure that this port is not blocked by a firewall.
- If you use the Charon-SSP Manager to manage remote Charon-SSP host and guest systems, additional ports may have to be allowed to pass through intermediate firewalls (e.g., the TCP port configured for an emulated serial console port, **unless** the embedded SSH tunnel is used).
- If using the embedded SSH tunnel of the Charon Manager, note that not all applications will be routed through this tunnel. For example, X11 will not use this tunnel. It is mainly designed to protect the management traffic and the serial console traffic. To protect all traffic, use an encrypted VPN connection.
- **To reset a lost management password**, please refer to [Resetting a Forgotten Management Password](#).

7.5.1.2 Troubleshooting information

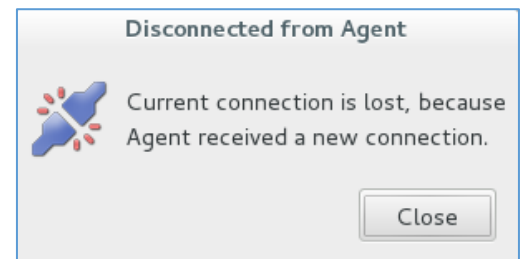
If you receive an error like the one displayed in this screenshot, verify that the host specified in the **IP address** field and **any firewall settings** are correct, and that the Charon Agent is running. On Charon-SSP systems installed using the RPM package installation, also check if the Charon-SSP agent is installed.



Use the following steps to check if the Charon-SSP Agent is running on the target system and to start it, if required:

| Step | Description |
|------|---|
| 1 | Log in on the Charon host (on Baremetal open a terminal window). |
| 2 | To check if the Charon-SSP Agent is running, use the following command: <pre>\$ ps -ef grep -i charon-agent</pre> If the agent is running, you will see an output like the following: <pre>root 10573 1 0 Feb09 pts/6 00:01:37 /opt/charon-agent/ssp-agent/ssp-agent</pre> |
| 3 | If the Charon-SSP Agent is not running, use the following commands to (re-)start it: <pre># systemctl start ssp-agentd</pre> or <pre># systemctl restart ssp-agentd</pre> If the Charon-SSP agent was not stopped cleanly using the stop command, it may not start using the start command. In such cases, the restart command can be used. |

There can only be a single connection to the Charon-SSP Agent at any time. A second connection from a different Charon-SSP Manager client disconnects the first connection, and the following message is displayed.



If you receive the error that the SSH tunnel cannot be created but there is no problem with basic connectivity, check the settings of the **sshd** process on the target system (AllowTCPForwarding). Also verify that you use an SSH key type supported by Charon-SSP (if using a very new key type, test with type RSA to verify this possibility).

7.5.1.3 Running the Charon-SSP Manager via SSH X11-Forwarding

In most cases, the Charon Manager will run on a local console of the Charon host system, or on a remote management system connecting to the Charon host system via the network.

However, in some cases it may be necessary to run the Charon Manager locally on the Charon host and display it on a remote Xserver via SSH X11-Forwarding.

This section provides a short overview of the steps required to configure SSH X11-Forwarding:

1. If not already there, copy the Charon Manager RPM package to the Charon host. For cloud images, the location of the Charon Manager packages is **/charon/storage**.
2. Log in to the Charon host as the **root** user.
3. Go to the location where the Charon Manager RPM package is stored (**# cd <path-to-package>**)
4. Install the Charon Manager package (**# yum install charon-manager-ssp-*{version}*.rpm**). On Linux 8.x, use the **dnf** command instead of **yum**.
Please note: access to a package repository is required to install the Charon Manager dependencies.
5. Install the **xorg-x11-xauth** package (**# yum install xorg-x11-xauth**).
6. Allow X11 forwarding in the SSHD configuration file (edit **/etc/ssh/sshd_config** and ensure that the parameter **X11Forwarding** is set to **yes**).
7. Restart the SSHD (**# systemctl restart sshd**).
8. Log out.
9. Log back in again using **ssh -X** to enable X11 forwarding.
10. Start the Charon Manager (**\$ /opt/charon-manager/ssp-manager/ssp-manager**).
11. After a little while (depending on network performance) the Charon Manager window should open on your local desktop. Connect to **localhost** (that is the Charon host on which the Manager is running).

Please note:

- If your local system is a **Microsoft Windows** system, you can use PuTTY in connection with an Xserver, or you can use an integrated tool such as MobaXterm.
- Should you receive the error “**X11 connection rejected because of wrong authentication.**” for any application, it is possibly caused by this specific application being run under a different user than the user to which the **ssh -X** was executed. If this happens when running a command via **sudo**, you can try the following workaround:
 - On the Charon host add the following line to the file **/etc/sudoers**:
`Defaults env_keep += "DISPLAY XAUTHORIZATION XAUTHORITY"`
 - On the Charon host under the user to which the ssh command was executed, define the following environment variable:
`export XAUTHORITY=$HOME/.Xauthority`

If your program starts another program under a different user, try to SSH to the account under which the program is run.

- Local graphics emulation does not work properly across X11-Forwarding because it assumes a local graphics environment. Local graphics emulation can be started successfully if the Charon host desktop is accessed via VNC. However, depending on the network connection, the performance is not sufficient in many cases.

7.5.1.4 Charon-SSP Manager Overview

This section provides a first overview of the available menu functions and symbols of the Charon Manager.

General information:

- Charon-SSP Manager menu items are adapted to the type (e.g., conventional or Baremetal) of the managed target system. Hence, the following images show the Charon-SSP Manager overview for the conventional product, the Baremetal product, and for cloud-specific Charon-SSP product. Other Charon-SSP variants may show slight differences.
- Not all Charon-SSP Manager menu items shown below are supported on all types of Charon-SSP host systems. If a function is not supported on the host system, the menu item may not be displayed. Example: the **VNC Server** item is only supported on Baremetal systems. Hence, it will not be displayed when managing conventional RPM-based Charon-SSP host systems.
- The **title bar** of this screen indicates the managed system type in square brackets. In case of a cloud instance, it indicates the type of cloud. If the connection is created via the embedded SSH tunnel of the Charon Manager, the title bar will show that an SSH connection is being used. In the remaining sections of this document, screenshots of different Charon host systems may be used so the title bar may not always correspond to the Charon-SSP variant treated in this document. Older versions only show the address of the target system.

Charon-SSP Manager options if the managed host system is a conventional Charon-SSP or VE-enabled installation:

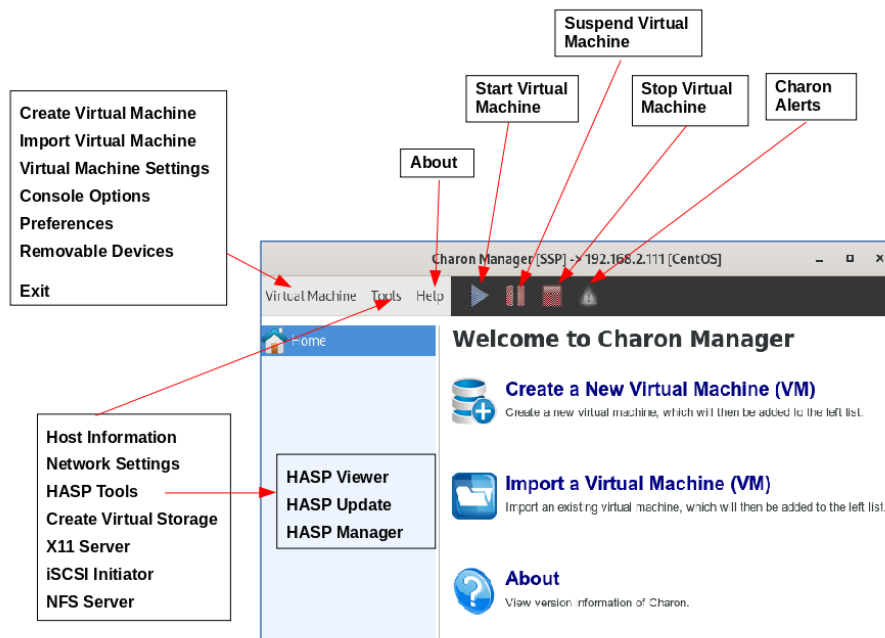


Figure 19: Charon-SSP Manager (conventional or VE) – Main window

Charon-SSP Manager options if the managed host system runs a Charon-SSP Baremetal installation:

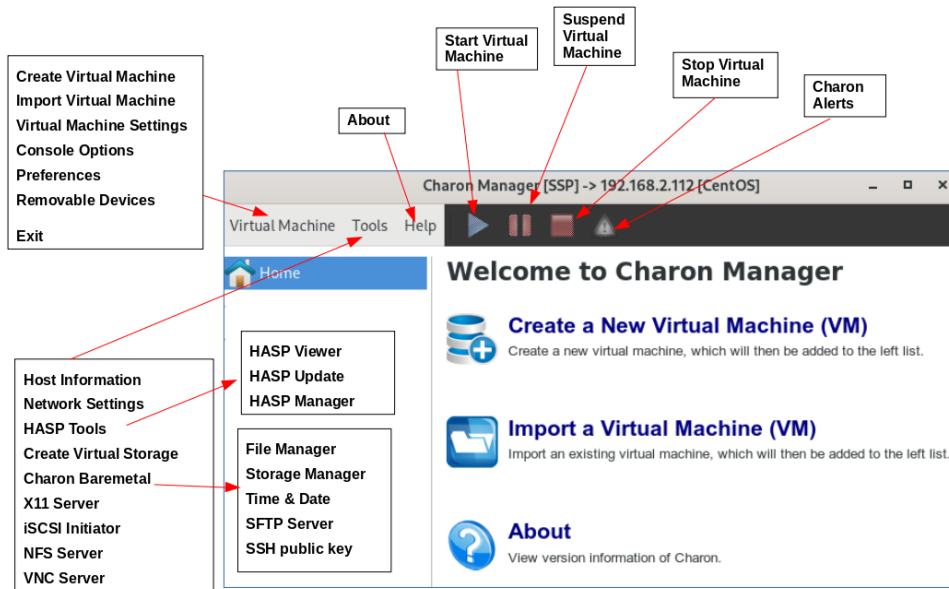


Figure 20: Charon-SSP Manager for Baremetal Systems

Charon-SSP Manager options if the managed host system runs a Charon-SSP Automatic Licensing product:

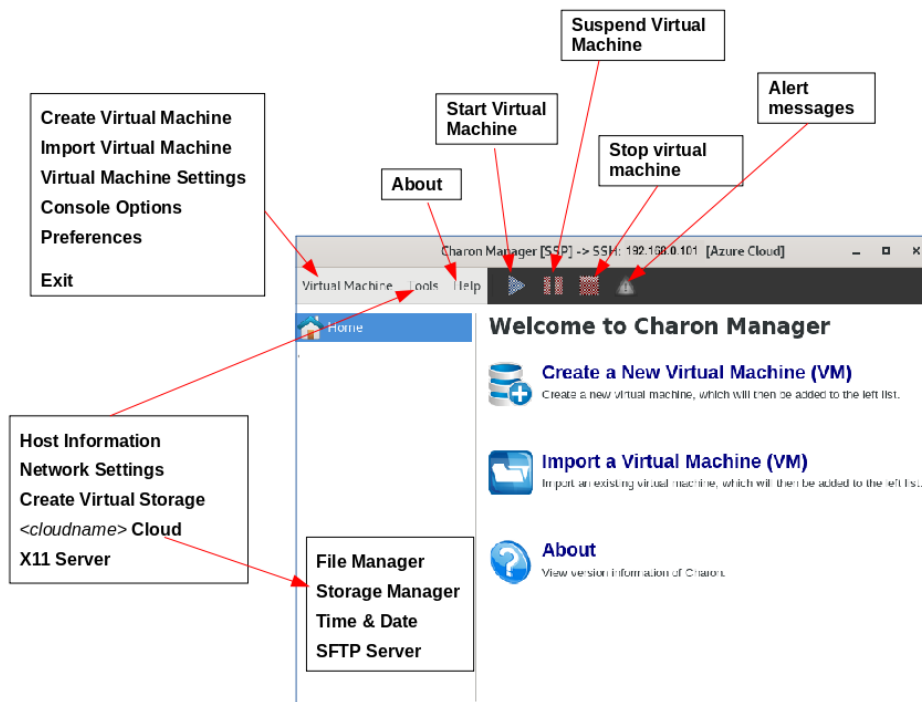


Figure 21: Charon-SSP Manager on a cloud-specific Charon-SSP AL instance

The Charon-SSP Manager **virtual machine pane on the left** has some additional functions applicable to all product variants:

- Double-clicking on **Home** sorts the virtual machines alphabetically. Repeating the action toggles between ascending and descending sort order.
- The position of a virtual machine in the list can be changed by manual drag-and-drop.
- A right-click in the pane when no virtual machine is selected opens a context menu to create or import a virtual machine.

For information on running the Charon Manager and the Charon Director on Microsoft Windows, please also refer to the appendix [Charon-SSP GUI for Microsoft Windows](#).

7.5.2 Creating a Virtual Machine

The first step to running an emulated SPARC system is to **create the initial configuration** using the following steps:

| Step | Description |
|------|---|
| 1 | <p>Open the New Virtual Machine window using either of the following methods:</p> <ul style="list-style-type: none"> From the opening screen titled Welcome to Charon Manager, click on Create a New Virtual Machine icon, <i>or</i> use the Create option in the Virtual Machine menu, <i>or</i> while Home is selected, right-click into an empty area in the virtual machine list pane and select the option to create a new virtual machine from the context menu. |
| 2 | <p>Select the appropriate Hardware Model by clicking the radio button labelled with the SPARC family that most closely matches the system to you wish to run.</p> <ul style="list-style-type: none"> The hardware family SUN-4M represents a SPARC V8 32-bit model. The hardware family SUN-4U represents a SPARC V9 64-bit model. The hardware family SUN-4V represents a SPARC V9 64-bit model with the 4V features. You can select either a SPARC T2 or a SPARC T4. <p>The configured model must be covered by your license.</p> |
| 3 | Enter a name for the emulated SPARC system in the Virtual machine name field. |
| 4 | Click on OK. |

The steps above create a basic new emulator configuration. The new virtual machine is listed the left-hand pane of the management interface showing the **Virtual machine name** you specified.

The screenshot below shows the management interface screen after two emulated SPARC systems were created.

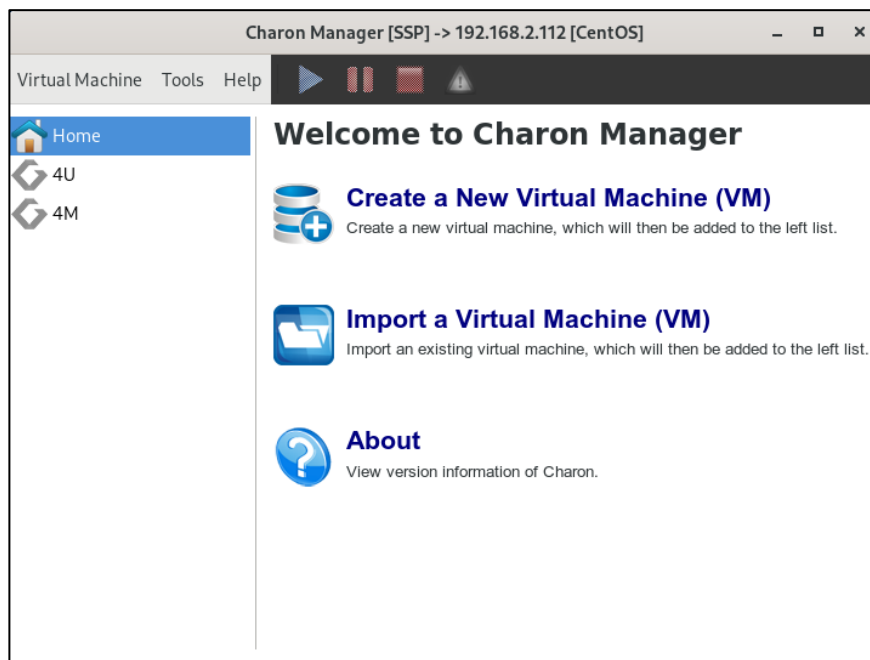


Figure 22: Charon-SSP Manager main window with virtual machine

The initial emulator configuration is only a configuration template. To complete the configuration, continue with the next section (*Configuring a Virtual Machine*).

7.5.3 Configuring a Virtual Machine

To open the emulator configuration window, first **select the virtual machine** in the left-hand pane of the Charon-SSP Manager. This shows the virtual machine overview in the right-hand pane, including the following tabs:

- **Summary** tab: Overview of the current configuration of the selected virtual SPARC.
- **Log** tab: Enables viewing the log files (VM log, TTYA/B log, crash log) of the selected virtual SPARC.
- **Console** tab: The built-in serial console of the selected virtual SPARC.

The image below shows an example of the summary page of an emulated SPARC on a non-cloud host system:

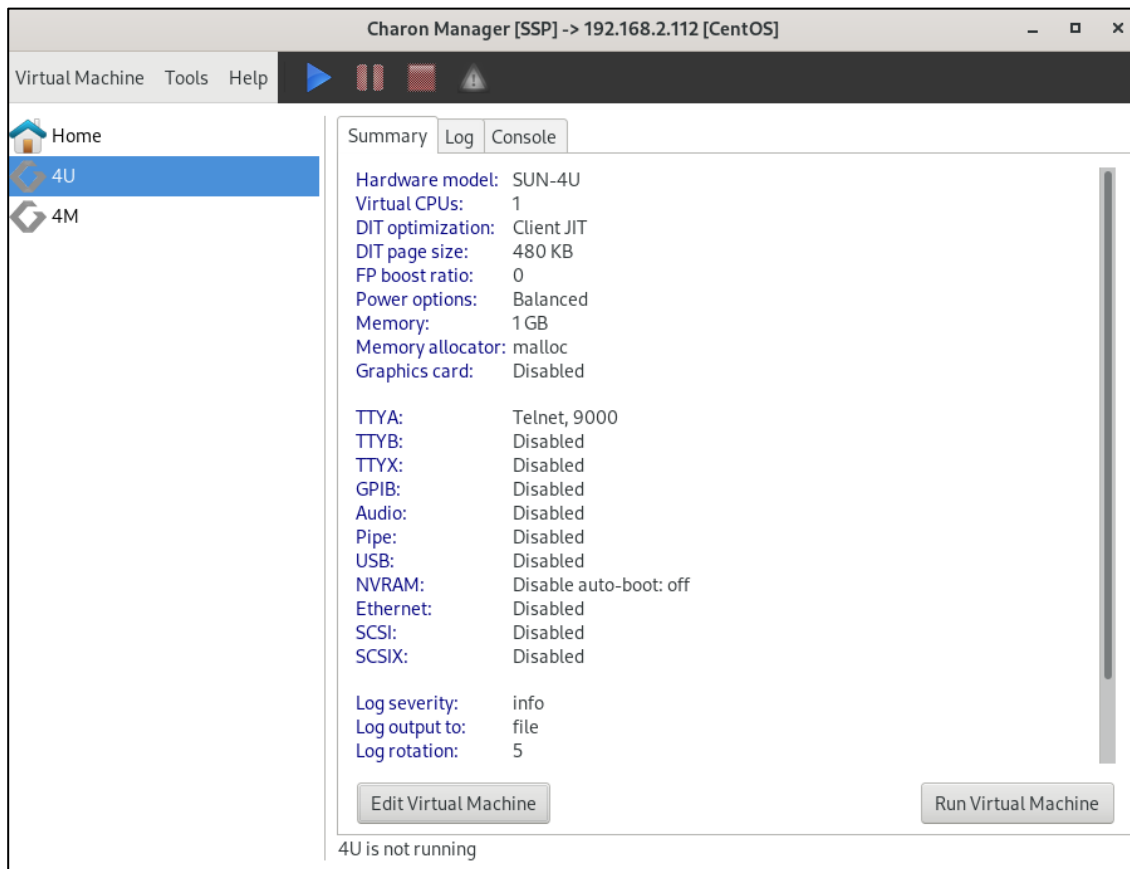


Figure 23: Charon-SSP virtual machine summary page

To continue with the configuration of the emulated SPARC system,

- click on the **Edit Virtual Machine** button in the Summary tab, or
- select **Virtual Machine Settings** from the emulated system context menu or the **Virtual Machine** menu.

This opens the **Virtual Machine Settings** window for the virtual machine.

Configuration changes are

- confirmed with **OK** and
- discarded with **Cancel** (the options are at the bottom of the configuration window).

For any configuration changes to take effect, the virtual machine must be restarted. However, it is also recommended that before making any configuration changes the virtual machine be shut down correctly.

The example below shows the configuration window of a SUN-4U system on an **on-premises** host system:

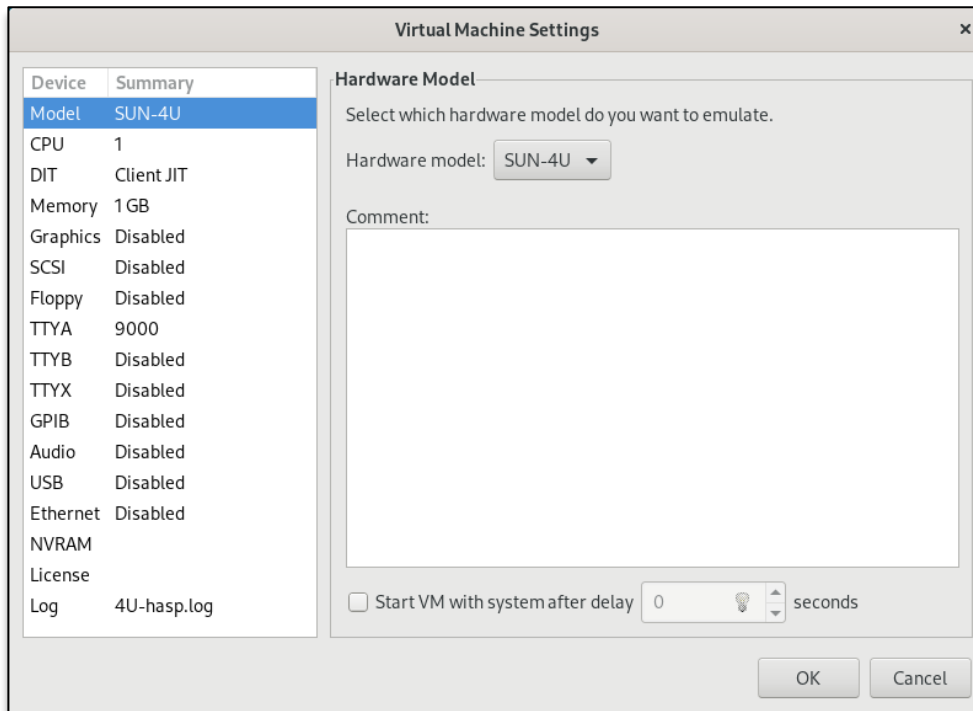


Figure 24: Charon-SSP virtual machine settings window (non-cloud)

The example below shows the configuration window of a SUN-4U system on a **Charon-SSP AL cloud-based** host system (with the **reduced feature set specific to cloud environments**):

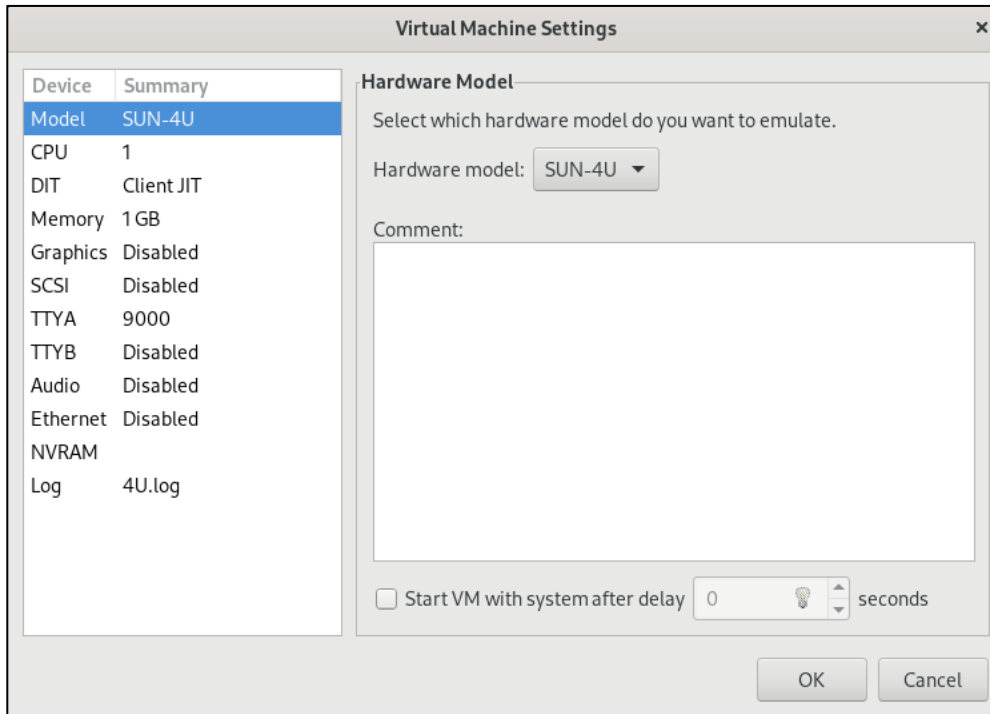


Figure 25: Charon-SSP Virtual Machine Settings Window (cloud)

The following sections describe the different configuration categories of the **Virtual Machine Settings** window.

7.5.3.1 Hardware Family Configuration (Model)

To view the virtual machine hardware family configured when creating the emulated system, select **Model** in the left-hand pane of the Settings window. This displays the current value in the **Hardware Model** pane.

Please note:

- The basic hardware model cannot be changed after creating the virtual machine.
- You can change the SPARC T2/T4 setting for SUN-4V.

Supported hardware families:

- **Charon-SSP/4M:**
 - Sun-4c and Sun-4m (represented by the Sun SPARCstation 20)
- **Charon-SSP/4U is:**
 - Sun-4u (represented by the Sun Enterprise 450)
- **Charon-SSP/4V(+)** is:
 - Sun-4v (represented by the SPARC T2 or the SPARC T4, depending on the selected type). **SPARC T4 is required to run a Solaris 11.4 guest system.**

Unless otherwise mentioned, the names **Charon-SSP/4U** and **Charon-SSP/4V** include **Charon-SSP/4U+** and **Charon-SSP/4V+**.

The following image shows a sample **Charon-SSP/4V** model configuration:

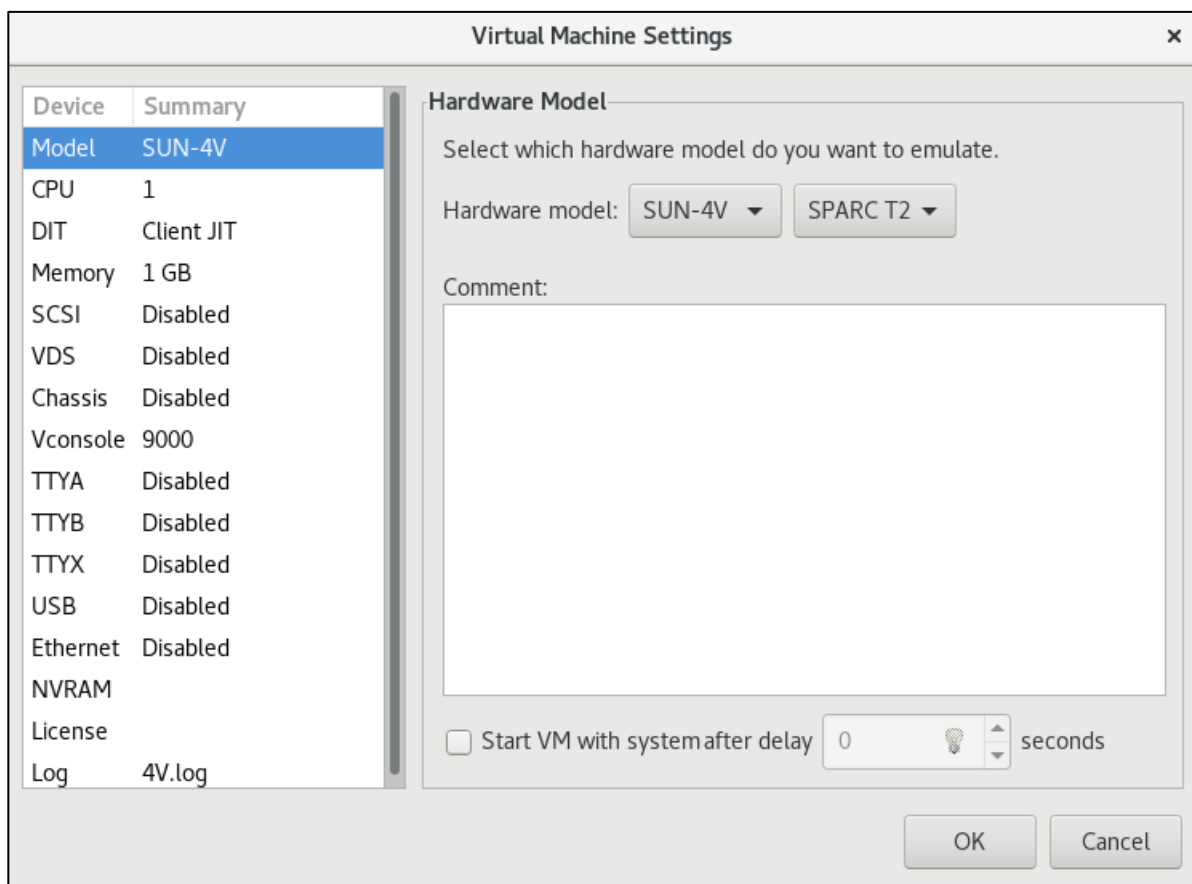


Figure 26: SUN-4V model configuration

Configurable options:

- The base architecture chosen (4M, 4U, or 4V) cannot be changed after creating the emulated SPARC system. For a SUN-4V, the selection of **SPARC T2** or **SPARC T4** can be changed (provided the guest OS supports it). The SPARC T4 is required for Solaris 11.4.
- **Comment** field: allows you to add additional optional information about the virtual machine.
- **Start VM with system:** This option integrates the startup of the Charon-SSP instance in the host system startup. With this option enabled, the virtual machine starts automatically when the system boots. Optionally, a delay can be configured (for example to wait for a license server to come online).
 - **Possible values for the delay parameter:** 0 to 300 seconds.
 - The auto-start information for an emulated system is stored in `/opt/charon-agent/ssp-agent/ssp/vm.dat`.

Additional information: the mechanism to start an emulator automatically when the host system boots changed starting with version 4.0.x. Charon-SSP no longer creates a start/stop script in `/etc/init.d` if this option is selected. Existing scripts are not deleted, but will no longer be activated. Instead, the Charon Agent is now responsible for starting emulator instances configured via the Charon Manager for starting with the host system boot.

If a VM is started automatically with the host system startup and stopped with host system shutdown, it is the responsibility of the user to **shut down the guest operating system cleanly** before host system shutdown. Failing to do so may cause data corruption in the guest system.

7.5.3.2 CPU Configuration

To view or change the current virtual machine CPU configuration, select **CPU** in the left-hand pane of the Settings window as shown in the on-premises host example below:

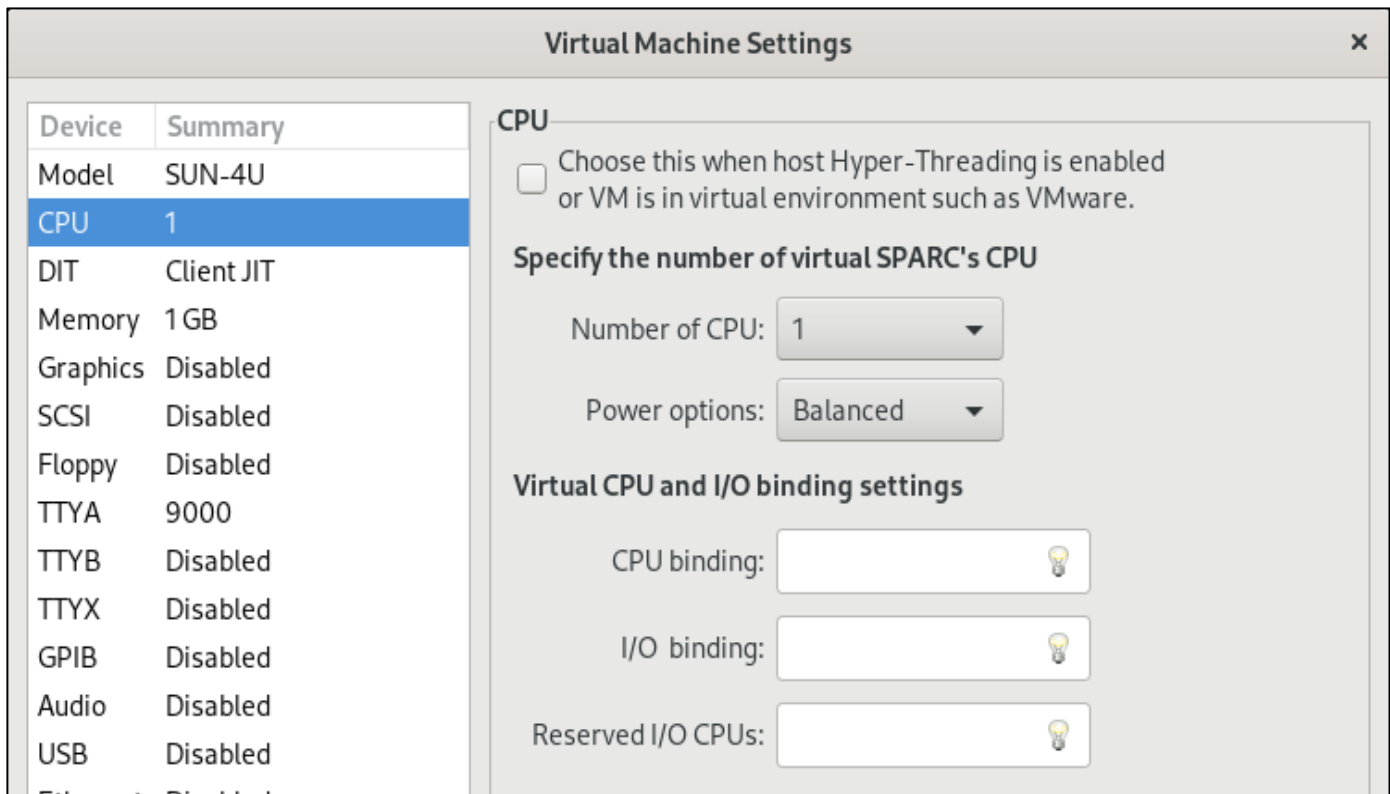


Figure 27: Charon-SSP virtual machine CPU configuration

The following table lists each of the **fields in the CPU configuration window** and describes their operation.

| Field | Description |
|---------------------------------|--|
| Hyper-threading checkbox | <p>With this mode enabled, Charon-SSP does not use CPU core affinity on the host system, but relies on the scheduler of the host operating system. In combination with the power option Power save (see below), this will allow rescheduling idle guest system CPU threads, thus making best use of the available, possibly limited, CPU capacity. For best performance, however, there should be enough CPU cores (= physical/real CPUs) to avoid two active emulator threads sharing one CPU core.</p> <p>Recommended setting:</p> <ul style="list-style-type: none"> The Charon-SSP hyper-threading option must be enabled if hyper-threading is enabled on the Charon host system (two threads enabled on one physical CPU core providing two logical CPUs to the host operating system). If the Charon host system is running as a VM under a Hypervisor (e.g., VMware) or in a cloud instance, where hyper-threading is often active by default, and you are not sure whether the underlying server has hyper-threading enabled or not, there will be no detrimental effect if you enable the Charon-SSP hyper-threading option as a precaution. |
| Number of CPU | <p>Configure the number of virtual SPARC CPUs. Supported number of CPUs:</p> <ul style="list-style-type: none"> Charon-SSP/4M: 1 to 4 virtual SPARC CPUs Charon-SSP/4U(+): 1 to 24 virtual CPUs Charon-SSP/4V(+): 1 to 64 virtual CPUs |

| Field | Description |
|--------------------------|--|
| Power options | <p>This option determines the host CPU behavior when the guest Solaris is in idle state (supported for Solaris 2.4 and above).</p> <ul style="list-style-type: none"> • Performance Choosing this option keeps the host CPU in a busy loop waiting for next Solaris activity. This option offers the best response time in Solaris but the host CPU usage is at 100% all the time. • Balanced (default) Choosing this option allows the host CPU to go into an idle state until the next Solaris activity. This option offers a good balance between Solaris response time and host CPU usage. • Power save (default in older versions if hyper-threading option was selected) The host CPU is in deep "sleep" mode when the guest Solaris is in idle state. If hyper-threading is enabled on the host system, the number of real/physical CPU cores is only half the number of vCPUs presented to the Linux operating system. In such environments, the hyper-threading option must be enabled. For best performance, the host system should have enough physical CPU cores to fully satisfy the emulator requirements. If this can be ensured, the power option can be set to Performance. In all other cases Power save must be used to allow optimal load distribution across the limited system resources. Since every customer environment is different, the above can only provide a general guideline and testing is always required to assess the overall performance of the system. |
| CPU binding | <p>Assign specific host CPUs to the processing of SPARC instructions. If configured, each virtual SPARC CPU must be assigned to exactly one specific host CPU core for instruction processing. This field consists of a comma-separated list of CPU IDs (index starts from 0). If left blank, the virtual machine software will assign affinity itself starting with the highest CPU ID (recommended). Cannot be used with hyper-threading mode enabled. CPU cores assigned to emulated CPUs are never shared between instances.</p> |
| I/O binding | <p>Assign specific host CPUs (depending on the hyper-threading configuration, the CPUs visible to the host OS can be CPU cores or logical CPUs) to the processing of guest I/O requests. This field consists of a comma-separated list of CPU IDs. If left blank, the virtual machine will assign I/O processing affinity itself starting from CPU ID 0 (recommended). CPUs listed here cannot be shared between instances.</p> <p>If there is an overlap with manually configured bindings in other instances or the automatically calculated CPU allocation for I/O, the instance will not start with the message:</p> <p>"Wrong IO affinity setting: already allocated by another thread."</p> |
| Reserved I/O CPUs | <p>Reserve CPUs on the host system for processing guest I/O requests. Allocation will start from the lowest CPU ID.</p> <p><u>Default:</u> if neither I/O binding nor Reserved I/O CPUs is set, Charon will assign 1/3 (min. 1; rounded down) of the CPUs visible to the Charon host to I/O processing starting from the lowest CPU ID (recommended). Depending on the hyper-threading configuration, the CPUs visible to the host OS can be CPU cores or logical CPUs.</p> <p>If there is an overlap between a manual configuration in one instance and the automatic calculation of CPUs used for I/O processing in other instances, overlapping CPUs available for I/O processing are shared between instances. If the number of CPUs used for I/O processing (configured or calculated automatically) + the number of emulated CPUs is higher than the number of available host CPUs, the emulator does not start and logs the error: "Wrong CPU affinity setting: no enough host CPUs."</p> |

Additional information:

- Manual I/O CPU bindings can be used to optimize I/O and DIT performance on a host system running multiple Charon-SSP instances, because it allocates dedicated CPUs for I/O processing to a system (no sharing).
- Manually configuring the number of CPUs reserved for I/O can be used to adjust the CPU pool used for I/O operations. Overlapping CPUs between several instances will be shared.
- Once any manual configuration is used, its influence on all concurrently active Charon-SSP instances must be considered in order to avoid undesired performance degradation.

Important: the section [Minimum Host System Hardware Requirements](#) provides information about the minimum CPU requirements for running Charon-SSP. However, the actual host system and emulator configuration must be adapted to the actual workload of the guest system. Customer-specific workload analysis and emulator-host sizing is beyond the scope of this user's guide. Please contact your Stromasys technical representative for guidance.

7.5.3.3 DIT Configuration

To view or change the current DIT configuration, select **DIT** in the left-hand pane of the Settings window.

There are three levels of DIT optimization:

- **OFF:** no DIT optimization.
- **Client JIT** or first level DIT: this is the equivalent of the DIT optimization available in older versions.
- **Server JIT** or second level DIT: this is a more aggressive optimization. It optimizes SPARC instructions at runtime, based on an MRU policy (Most Recently Used).

Server JIT is not available in Charon-SSP/4M. Client and server JIT are implemented in **two separate images** that will be configured automatically by Charon-SSP Manager depending on the configuration.

Comparison between the DIT configuration options:

| DIT Optimization | Translation speed | Command execution after Translation | Memory requirements |
|------------------|-------------------|-------------------------------------|---------------------|
| OFF | n/a | Slow | n/a |
| Client JIT | Fast | Faster | Approx. 2GB RAM |
| Server JIT | Slow | Fastest (depending on application) | Approx. 6GB RAM |

Due to the slower and more resource-consuming translation in Server JIT mode, mostly long-running applications will benefit from Server JIT.

The following sections show the details of both modes.

7.5.3.3.1 Client JIT

This section describes the configuration options available in Client JIT mode. The image below shows a sample of the Charon-4U configuration screen:

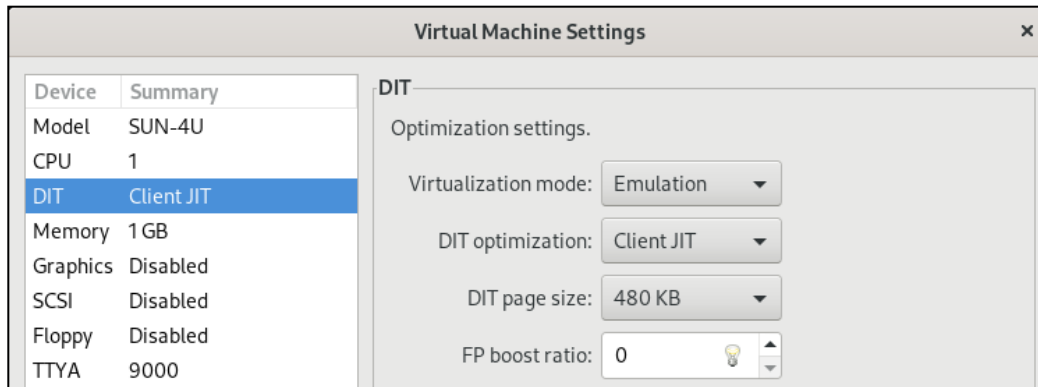


Figure 28: Charon-SSP Client JIT configuration window

The following table lists each of the **fields available for DIT Client JIT mode** and describes their operation.

| Field | Description |
|----------------------------|--|
| Virtualization Mode | <p>Possible values:</p> <ul style="list-style-type: none"> Emulation: selects the Charon-SSP/4U or Charon-SSP/4V emulator, Intel VT-x/EPT: selects Charon-SSP/4U+ or Charon-SSP/4V+ depending on the base hardware family selected. Also valid for AMD-v/NPT. This option is not available on Charon-SSP/4M, if Charon-SSP/4U+/4V+ is not installed, or on unsupported platforms. This setting is not stored in the emulator configuration file, but in the file <code>/opt/charon-agent/ssp-agent/ssp/vm.dat</code>. <p>Please note: Attempting to run Charon-SSP/4U+/4V+ on insufficient hardware will cause the instance to exit with an error message. The exact message depends on the host hardware, for example, “MMU module insertion failed, please check if VT-X is enabled in BIOS”, “The host CPU doesn’t support Intel VT-x / EPT”, or “The host CPU doesn't support VT-x/EPT or AMD-v/NPT”.</p> <p>Check the BIOS settings and the flags vmx and ept (or svm and npt for AMD) in the output of cat /proc/cpuinfo or the flags line in the CPU tab of System info (Baremetal product) to verify hardware support.</p> |
| DIT Optimization | <p>This option controls the Dynamic Instruction Translation (DIT). DIT is a just in time compilation technology to dynamically optimize the SPARC instruction execution on x86-64 platforms. It can be set to OFF, Client JIT, or Server JIT (Charon-4U/4V only).</p> |
| | Client JIT parameters: |
| DIT page size | <p>This option controls the size of the translation buffer holding the translated binary code that results from the DIT optimization. It can be increased to a maximum of 2048KB. This parameter should only be changed if the log file indicates that the DIT optimization was disabled because the translation buffer size was too small. This option is not available on Charon-4M.</p> |
| FP boost ratio | <p>Defines the level of floating-point optimization. The parameter can be set to a value from 0 to 100. The default is 0 (= no boost). Most floating-point applications will profit from increasing this ratio. However, some applications may not be compatible with the optimization, resulting in degraded performance. So testing is required. This option is not available on Charon-4M.</p> |

7.5.3.3.2 Server JIT

This section describes the configuration options available in Server JIT mode (**Charon-4U/4V only**).

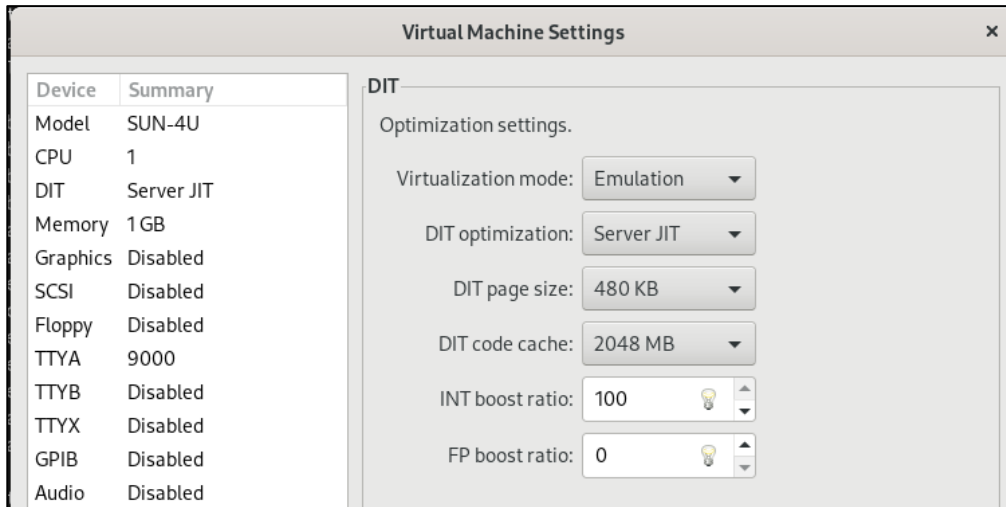


Figure 29: Charon-SSP Server JIT configuration window

The following table lists each of the **fields available for DIT Server JIT mode** and describes their operation.

| Field | Description |
|----------------------------|---|
| Virtualization Mode | <p>Possible values:</p> <ul style="list-style-type: none"> Emulation: selects the Charon-SSP/4U or Charon-SSP/4V emulator, Intel VT-x/EPT: selects Charon-SSP/4U+ or Charon-SSP/4V+ depending on the base hardware family selected. Also valid for AMD-v/NPT. This option is inactive on Charon-SSP/4M or if Charon-SSP/4U+/4V+ is not installed. This setting is not stored in the emulator configuration file, but in the file <code>/opt/charon-agent/ssp-agent/ssp/vm.dat</code>. <p>Please note: Attempting to run Charon-SSP/4U+ on insufficient hardware will cause the instance to exit with an error message (depending on host hardware, for example, “MMU module insertion failed, please check if VT-X is enabled in BIOS”, “The host CPU doesn’t support Intel VT-x / EPT”), or “The host CPU doesn't support VT-x/EPT or AMD-v/NPT”.</p> <p>Check the BIOS settings and the flags vmx and ept in the output of <code>cat /proc/cpuinfo</code>, or the flags line in the CPU tab of System info (Baremetal product) to verify hardware support.</p> |
| DIT Optimization | <p>This option controls the Dynamic Instruction Translation (DIT). DIT is a just in time compilation technology to dynamically optimize the SPARC instruction execution on x86-64 platforms. It can be set to OFF, Client JIT, or Server JIT (Charon-4U/4V only).</p> |
| | Server JIT parameters (not available on Charon-4M): |
| DIT page size | <p>This option controls the size of the translation buffer holding the translated binary code that results from the DIT optimization. It can be increased to a maximum of 2048KB. This parameter should only be changed if the log file indicates that the DIT optimization was disabled because the translation buffer size was too small.</p> |
| DIT code cache | <p>Size of cache between 1024MB and 8192MB in steps of 1024MB.</p> |
| FP boost ratio | <p>Defines the level of floating-point optimization. The parameter can be set to a value from 0 to 100. The default is 0 (= no boost). Most floating-point applications will profit from increasing this ratio. However, some applications may not be compatible with the optimization, resulting in degraded performance. So testing is required.</p> |
| INT boost ratio | <p>Defines the level of integer operation optimization. The parameter can be set to a value from 0 to 100. The default is 100 (= maximum boost). The higher the value the more resources are required. Hence high values are likely to provide most benefit if the guest system applications run for a long time.</p> |

7.5.3.4 Memory Configuration

To view or change the current memory configuration, select **Memory** in the **Device** column on the left.

The images below show examples of the memory configuration window:

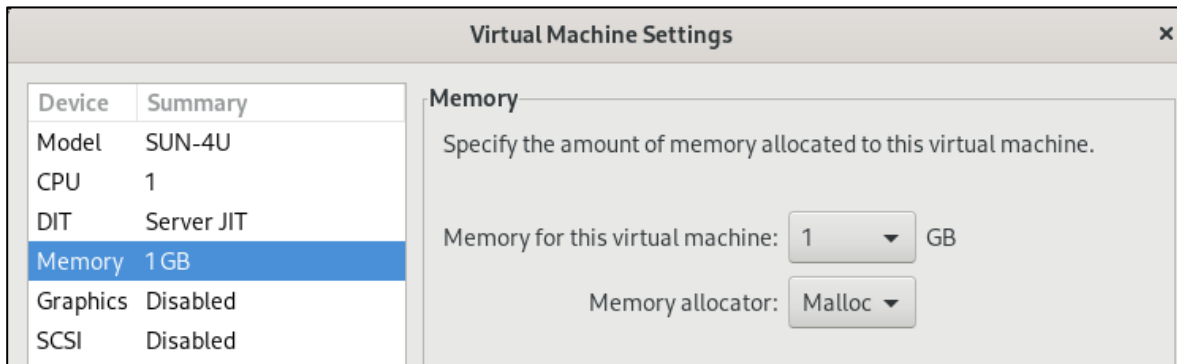


Figure 30: Charon-SSP/4U memory options

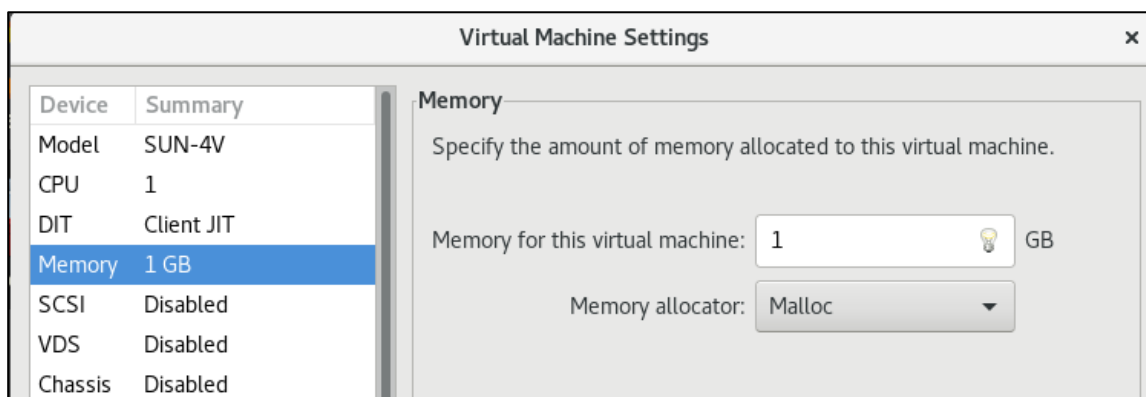


Figure 31: Charon-SSP/4V memory options (no dropdown menu)

The following table lists each of the **fields in the memory configuration window** and describes their effect.

| Field | Description |
|--|---|
| Memory for this virtual machine | <p>Set the amount of RAM allocated to the virtual SPARC machine. Memory must be allocated in certain increments. The allocation rules for each virtual machine family are as follows:</p> <ul style="list-style-type: none"> SUN-4M: 64MB, 128MB, 256MB and 512MB SUN-4U: 1 to 128GB in 1GB increments SUN-4V: 1 to 1024GB in 1GB increments (not a drop-down list but manual entry). Actual limits are different depending on guest OS: Solaris 10: 1TB, Solaris 11: 512 GB. The GUI allows larger values, but this is reserved for future use. <p>Please note that Solaris 11 requires a minimum amount of memory to boot (the real memory requirements usually are significantly higher):</p> <ul style="list-style-type: none"> Solaris 11.1: 2GB Solaris 11.2: 2GB Solaris 11.3: 2GB Solaris 11.4: 4GB |

| Field | Description |
|-------------------------|--|
| Memory allocator | This option specifies the memory allocation method used for the virtual machine. The default is malloc . It is appropriate for most cases. Please contact Stromasys if your environment has special memory requirements. Options: <ul style="list-style-type: none"> • Malloc: all virtual machine RAM is allocated from system heap. • Mmap: all virtual machine RAM is allocated from file backed virtual memory by memory mapping. |

7.5.3.5 Graphics Configuration

This configuration option is not applicable to SUN-4V(+).

Please note:

- The **graphical performance** depends on many parameters, for example, the performance of host system, emulated system, and network. A test is recommended.
- If the **integrated SSH tunnel of the Charon-SSP Manager** is used, the ports used for mouse and keyboard events are redirected through the tunnel. The remote port (graphics data) is not redirected. Therefore, in such situations, firewalls must allow the port. If a VPN connection is used to communicate with the Charon host and guest, all connections can be routed through the VPN. When running traffic over a public network, the use of an encrypted VPN connection is highly recommended.

To view or change the current graphics emulation configuration, select **Graphics** in the left-hand pane of the Settings window. This opens the graphics configuration window. As shown below, the graphics emulation is disabled by default.

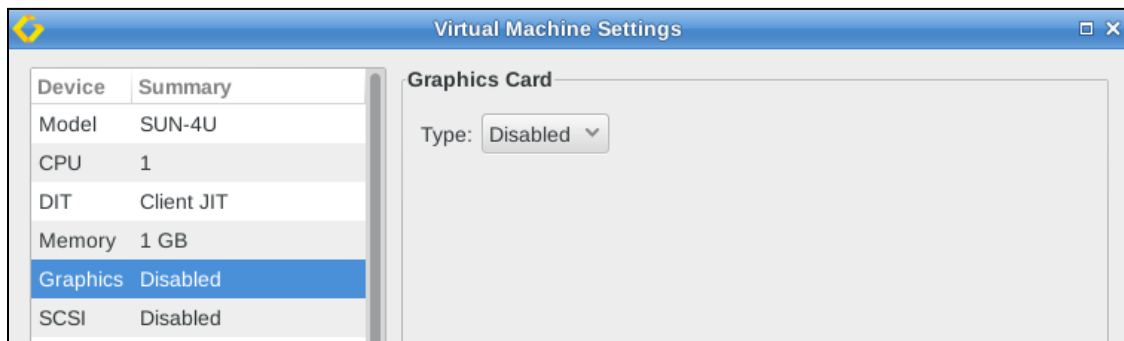


Figure 32: Graphics emulation disabled by default

To enable it, select a graphics card type from the drop-down menu. Possible values are

- CGSIX or CGTHREE on Charon-SSP/4M (CGSIX emulation is not supported for SunOS 4.x guest systems)
- CGSIX or Rage XL on Charon-SSP/4U(+)

The CGTHREE adapter is a graphic adapter with frame buffer; the CGSIX adapter is a graphic adapter with frame buffer and 2D acceleration, the Rage XL is a graphic adapter with frame buffer, 8MB Memory, and 2D acceleration.

The following image shows the options on a SUN-4U system:

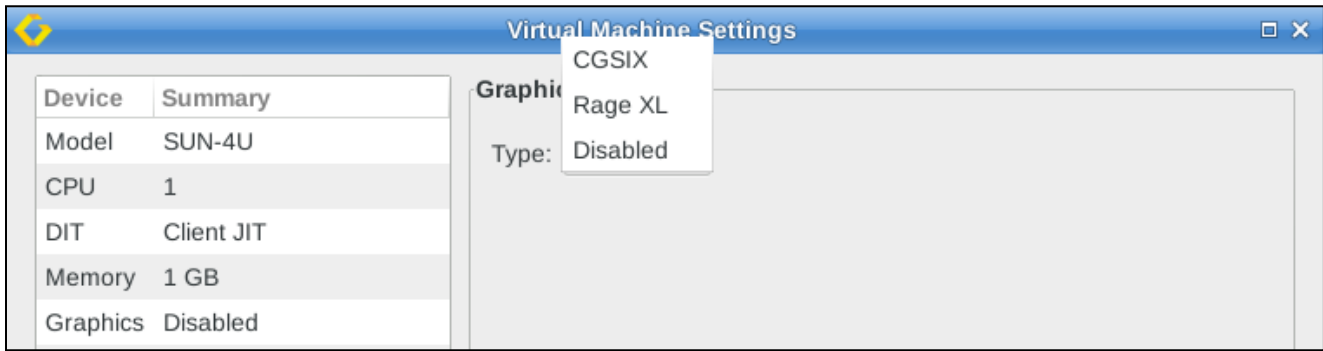


Figure 33: Graphics device selection on SUN-4U

To start configuring a graphics device, select one of the supported graphics options. This opens the configuration window as shown in the Charon-4U example below:

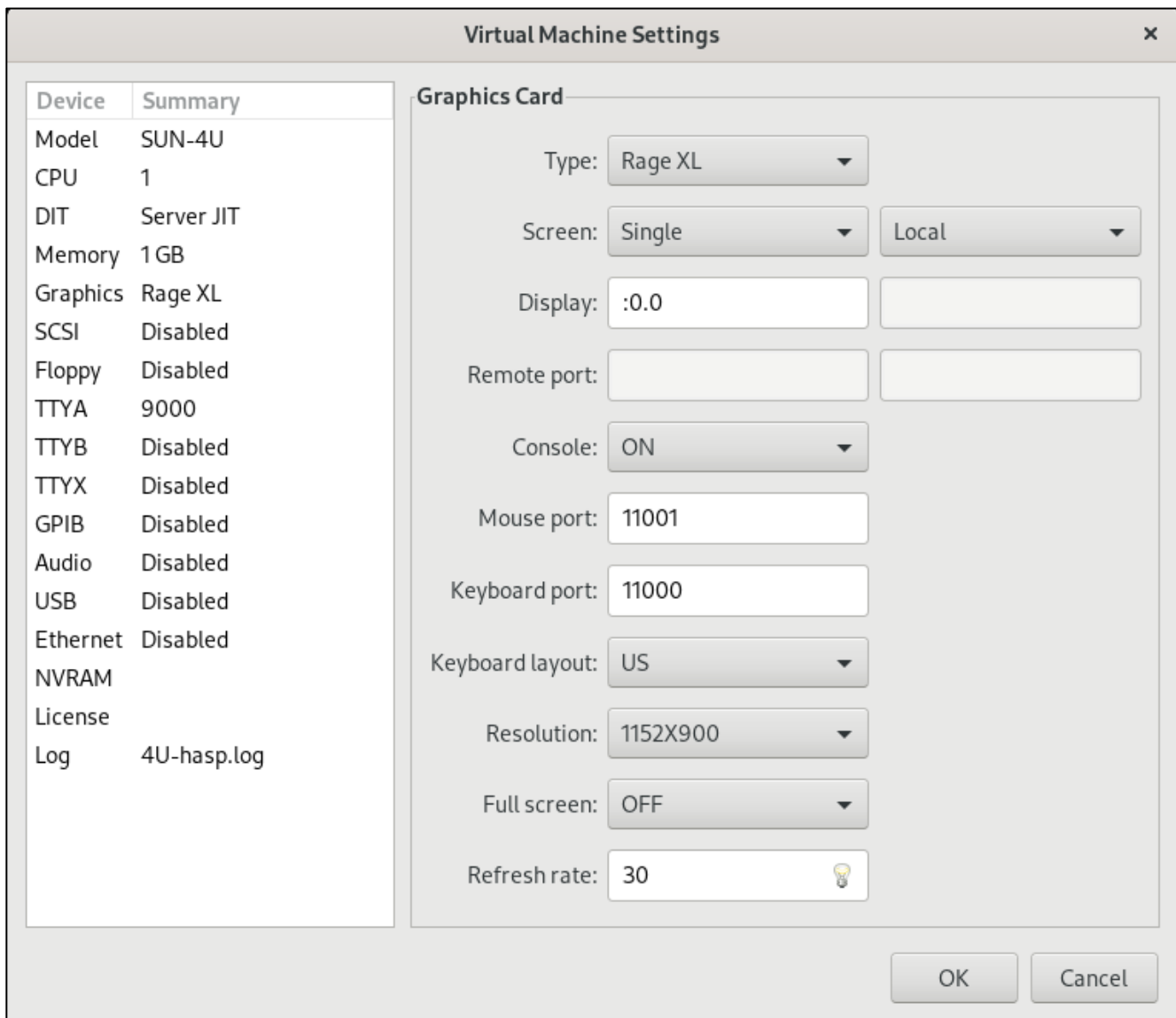


Figure 34: Graphics configuration window (4U Rage XL)

The following table describes the **graphics configuration options**:

| Field | Description |
|------------------------|---|
| Type | <p>Selection of supported graphics options:</p> <ul style="list-style-type: none"> • CGSIX or CGTHREE on Charon-SSP/4M (CGSIX emulation is not supported for SunOS 4.x guest systems) • CGSIX or Rage XL on Charon-SSP/4U(+) • Disabled |
| Screen | <p>Number of screens:</p> <ul style="list-style-type: none"> • Single: use one screen • Dual: use two screens <p>Location for displaying the graphics output:</p> <ul style="list-style-type: none"> • Local: display graphics output only on the local system (disabled on cloud installations) • Remote: graphics output can be displayed on the local system or a remote system using the Charon Manager (starting with 5.3.9 for Charon Manager on Linux and version 5.3.11 for Charon Manager on Windows). This option requires that the Charon Manager be installed on the remote system. |
| Display | <p>Defines the DISPLAY variable to be used by the graphics output. The default value is “:0.0” (display 0 screen 0). This value must be set to match the display configuration of the system on which the graphics output is to be displayed. Use the command echo \$DISPLAY on the target Linux system to identify the correct value.</p> <p>If a dual screen configuration is selected, two display variables can be defined.</p> |
| Remote port | <p>Defines the port(s) to which a Charon-SSP Manager on a remote system connects to display the graphics output of the guest system. The default value is 11100 for a single screen configuration. For a dual screen configuration, the default ports are 11100 and 11101. Only relevant for remote screen configurations. The ports must be unique on the host system. See also the note about the Charon Manager integrated SSH tunnel at the beginning of this section.</p> |
| Console | <p>Defines whether the graphical device should act as the console of the guest system.</p> <ul style="list-style-type: none"> • ON: the graphics device is the system console of the guest system (default). In this case, the serial console window in Charon-SSP Manager is not available. • OFF: the serial console in Charon-SSP Manager or an external serial console is used. |
| Mouse port | <p>Port for transmitting mouse event data. Default 11001. The port must be unique on the host system.</p> |
| Keyboard port | <p>Port for transmitting keyboard event data. Default 11000. The port must be unique on the host system.</p> |
| Keyboard layout | <p>The appropriate keyboard layout can be selected from the drop-down menu.</p> <p>The META key of the Solaris keyboard is mapped to the Windows key on the PC keyboard.</p> |
| Resolution | <p>The appropriate resolution can be selected from the drop-down menu.</p> <p>CG3 supports 800 x 600, 1024 x 768, and 1152 x 900; CG6 and Rage XL support 1024 x 768, 1152 x 900, 1280 x 1024, and 1600 x 1280.</p> |
| Full screen | <p>If set to ON, the emulated graphics device will start in full-screen mode. Best results are achieved if the resolution of the host system display matches the resolution of the emulated device. To toggle between full-screen and normal mode during operation use the key combination CTRL+SHIFT+F after clicking into the window to give it focus.</p> |
| Refresh rate | <p>The refresh rate for the graphical output can be set to a value between 20 and 100. Charon-SSP/4U(+) only.</p> |

Mouse and keyboard capture and release:

- When you click into the graphics device window, it will **capture mouse and keyboard**.
- To release mouse and keyboard press LEFT-CTRL+ESC.
If running Charon Manager on Windows, use LEFT-CTRL+ALT.

Use the toggle key combination (**CTRL+SHIFT+F**) to switch between normal window mode and full-screen mode (first click into the graphics window to make sure it has focus).

In addition to configuring the graphics emulation in the Charon-SSP Manager, there are several prerequisites as described in the following table:

| Where | Description |
|----------------------|--|
| Host system | <ul style="list-style-type: none"> • Ensure that the required ports for display, mouse, and keyboard events are not blocked by a firewall. • Only relevant for non-baremetal systems: For a local graphics device, ensure that the local root user can display X applications on the defined display. Verify this using the xhost command without parameters. If the root user does not have access, the local graphics display will not open and the Charon-SSP log will show repeatedly that a connection to the display is attempted and disconnected again. To add the permissions, use the command xhost SI:localuser:root. Note: starting with version 3.1.x, this setting is added automatically by the Charon Manager upon start (for the duration of the Linux login session). |
| Solaris guest | <ul style="list-style-type: none"> • Ensure that the required drivers (SUNWc6g*, SUNWdfb*, SUNWm64*) are installed on the system. They are part of the standard system and are normally installed if the matching devices are found. Should they be missing, the packages can be installed or the drivers can be copied from the installation CD (must be same version and patch level as the Solaris guest). The names of the drivers are <i>cgsix</i>, <i>cgthree</i>, and <i>m64</i>. • After configuring the graphical device or changing the configuration between single and dual screen configurations, restart the emulator and reboot the system with the boot <device> -r option to create the correct device special files and the <i>/dev/fb*</i> links that point to these devices. • If the Solaris graphical user interface is to be used on the device, ensure that <ul style="list-style-type: none"> ○ <i>/usr/openwin/bin</i> is in the path of the user, ○ dtlogin is enabled at system start (usually enabled by /usr/dt/bin/dtconfig -e) and running (process can be started with /etc/init.d/dtlogin start). • Ensure that the X-server uses the correct fb device (default <i>/dev/fb</i>). Otherwise, it may fail with the message that the device does not exist. In this case, perform the following steps: <ul style="list-style-type: none"> ○ Create the directory /etc/dt/config if it does not exist. ○ Copy /usr/dt/config/Xservers into /etc/dt/config or edit a pre-existing file in this directory. ○ Modify the X-server start line to contain the correct <i>/dev/fb*</i> line. You can find the existing framebuffer device links using ls -l /dev/fb*. Sample line in the Xservers file: :0 Local local_uid@console root /usr/openwin/bin/Xsun :0 -dev /dev/fb0 -nobanner If you use a dual monitor configuration, you must add a second -dev entry. On Solaris 10 the path for the Xserver is <i>/usr/X11/bin/Xserver</i>. |

The following image shows a dual screen setup on one physical screen (for illustration purposes only):

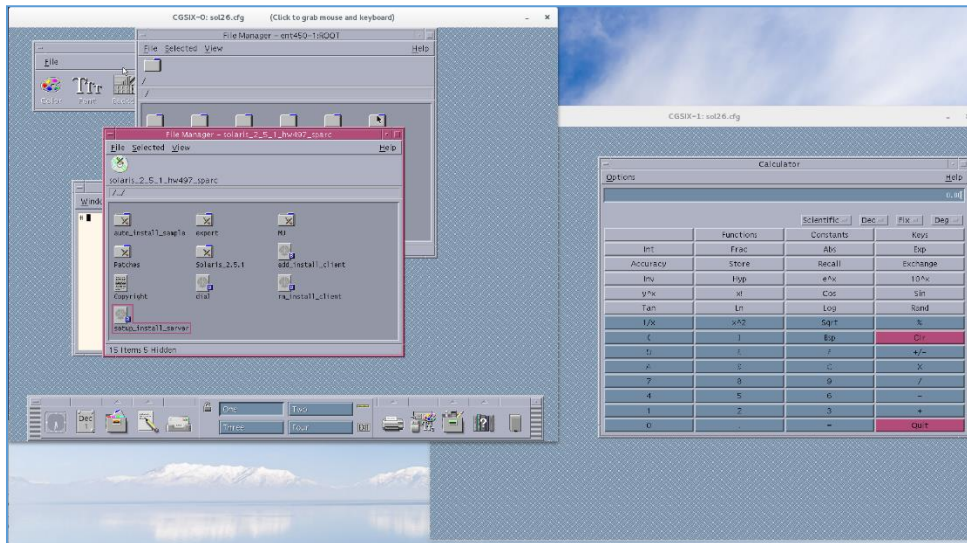


Figure 35: Dual screen display on one physical monitor

7.5.3.6 SCSI Storage Configuration

7.5.3.6.1 Drivers for Emulated SCSI Devices

The following SCSI device drivers are required by the guest operating system to operate the SCSI devices provided by the Charon-SSP emulator. If the physical Solaris system that is to be migrated does not have these drivers, they must be copied from another Solaris system or an installation ISO of the same version.

- `sd / st` (SCSI disk/tape)
- `glm` (SCSI HBA driver)

Important notes for Solaris 11.4:

Solaris 11.4 installation media no longer include the GLM driver. The installation ISO or CD-ROM must be configured as a VDS device (with ID 6). To enable the use of the SCSI devices provided by Charon-SSP to a Solaris 11.4 guest system, the required SCSI driver must be installed. For this, the GLM driver from Solaris 11.3 can be installed in Solaris 11.4.

Steps to install the driver:

1. Copy `glm` from Solaris 11.3 (`/kernel/drv/sparcv9/glm`) to the same path on Solaris 11.4.
2. On Solaris 11.4, run the following command:

```
# add_drv -m '* 0644 root sys' -c scsi glm (warning messages can be ignored)
```

or

reboot the system with the command: `boot <disk> -rv`

7.5.3.6.2 SCSI Configuration Window Overview

To view or change the current virtual machine SCSI configuration, select **SCSI** in the left-hand pane of the Settings window. This opens a SCSI configuration window like the one shown below.

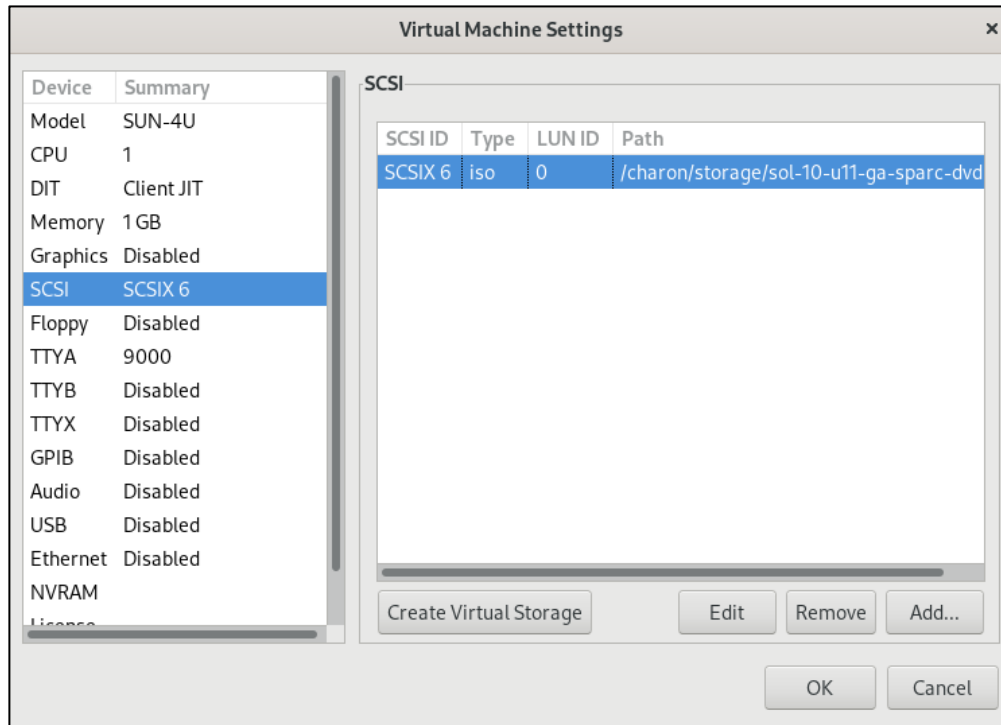


Figure 36: Virtual SCSI device configuration window

This window allows the following actions:

- Creating virtual disk and tape container files (**Create Virtual Storage** button). The **Create Virtual Storage** option is also available in the **Tools** menu of the Charon-SSP Manager. The functions provided are identical to the functions provided via the **Virtual Machine Settings** window shown above.
- Attaching virtual storage devices (both physical devices and container files) to the virtual machine (**Add** button).
- Editing or removing an existing virtual storage device. Select an existing device to make the **Edit** and **Remove** buttons available.

7.5.3.6.3 Creating a New Virtual Disk Container File

It is often convenient to use container files for virtual disk and tape devices. This section describes how to create disk container files. To create a virtual disk container file, **click** on **Create Virtual Storage** in the SCSI device **Virtual Machine Settings** window. This displays the **Create Virtual Storage** dialog opened on the virtual disk tab as shown below.

The screenshot shows the 'Create Virtual Storage' dialog with the following fields and values:

- Virtual disk type: Custom
- Virtual disk name: datadisk1.vdisk
- Location: storage
- Virtual disk geometry:
 - Block number: 10000000
 - Block size: 512 Bytes
 - Disk size: 5.1 GB/4.8 GiB
- Progress bar: 0%
- Buttons: Create, Close

Figure 37: Create new custom virtual disk

To create a virtual disk container file, follow the instructions listed below.

| Step | Description |
|------|--|
| 1 | <p>Select the virtual disk type from the drop-down list Virtual disk type.</p> <ul style="list-style-type: none"> If you select a preconfigured Virtual disk type the Block number field is updated to match that model. If you specify the type Custom, enter the container file size in 512-byte blocks at the field Block number. The size of the custom disk is shown in KB/KiB, MB/MiB, or GB/GiB depending on the configured number of blocks. Please note: currently, the maximum size of a disk presented to a Charon-SSP emulator is 2TB. |
| 2 | Specify a name for the virtual disk container file in the field Virtual disk name . |
| 3 | Select the location on the host filesystem for the container file by clicking on the Location selection button and selecting the correct path. The default is different depending on the Charon-SSP product. |
| 4 | Click on Create to create the virtual disk container file. Depending on the size of the container file, this may take some time. Close the window when done with creating container files. |

Before the disk can be used by the Solaris guest system, it must be added to the system configuration and formatted by the Solaris guest according to the customer specific requirements.

7.5.3.6.4 Creating a New Virtual Tape Container File

To create a virtual tape container file, click the **Create Virtual Storage** button in the SCSI device **Virtual Machine Settings** window. This opens the **Create Virtual Storage** window. Select the **Virtual Tape** tab.

To create a virtual tape container, follow the instructions below:

| Step | Description |
|------|--|
| 1 | Specify a name for the virtual tape container file in the field Virtual tape name . |
| 2 | Select the location on the host filesystem for the container file by clicking on the Location selection button and selecting the correct path. |
| 3 | Specify a size for the virtual tape file in megabytes (MB) in the field Tape size . The vtape file will expand automatically if more space is needed while writing to the tape. |
| 4 | Click on Create to create the virtual tape container file. Depending on the size of the container file, this may take some time. |

Using a virtual tape:

Once a virtual tape device has been created, it can be added to the Charon-SSP configuration and used by the Solaris guest system.

To simulate “swapping a tape” during guest system operation, the following steps are required:

| Step | Where | Description |
|------|-------|---|
| 1 | Guest | Rewind tape if required (will lead to overwriting existing information), write content to it, and “eject” it: <pre># mt -f <device-name> rewind # tar -cvf <device-name> <files-to-save> # mt -f <device-name> offline</pre> |
| 2 | Host | Move virtual tape container file to another name/location and create an empty new container with the original name. <pre># mv <vtape-container-name> <vtape-container-archived> # touch <vtape-container-name></pre> |
| 3 | Guest | Display tape status (thereby loading the new file), rewind tape if required, and write content to it: <pre># mt -f <device-name> status # mt -f <device-name> rewind # tar -cvf <device-name> <more-files-to-save></pre> |

Solaris tape device names have the format `/dev/rmt/<device>` where device can be a digit (e.g., `/dev/rmt/0`) or a combination of digits and certain letters (e.g., `/dev/mnt/0n` is the first drive set to no rewind).

Should the devices not exist after adding a virtual tape drive, boot the emulated SPARC guest system with the `-r` (reconfigure) parameter. Example: `boot disk0 -r`.

7.5.3.6.5 Adding or Editing a Virtual SCSI Device

To **add** a new virtual SCSI device, **click** on the **Add** button.

To **modify** an existing virtual SCSI device, select it from the list of configured devices and **click** on the **Edit** button. The **Edit** button appears when an existing virtual SCSI device is selected.

In both cases, a window like the one below opens with the configuration parameters of the virtual SCSI device.

Figure 38: Add a virtual SCSI device (Charon-SSP/4U example)

► Important note about SCSI bus and target ID configuration for emulated SCSI devices:

Charon-SSP does not place any restrictions on the SCSI bus and target ID configured for emulated SCSI devices, e.g., a virtual CD-ROM. However:

- Charon-SSP/4M normally expects the boot CD-ROM device to have SCSI ID 6 / LUN 0.
- Charon-SSP/4U normally expects the boot CD-ROM device to be on the external bus and SCSI ID 6 / LUN 0.
- Charon-SSP/4V normally expects the boot CD-ROM device on the internal (primary) bus and SCSI ID 6 / LUN 0.

If you encounter the problem that the boot CD-ROM is not found when trying to boot from it, verify its expected location in the OBP environment (using the `devalias` command).

Please refer to the VDS section for Solaris 11.4 boot device configuration.

The following table lists the fields in the **Add/Edit SCSI Device** configuration window and describes their possible values and meaning.

| Field | Description |
|------------------|---|
| SCSI bus | Specify either the Primary SCSI Bus or the External SCSI Bus . The External SCSI Bus is not supported on Charon-SSP/4M. |
| SCSI ID | SCSI target identification number. <ul style="list-style-type: none"> • Charon-SSP/4M: possible values are a 3-bit narrow SCSI target IDs between 0 and 7. • Charon-SSP/4U/4V: possible values are a 4-bit wide SCSI target IDs between 0 and 15. <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px;"> <p>The SCSI target ID 7 is reserved for the SCSI host bus adapter. It cannot be used for a user-configurable SCSI device.</p> </div> |
| LUN ID | Logical Unit Number. A SCSI device is identified by a combination of bus, target ID (SCSI ID), and LUN ID. The LUN ID parameter must be configured to match the storage device configuration and the guest OS support. Valid IDs are 0 through 7. Default value is 0. <ul style="list-style-type: none"> • The LUN configuration must start from LUN 0 and be contiguous without any gaps. • The LUNs configured for one SCSI target ID must belong to the same virtual device type. Mixing device types leads to the error SCSI X is in use. • Even on Solaris systems that support disks with non-zero LUN IDs, an old sd driver may not recognize such disks without manual configuration. Please refer to section Disks with non-zero LUN ID below. |
| Removable | Default: OFF . If enabled, the emulator will start even if the device/file does not exist on the host. |
| SCSI device type | Drop-down list of configurable device types. The list below shows the device types that are available in cloud and on-premises installations: <ul style="list-style-type: none"> • Virtual Disk: Virtual disk device backed by a container file (*.vdisk) • Virtual CDROM: Virtual CD-ROM device backed by a container file (*.iso) • Virtual Tape: Virtual tape device backed by a container file (*.vtape) • Physical Disk: Virtual disk device mapped to a physical disk or a physical disk partition on the host system <p>The following device types are only supported in conventional and Baremetal non-cloud installations:</p> <ul style="list-style-type: none"> • Physical CDROM: Virtual CD-ROM connected to a host-attached physical optical drive • Physical Tape: Virtual tape device connected to host attached physical tape drive. • Generic Device: Generic SCSI device. Useful to connect special SCSI peripherals, such as tape robots or SCSI-connected serial devices and scanners (testing of specific device is always required). The actual device type of a generic device (e.g., disk or tape) can be identified with the command: <code>lsscsi -g</code> If the command does not exist, use <code>yum install lsscsi</code> to install the package from your standard repository. |

| Field (cont'd) | Description |
|------------------|---|
| SCSI device path | <p>Click on the SCSI device path button to configure the mapping of the virtual SCSI device. This will open a file browser. To sort the file browser display by name, click on the corresponding heading.</p> <p>Select the correct device or file using the file browser, or type the correct name in the file name field.</p> <p>The list below shows sample device paths for each SCSI device type option.</p> <ul style="list-style-type: none"> • Virtual Disk: /usr/local/vm/scsi0.vdisk • Virtual CDROM: /usr/local/share/iso/sunos_4.1.4.iso • Virtual Tape: /usr/local/vm/scsi1.vtape • Physical Disk: /dev/sda, or /dev/disk/by-uuid/31fa8e8c-a6c0-45f7-9892-da13ba81e0e5 <p>It is strongly recommended to use a persistent device name from</p> <ul style="list-style-type: none"> • /dev/disk/by-id, or • /dev/disk/by-uuid <p>instead of a non-persistent /dev/sdX device name.</p> <ul style="list-style-type: none"> • Physical CDROM: /dev/sr1 • Physical Tape: /dev/st0 • Generic Device: /dev/sg0 <p>Note: if you manually enter a device name instead of selecting a device from the file browser window, make sure that the file/device exists (relative to the path of the opened file browser) or is set to removable.</p> |

7.5.3.6.5.1 Disks with non-zero LUN ID

Even if Solaris itself supports disks with non-zero LUN IDs (for example, Solaris 10 and 11), it is possible that old **sd** SCSI drivers do not recognize such disks without manual configuration.

In such cases, the following steps can be applied:

1. Verify that the non-zero disk is visible on the OBP console using the `probe-scsi` command. If the disk is not visible, this indicates a different problem.
2. Boot the Solaris guest system (if the system is unbootable, the recovery steps are outside the scope of this section).
3. Use the `format` command to check which disks are visible (leave the format command without performing any actions).
4. If the disks with the non-zero LUN IDs are **not visible**, perform the following steps:
 - a. Go to the kernel driver directory: `# cd /kernel/drv`
 - b. Make a backup copy of the file `sd.conf`.
 - c. Open the file `sd.conf` in a text editor.
 - d. For each of the missing disks with a non-zero LUN ID, copy the lines of the LUN ID = 0 entry of the relevant SCSI target ID.
 - e. Modify the copied lines to point to the correct non-zero LUN ID as shown in the example below:

```
# cat /kernel/drv/sd.conf
#
# Copyright 2009 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)sd.conf      1.11      09/04/15 SMI"

name="sd" class="scsi"
        target=0 lun=0;

# new entry for disk with target-ID=0, LUN-ID=1
name="sd" class="scsi"
        target=0 lun=1
...

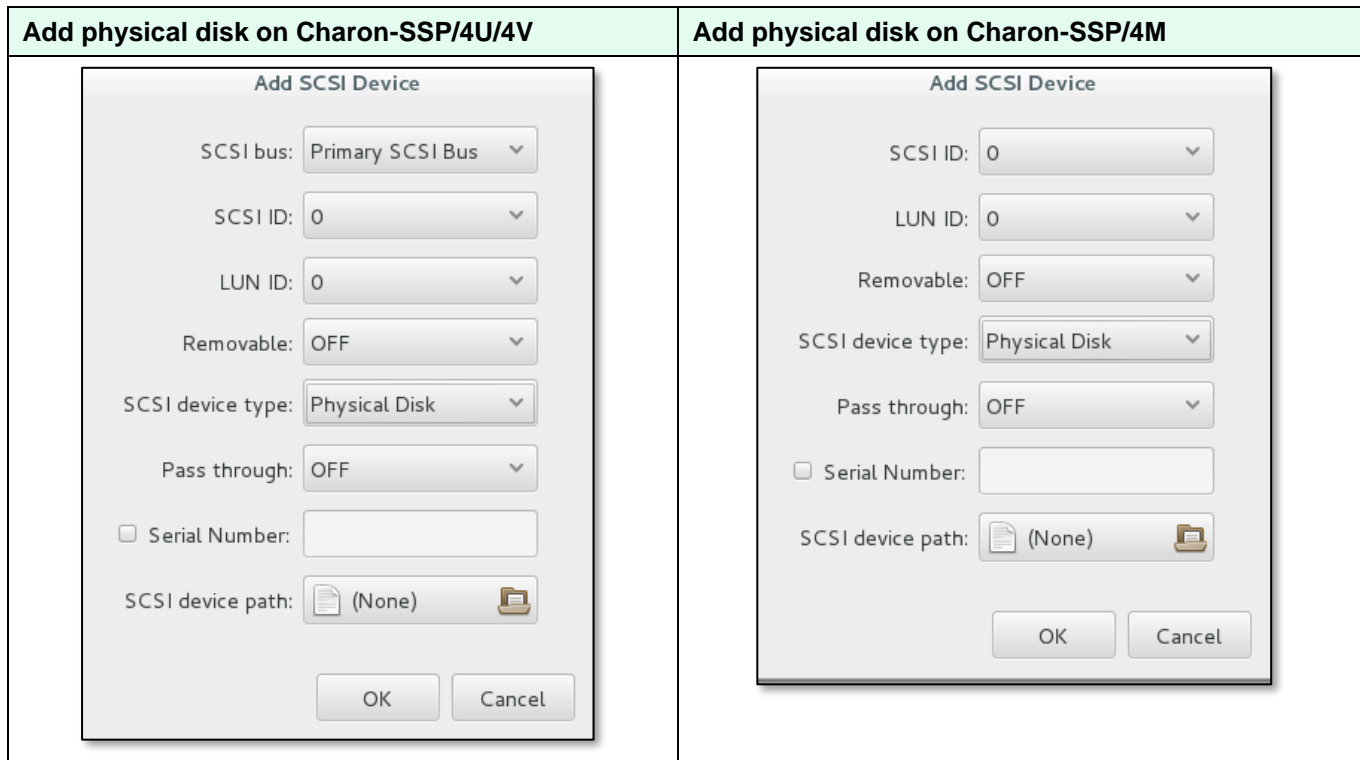
```

- f. Save the file.
- g. Reboot the system with the `-x` switch.
- h. Verify that your disks are visible.

7.5.3.6.6 Physical Disk Parameters on Charon-SSP

The Charon-SSP virtual machines offer additional options when adding physical disks as virtual SCSI devices. The windows for adding and editing devices contain the same fields. The configuration windows are different for Charon-SSP/4U/4V and Charon-SSP/4M because only Charon-SSP/4U and Charon-SSP/4V support a second SCSI bus.

The two different **configuration windows for physical disks** are shown below:



The following table describes the **additional parameters available for physical disks** on Charon-SSP:

| Field | Description |
|----------------------|--|
| Pass through | You can select OFF (default) or ON . SCSI pass-through is used to pass SCSI commands directly to a SCSI disk device. Such devices can be locally or remotely connected SCSI disks (e.g., local disks, iSCSI connected disks, Fibre Channel disks). On the host side, this feature depends on the generic SCSI driver (SG) capabilities of the host operating system. The emulator does not depend on special adapter types. This feature is useful, for example, for using shared disks in cluster environments (fencing / persistent reservations). Also, with SCSI pass-through, the data bypasses the Linux I/O cache layer. Read/write operations go directly to the storage controller. |
| Serial Number | <p>The serial number is a physical characteristic of hard disks and can be used to identify disks persistently and unambiguously. The Linux device file naming (i.e., /dev/sdX) may change when the host system reboots. Therefore, it is advisable to use a persistent identification for physical disks. In addition to the serial number, the persistent name of a disk from /dev/disks/by-id or /dev/disks/by-uuid can also be used for this purpose (as the SCSI device path).</p> <p>If the Serial Number field is enabled, the field SCSI device path is disabled.</p> <p>Identification of the serial number:</p> <ul style="list-style-type: none"> Use the following command to locate the serial number of a disk (either ID_SERIAL_SHORT or ID SCSI_SERIAL can be used): <pre># udevadm info -q property -n /dev/sdc grep SERIAL</pre> (e.g., device /dev/sdc) On Baremetal or cloud-specific Charon-SSP AL systems, you can also find the serial number using the Storage Manager started from the Tools > Charon Baremetal or Tools > <cloudname> Cloud menu of the Charon-SSP Manager. |

7.5.3.6.7 Removing a SCSI Storage Device

To remove a SCSI storage device, perform the following steps:

- **Select** the device in the Virtual Machine Settings **SCSI configuration** window.
- Then **click** the **Remove** button.

The device is removed immediately from the configuration. The Charon-SSP Manager does not ask for confirmation.

If the virtual SCSI storage device is attached to a container file, the file itself is not removed with the configuration.

7.5.3.7 Configuring VDS Storage Devices

Please note: this feature is only supported for Charon-SSP/4V. Guest systems starting with Solaris 10-U6 can use VDS and/or SCSI storage devices.

The VDS configuration allows adding CD-ROM and/or disks to an emulator configuration that correspond to disks connected to a VDS (Virtual Disk Server) on a physical system. The devices are addressed as **vdisk** and **vcddrom** on the OBP console. To view or change the current virtual machine VDS disk configuration, select **VDS** in the left-hand pane of the Settings window. This opens a VDS configuration window like the one shown below:

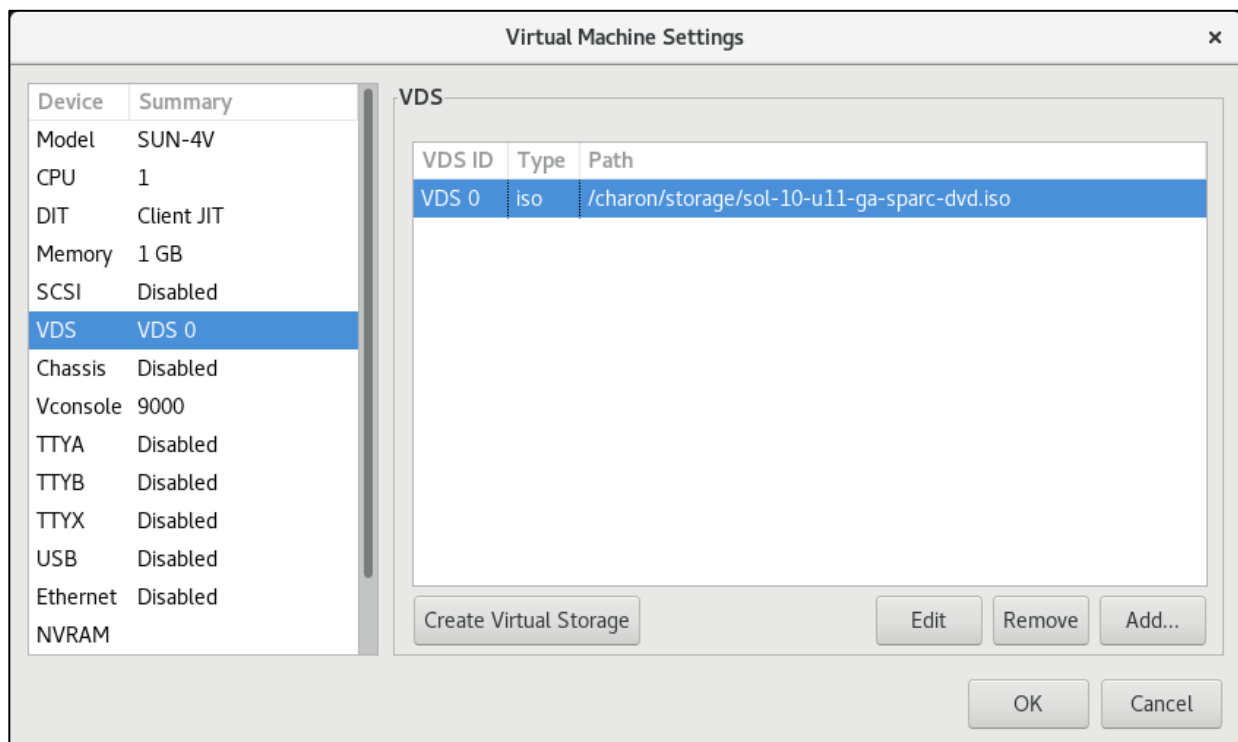


Figure 39: VDS configuration window

This window allows the following actions:

- **Creating** virtual disk files (**Create Virtual Storage** button). The **Create Virtual Storage** option is also available in the **Tools** menu of the Charon-SSP Manager. The functions provided are identical to the functions provided via the **Virtual Machine Settings** window shown above. Please refer to the corresponding sections in the [SCSI Storage Configuration](#) chapter.
- **Attaching** virtual storage devices (both physical devices and container files) to the virtual machine (**Add** button).
- **Editing** or removing an existing virtual storage device. Select an existing device to make the **Edit** and **Remove** buttons available.

7.5.3.7.1 Adding or Editing an VDS Device

To **add** a new VDS device, **click** on the **Add** button.

To **modify** an existing VDS device, select it from the list of configured devices and **click** on the **Edit** button. The **Edit** button appears when an existing device is selected.

In both cases, a window like the one below opens with the configuration parameters of the VDS device.

The dialog box titled "Add VDS Device" contains the following fields and controls:

- VDS ID: 0 (dropdown menu)
- VDS device type: Virtual Disk (dropdown menu)
- VDS device path: (None) (text field with a file icon)
- Write through: OFF (dropdown menu)
- OK and Cancel buttons at the bottom.

Figure 40: Add an VDS device

The dialog box titled "Add VDS Device" contains the following fields and controls:

- VDS ID: 0 (dropdown menu)
- VDS device type: Physical Disk (dropdown menu)
- Pass through: OFF (dropdown menu)
- Serial Number: (text field)
- VDS device path: (None) (text field with a file icon)
- Write through: OFF (dropdown menu)
- OK and Cancel buttons at the bottom.

Figure 41: VDS physical disk parameters

► Important note about the VDS ID:

Charon-SSP/4V expects the boot CD-ROM device on the on VDS ID 6.

This device can be booted with the command: `boot vcdrom`.

The following table describes the fields in the **Add/Edit VDS Device** configuration window:

| Field | Description |
|------------------------|--|
| VDS ID | VDS device identification number. Valid range 0 to 119. |
| VDS device type | Drop-down list of configurable device types: <ul style="list-style-type: none"> • Virtual Disk: Virtual disk device backed by a container file (*.vdisk) • Virtual CDROM: Virtual CD-ROM device backed by a container file (*.iso) • Physical Disk: Virtual disk device mapped to a physical disk or a physical disk partition on the host system • Physical CDROM: Virtual CD-ROM connected to a physical optical drive on the host system (not available in cloud environments) |
| VDS device path | <p>Click on the VDS device path button to configure the mapping of the VDS device. This will open a file browser. To sort the display by name, click on the corresponding heading.</p> <p>Select the correct device or file using the file browser, or type the correct name in the file name field.</p> <p>The list below shows sample device paths for each VDS device type option.</p> <ul style="list-style-type: none"> • Virtual Disk: /usr/local/vm/data1.vdisk • Virtual CDROM: /usr/local/share/iso/solaris-11-4.iso • Physical Disk: /dev/sda, or /dev/disk/by-uuid/31fa8e8c-a6c0-45f7-9892-da13ba81e0e5 It is strongly recommended to use a persistent device name from <ul style="list-style-type: none"> • /dev/disk/by-id, or • /dev/disk/by-uuid instead of a non-persistent /dev/sdX device name. • Physical CDROM: /dev/sr1 <p>Note: if you manually enter a device name instead of selecting a device in the file browser, make sure that the file/device exists (relative to the path of the opened file browser).</p> |
| Pass through | Physical disks only! You can select OFF (default) or ON . VDS pass-through is used to pass selected SCSI commands directly to a VDS disk device. This feature is useful, for example, for using shared disks in cluster environments (fencing / persistent reservations). Also, with SCSI pass-through, the data bypasses the Linux I/O cache layer. Read/write operations go directly to the storage controller. |

| Field | Description |
|----------------------|---|
| Serial Number | <p>Physical disks only!</p> <p>The serial number is a physical characteristic of hard disks and can be used to identify disks persistently and unambiguously. The Linux device file naming (i.e., /dev/sdX) is not persistent. Therefore, it is advisable to use a persistent identification for physical disks. In addition to the serial number, the persistent name of a disk from <code>/dev/disks/by-id</code> or <code>/dev/disks/by-uuid</code> can also be used for this purpose (as the device path). If the Serial Number field is enabled, the field SCSI device path is disabled.</p> <p>Identification of the serial number:</p> <ul style="list-style-type: none"> Use the following command to locate the serial number of a disk (either <code>ID_SERIAL_SHORT</code> or <code>ID_SCSI_SERIAL</code> can be used): <pre># udevadm info -q property -n /dev/sdc grep SERIAL</pre> (e.g., device /dev/sdc) On Baremetal or cloud-specific Charon-SSP AL systems, you can also find the serial number using the Storage Manager started from the Tools > Charon Baremetal or Tools > <cloudname> Cloud menu of the Charon-SSP Manager. |
| Write through | <p>Optionally enables write-through mode. That is, the data will be written directly to the Linux storage without using the buffer cache. The default is write-back mode, which will send a write success confirmation once the data has successfully been written to cache.</p> <p>Possible values:</p> <ul style="list-style-type: none"> OFF: write through is disabled (default value; write back) ON: write through is enabled |

7.5.3.7.2 Removing a VDS Storage Device

To remove a VDS storage device, perform the following steps:

- Select** the device in the Virtual Machine Settings **VDS configuration** window.
- Then **click** the **Remove** button.

The device is removed immediately from the configuration. The Charon-SSP Manager does not ask for confirmation.

If the virtual VDS storage device is attached to a container file, the file itself is not removed with the configuration.

7.5.3.8 Configuring Chassis Identification Parameters

Please note: this feature is only available on Charon-SSP/4V.

The Chassis configuration section allows the configuration of chassis identification parameters if required for the proper operation of the emulated system.

The image below shows a sample Chassis configuration window:

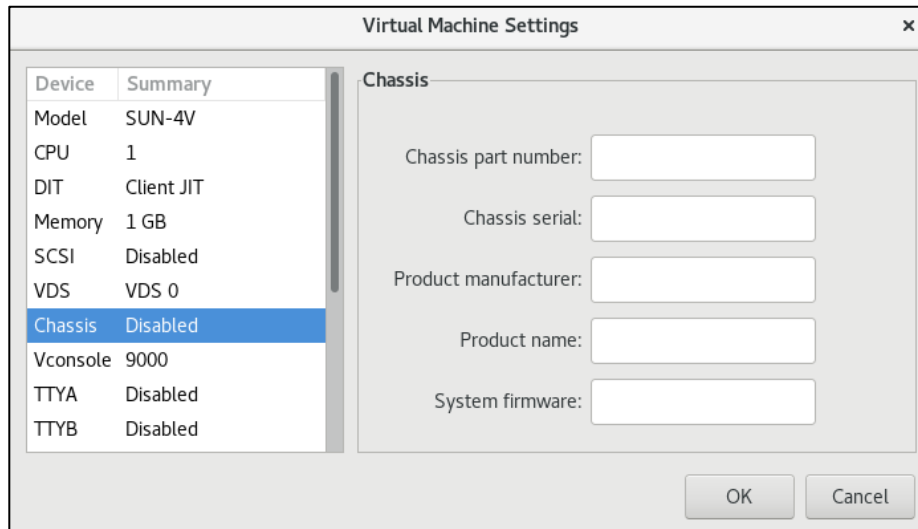


Figure 42: Charon-SSP/4V chassis configuration

The configurable parameters are:

- Chassis part number
- Chassis serial number
- Product manufacturer
- Product name
- System firmware

The configuration should match the parameters retrieved from the original system. You can use the following Solaris commands to retrieve the information:

- Chassis parameter: `ipmitool -I bmc fru`
- System firmware: `prtdiag -v | grep Firmware`

7.5.3.9 Configuring a Floppy Drive

Please note: this feature is only supported in conventional and Baremetal non-cloud installations of Charon-SSP/4M and Charon-SSP/4U (starting with version 5.3.1 of 4U).

A physical or a virtual floppy drive can be added to the configuration. The way to configure a virtual floppy is very similar to the virtual disk or virtual tape configuration described above (the virtual floppy is created via the **Create Virtual Storage** option).

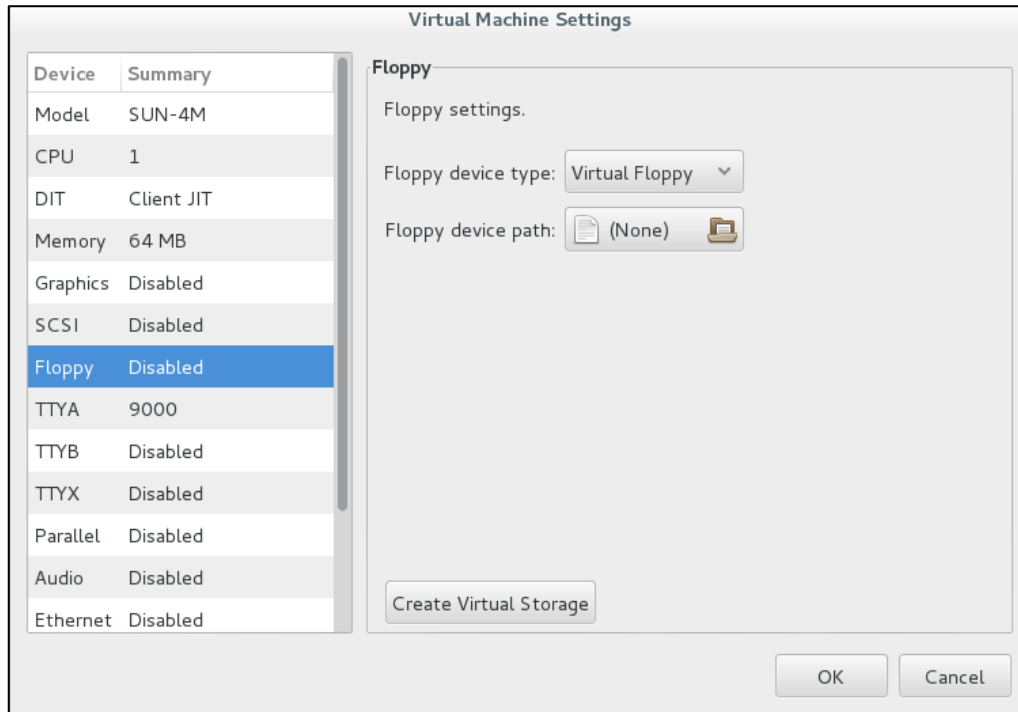


Figure 43: Floppy configuration screen

7.5.3.10 Vconsole Configuration (Charon-SSP/4V only)

The Vconsole device represents the serial console device of a Charon-SSP/4V instance. To view or change the current virtual machine console configuration, select **Vconsole** in the left-hand pane of the Settings window. This opens the **Vconsole** configuration window, shown below.

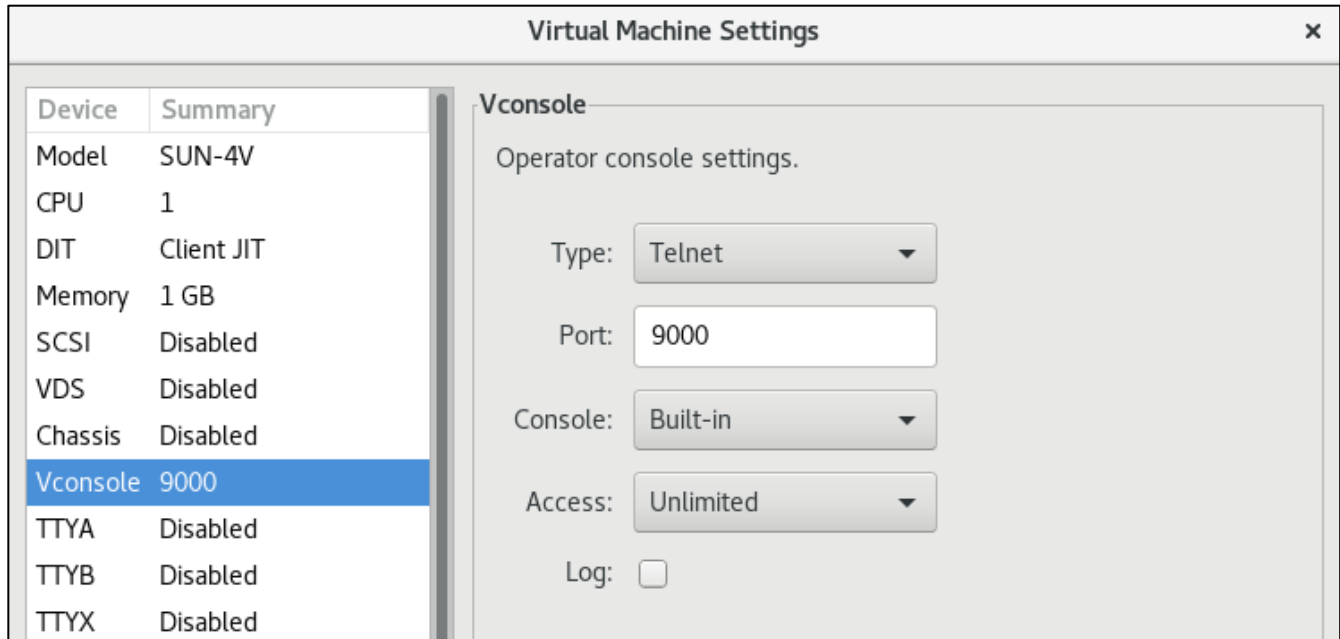


Figure 44: Vconsole configuration window

The **emulated console type** can have one of four values as described below.

Use the **Type** drop-down list to set the value.

- **TCP raw:** configure the console device as a network device (TCP socket) without any protocol enabled.
- **Telnet:** configure the console device as a network device (TCP socket) with the telnet protocol enabled.
- **Physical:** configure the console device as physical terminal directly attached to the host system.
- **Disabled:** disable the virtual console device entirely.

7.5.3.10.1 Vconsole Network Configuration

When configuring a network console device, the user can select one of two modes:

- **TCP raw** (serial line without protocol), or
- **Telnet** (serial line with telnet protocol support).

The image in the section above shows a sample of the Vconsole configuration for type telnet.

The following table lists all the **additional fields in the Vconsole network device configuration window** and describes their function:

| Fields | Description | | | | | |
|------------|--|--|--|------------|--|----------|
| Port | <p>This option specifies the TCP/IP port to use when listening for incoming console client connections. The ports configured for the network console and the serial ports must be unique on the Charon-SSP host system. Using a port that is already in use results in the following error messages in the virtual machine log file.</p> <pre>2019-08-27 09:54:03 ERROR SocketIO Failed to open socket server (port: 9000). 2019-08-27 09:54:03 ERROR Serial Fail to initialize serial device. 2019-08-27 09:54:03 ERROR VM Failed to initialize VCONSOLE</pre> <p>To access the console of a guest system across the network, make sure the port configured for the console is permitted by any intermediate firewalls. If using the Charon Manager with the embedded serial console, the connection can be routed across the Charon Manager SSH tunnel.</p> | | | | | |
| | <p>Specify in which way the network console will be viewed. Not visible if graphics device is configured with console enabled.</p> <table border="1"> <thead> <tr> <th>Console</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Built-in</td> <td>The built-in console is displayed and accessible from the Console tab in the Charon Manager. The console process listens on TCP port 9000 by default.</td> </tr> <tr> <td>External</td> <td>An external network console device allows an external client (e.g., telnet) to connect to the console port and interact with the virtual console device. If you choose this option, the console tab of the Charon-SSP Manager will not be shown for the system in question. By default, Charon-SSP will try to start a PuTTY window on the Charon host system – independent of where the Charon Manager runs – if it finds a valid DISPLAY variable. For more information, please refer to the section External Serial Console Considerations. The console process listens on TCP port 9000 by default.</td> </tr> </tbody> </table> | Console | Description | Built-in | The built-in console is displayed and accessible from the Console tab in the Charon Manager. The console process listens on TCP port 9000 by default. | External |
| Console | Description | | | | | |
| Built-in | The built-in console is displayed and accessible from the Console tab in the Charon Manager. The console process listens on TCP port 9000 by default. | | | | | |
| External | An external network console device allows an external client (e.g., telnet) to connect to the console port and interact with the virtual console device. If you choose this option, the console tab of the Charon-SSP Manager will not be shown for the system in question. By default, Charon-SSP will try to start a PuTTY window on the Charon host system – independent of where the Charon Manager runs – if it finds a valid DISPLAY variable. For more information, please refer to the section External Serial Console Considerations . The console process listens on TCP port 9000 by default. | | | | | |
| Access | <table border="1"> <tbody> <tr> <td>Unlimited</td> <td>Connection to the console is possible via a remote network connection.</td> </tr> <tr> <td>Local only</td> <td>Connection to the console is only possible from the local host.</td> </tr> </tbody> </table> | Unlimited | Connection to the console is possible via a remote network connection. | Local only | Connection to the console is only possible from the local host. | |
| | Unlimited | Connection to the console is possible via a remote network connection. | | | | |
| Local only | Connection to the console is only possible from the local host. | | | | | |
| Log | <p>When this box is checked, Charon-SSP writes a console log using the path <code>/opt/charon-agent/ssp-agent/ssp/sun-4v/<vm-name>/<vm-name>_vconsole.log</code>. The path can be changed in the configuration file.</p> <p>Note: the log file configured here is separate from the file the Charon-SSP Manager uses to cache the console output for the built-in serial console of the Charon-SSP Manager.</p> | | | | | |

7.5.3.10.2 Vconsole Physical Line Configuration

The image below shows the configuration window for a physical console device of a Charon-SSP/4V system.

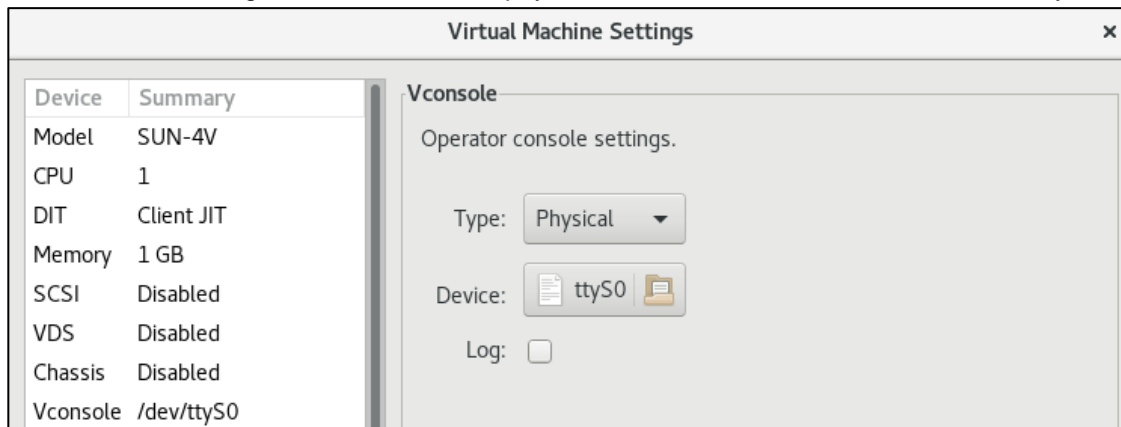


Figure 45: Vconsole physical serial line

Physical serial console configuration options:

- **Device:** opens a file browser to let the user select from the directly attached serial ports available on the host system (tty* devices).
- **Log:** used to enable and disable the console log as described in the network settings.

7.5.3.11 TTYA Configuration

To view or change the current virtual machine console configuration, select **TTYA** in the left-hand pane of the Settings window. This opens the **TTYA** configuration window, shown below. In this example, TTYA is disabled.

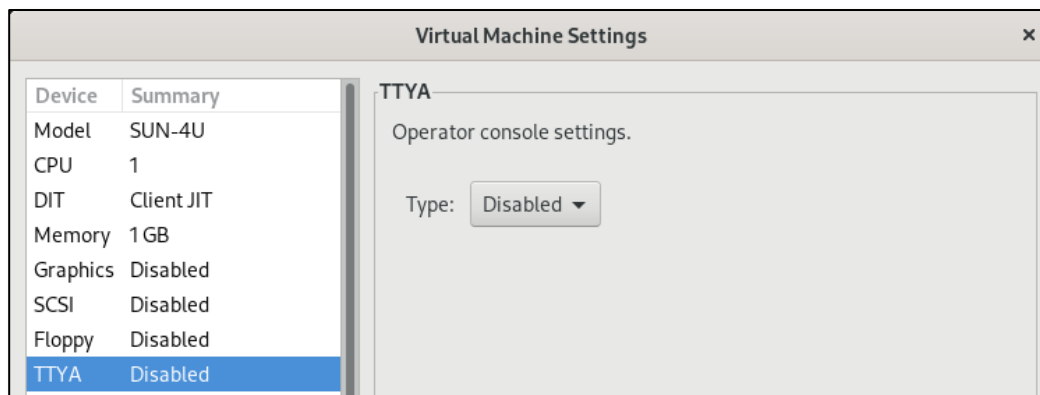


Figure 46: Virtual machine TTYA disabled

Note: on Charon-SSP/4U and Charon-SSP/4M, TTYA can be configured as the serial console or, if the graphical device is configured to be the system console, TTYA can be used as a normal serial line. On Charon-SSP/4V systems, it can only be used as a normal serial line.

The **virtual console type** can have one of four values as described below.

Use the **Type** drop-down list to set the value.

- **TCP raw:** configure the console device as a network device (TCP socket) without any protocol enabled.
- **Telnet:** configure the console device as a network device (TCP socket) with the telnet protocol enabled.
- **Physical:** configure the console device as physical terminal directly attached to the host system.
- **Disabled:** disable the virtual console device entirely.

The following sections describe the specific configuration details of physical and network consoles.

7.5.3.11.1 TTYA Physical Line Configuration

The image below shows the configuration window for a physical console device on Charon-SSP/4U.

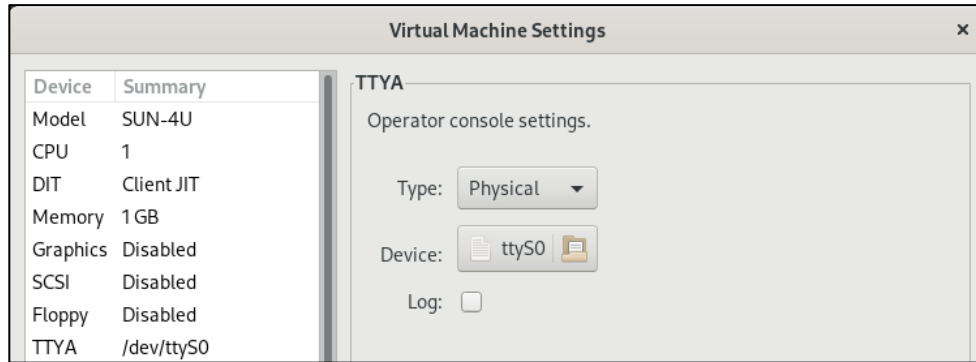


Figure 47: TTYA physical console device:

Physical console configuration options:

- **Device:** opens a file browser to let the user select from the directly attached serial ports available on the host system (*tty** devices).
- **Log:** used to enable and disable the console log (not available on Charon-SSP/4V). For details, see description in the next section.

7.5.3.11.2 TTYA Network Configuration

When configuring a network console device, the user can select one of two modes:

- **TCP raw** (serial line without protocol), or
- **Telnet** (serial line with telnet protocol support).

TTYA cannot be used as the system console for Charon-SSP/4V systems. Such systems must use the Vconsole device.

The image below shows a network console device with telnet protocol support on Charon-SSP/4U:

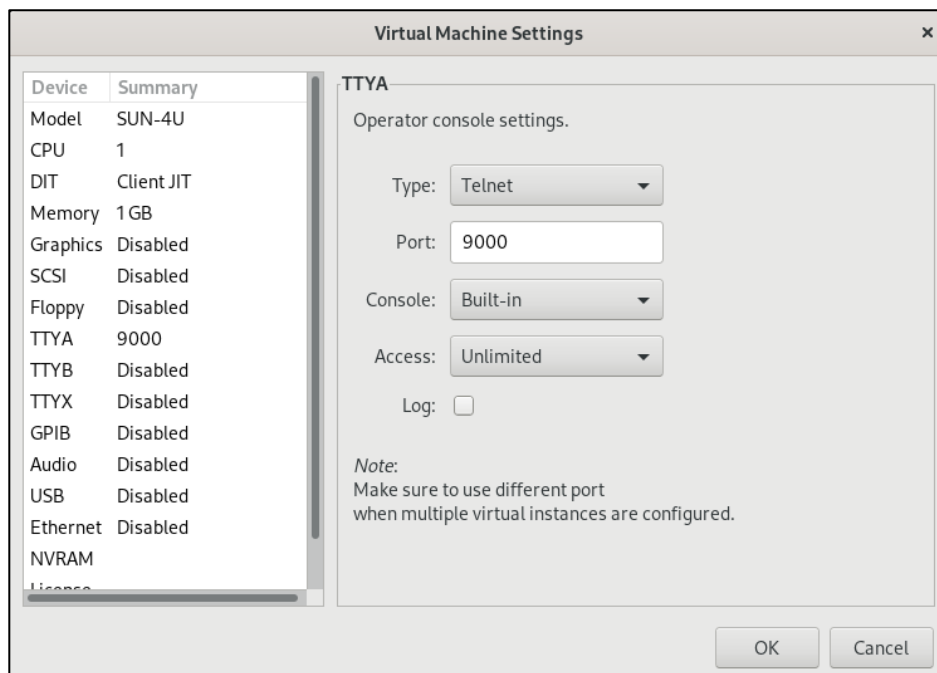


Figure 48 TTYA network console via TCP socket:

The following table lists all the **additional fields in the TTYA network device configuration window** and describes their function.

| Fields | Description | | | | | | |
|----------------|--|--|--|------------|--|----------|--|
| Port | <p>This option specifies the TCP/IP port to use when listening for incoming console client connections. The ports configured for the network console and the serial ports must be unique on the Charon-SSP host system. Using a port that is already in use results in the following error messages in the virtual machine log file.</p> <pre>2015-03-23 11:45:50 ERROR SocketIO Failed to open socket server! 2015-03-23 11:45:50 ERROR serial fail to init serial! 2015-03-23 11:45:50 ERROR vm Failed to initialize device:4</pre> <p>To access the console of a guest system across the network, make sure the port configured for the console is permitted by any intermediate firewalls. If using the Charon Manager with the embedded serial console, the connection can be routed across the Charon Manager SSH tunnel.</p> | | | | | | |
| Console | <p>Not applicable to Charon-SSP/4V. Please refer to the Vconsole section instead</p> <p>Specify in which way the network console will be viewed. Not visible if graphics device is configured with console enabled.</p> | | | | | | |
| | <table border="1"> <thead> <tr> <th>Console</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Built-in</td> <td>The built-in console is displayed and accessible from the Console tab in the Charon Manager. The console process listens on TCP port 9000 by default.</td> </tr> <tr> <td>External</td> <td>An external network console device allows an external client (e.g., telnet) to connect to the console port and interact with the virtual console device. If you choose this option, the console tab of the Charon-SSP Manager will not be shown for the system in question. By default, Charon-SSP will try to start a PuTTY window on the Charon host system – independent of where the Charon Manager runs – if it finds a valid DISPLAY variable. For more information, please refer to the section External Serial Console Considerations. The console process listens on TCP port 9000 by default.</td> </tr> </tbody> </table> | Console | Description | Built-in | The built-in console is displayed and accessible from the Console tab in the Charon Manager. The console process listens on TCP port 9000 by default. | External | An external network console device allows an external client (e.g., telnet) to connect to the console port and interact with the virtual console device. If you choose this option, the console tab of the Charon-SSP Manager will not be shown for the system in question. By default, Charon-SSP will try to start a PuTTY window on the Charon host system – independent of where the Charon Manager runs – if it finds a valid DISPLAY variable. For more information, please refer to the section External Serial Console Considerations . The console process listens on TCP port 9000 by default. |
| | Console | Description | | | | | |
| Built-in | The built-in console is displayed and accessible from the Console tab in the Charon Manager. The console process listens on TCP port 9000 by default. | | | | | | |
| External | An external network console device allows an external client (e.g., telnet) to connect to the console port and interact with the virtual console device. If you choose this option, the console tab of the Charon-SSP Manager will not be shown for the system in question. By default, Charon-SSP will try to start a PuTTY window on the Charon host system – independent of where the Charon Manager runs – if it finds a valid DISPLAY variable. For more information, please refer to the section External Serial Console Considerations . The console process listens on TCP port 9000 by default. | | | | | | |
| | | | | | | | |
| Access | <table border="1"> <tbody> <tr> <td>Unlimited</td> <td>Connection to the console is possible via a remote network connection.</td> </tr> <tr> <td>Local only</td> <td>Connection to the console is only possible from the local host.</td> </tr> </tbody> </table> | Unlimited | Connection to the console is possible via a remote network connection. | Local only | Connection to the console is only possible from the local host. | | |
| | Unlimited | Connection to the console is possible via a remote network connection. | | | | | |
| Local only | Connection to the console is only possible from the local host. | | | | | | |
| Log | <p>Not applicable to Charon-SSP/4V.</p> <p>When this box is checked, Charon-SSP writes a console log using the path below (sample for TTYA): <code>/opt/charon-agent/ssp-agent/ssp/[sun-4m sun-4u]/<vm-name>/<vm-name>_ttya.log</code>. The path can be changed in the configuration file.</p> <p>Note: the log file configured here is separate from the file the Charon-SSP Manager uses to cache the console output for the built-in serial console of the Charon-SSP Manager.</p> | | | | | | |

7.5.3.12 TTYB Configuration

To view or change the virtual machine TTYB configuration, select **TTYB** in the **Device** column of the left-hand pane. The virtual TTYB serial device can be configured as both a physical or network connected device. The configuration of this device is very similar to TTYA (without the console specific configuration). For further details related to configuring this device, consult the section *TTYA Configuration*.

7.5.3.13 TTYX Configuration

Please note: this feature is only supported in conventional and Baremetal non-cloud installations

The TTYX configuration provides additional asynchronous serial ports to the virtual SPARC system.

The additional serial port emulation in Charon-SSP provides different modes:

1. **For Charon-SSP 4U(+):**
 - On-board Mode, which extends the original number of two built-in serial ports by a maximum of 14 additional serial ports (On-board mode).
 - DIGI AccelePort 920 emulation for up to 4 boards with 8 ports each, i.e., 32 serial ports (DIGI AccelePort mode).
 - DIGI PCI pass-through provides the guest Solaris system with direct access to physical serial DIGI devices.
2. **For Charon-SSP 4V(+):**
 - On-board Mode, which extends the original number of two built-in serial ports by a maximum of 14 additional serial ports
3. **For Charon-SSP 4M:**
 - SBus serial card emulation with 8 ports.

7.5.3.13.1 Prerequisites

The following table shows the prerequisites for the TTYX configuration:

| TTYX Option | Host system serial ports | Solaris in virtual SPARC |
|--|--|---|
| TTYX on board mode (SUN-4U and SUN-4V only) | <ul style="list-style-type: none"> • Supported serial ports or • Network ports (sockets) | <ul style="list-style-type: none"> • No additional drivers needed. • First boot after configuration must use the command boot <disk> -r (reconfigure) to create the appropriate special devices for the new serial ports. |
| DIGI AccelePort mode (SUN-4U only) | <ul style="list-style-type: none"> • Supported serial ports or • Network ports (sockets) | <ul style="list-style-type: none"> • Solaris driver required. Charon-SSP emulates the Digi AccelePort 8r 920-PCI board. If you do not have the drivers, please contact Digi support. • Because the emulated board does not support interrupts, the option no interrupt must be set when installing the driver. |
| DIGI board PCI pass-through* (SUN-4U only) | Max. 4 physical serial controllers: <ul style="list-style-type: none"> • Digi AccelePort 920, or • Digi Accele C/X PCI pass-through (PPT) driver is delivered with Charon-SSP. | <ul style="list-style-type: none"> • Solaris driver required. If you do not have the drivers, please contact Digi support. • The option no interrupt must be set when installing the driver. |
| TTYX 4M (SUN-4M only) | <ul style="list-style-type: none"> • Supported serial ports or • Network ports (sockets) | <ul style="list-style-type: none"> • No additional drivers needed (STC driver). • First boot after configuration must use the command boot <disk> -r (reconfigure) to create the appropriate special devices for the new serial ports. |

*Please see [Host System Software Requirements](#) for supported platforms.

When choosing a PCI serial board or an external port server you must consider the requirements of your applications on Solaris. If **modem control** is required, the physical serial ports mapped to the emulated ports must provide this feature. If you have specific requirements with respect to the **electrical interface** provided on a serial port (e.g., RS232 or RS485), you must consider this when selecting the physical serial port board or server.

In addition, or as an alternative, to serial ports added via the TTYX configuration, IP-based port servers are also possible. Their drivers create virtual serial ports on Solaris. This option is independent of what is described in this chapter. It can be useful, for example if Charon-SSP is running in a VM and needs access to a larger number of physical serial ports.

7.5.3.13.2 TTYX On-Board Mode on Charon-SSP/4U/4V

To add, change, or remove additional serial ports in TTYX on-board mode, click on **TTYX** in the left-hand pane of the **Virtual Machine Settings** window. A window like the following screenshot of a Charon-SSP/4U instance opens.

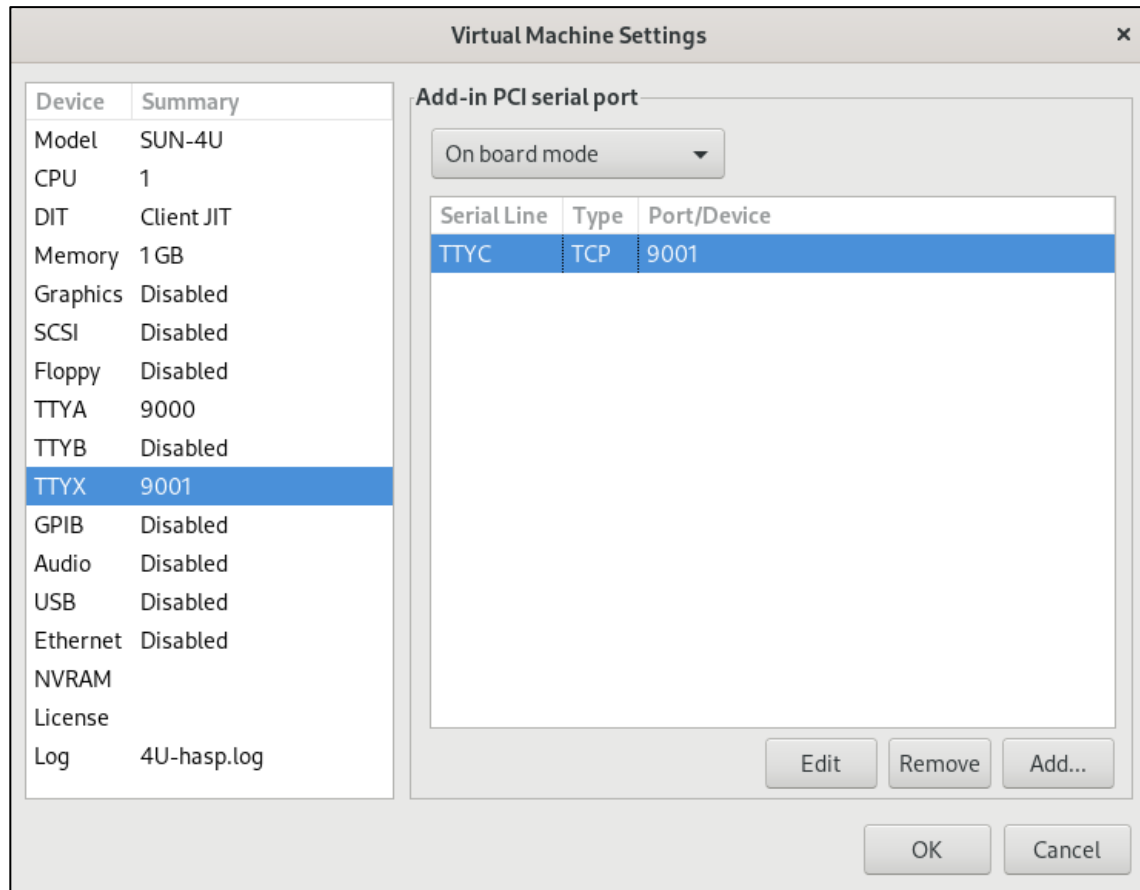


Figure 49: TTYX configuration window (On-board mode)

The example shows one configured serial port (socket) and the options to **Edit**, **Remove**, or **Add**.

The **Edit** and **Remove** buttons become available when selecting an existing port.

The option **On board mode** is selected, i.e., the serial ports are configured in TTYX on-board mode (on a Charon-SSP/4V instance this selection does not exist because on-board mode is the only option).

Additional information:

- If the serial port mode selection can be configured (Charon-SSP/4U), select it as the first step. If you configure serial ports first and then change this option (e.g., from on-board to Digi Board), your configured serial ports will be deleted.
- On Charon-SSP/4V, TTYA and TTYB must be enabled if additional ports are configured in TTYX. Otherwise, the additional devices will not be created in `/dev/term` on Solaris.

7.5.3.13.2.1 Adding Serial Ports in TTYX On-Board Mode

This section describes how to add ports in on-board mode using either physical serial ports or network ports (TCP sockets).

Adding network based serial ports (TCP sockets):

To add serial ports that are TCP sockets (raw mode), click on **Add**. Then

- select the name of the port from the **Serial line** menu
- select the TTY type **TCP**
- enter a unique port number for the socket
- click on **OK**

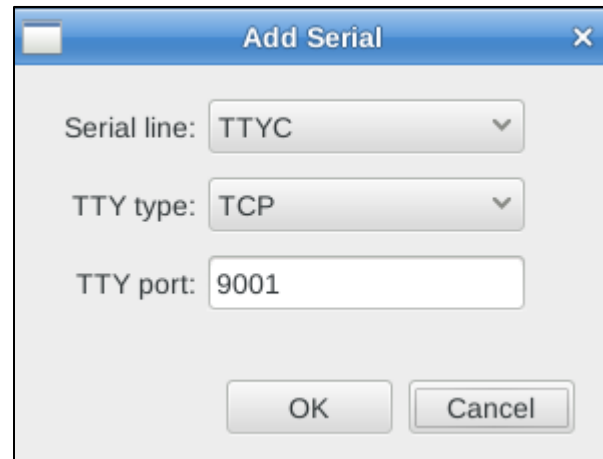


Figure 50: Adding network port (socket) as serial port

The socket number must be unique for each serial line used on the host. Otherwise, a socket IO error will occur and a corresponding message will be written the log file of the virtual system.

Adding physical serial ports:

To add physical serial ports, click on **Add**. Then

- select the name of the port from the **Serial line** menu
- select the TTY type **Physical**
- select the name of the physical serial port by clicking on the **TTY device** browser button (*tty** devices)
- click on **OK**

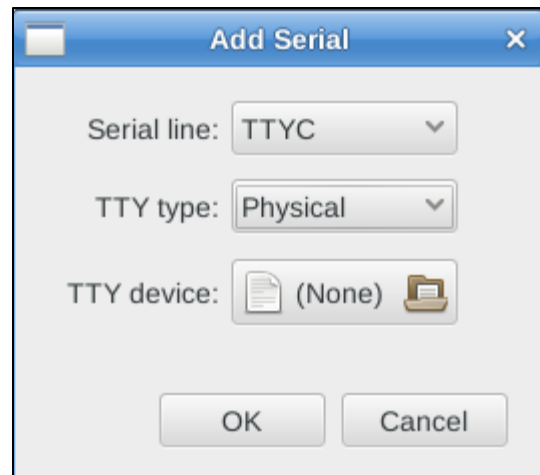


Figure 51: Adding physical serial port

The additional serial ports are named TTYC, TTYD, etc. To activate them in Solaris, reboot the system with the **reconfigure (boot <disk> -r)** option.

7.5.3.13.2 Modifying or Removing TTYX On-Board Mode Ports

Modifying serial ports:

If you want to change the configuration of a port, perform the following steps:

- Open the TTYX configuration window and select the port in question.
- Click on **Edit**. This opens a window very similar to the window for adding a port.
- Make your changes and click on **OK** to save them.

Deleting serial ports:

If you want to delete ports, perform the following steps:

- Open the TTYX configuration window and select the port you want to delete.
- Click on **Remove** to delete this serial port.

After removing ports, you can adjust the Solaris configuration by again by booting with the **boot <disk> -r** command.

7.5.3.13.2.3 Managing TTYX On-Board Mode Ports on Solaris

After booting Solaris with the **reconfigure** option, the new serial ports should be available in Solaris. The easiest way to manage them is the graphical management tool that comes with Solaris.

- In older versions, this is the **admintool**.
- In Solaris 10, this is the Solaris Management Console **smc**.

The following two examples show how the additional ports are displayed in each of the tools. Details about the port configuration can be found in the appropriate Solaris system management guides.

Example of newly added serial ports in the **admintool** of Solaris 2.6:

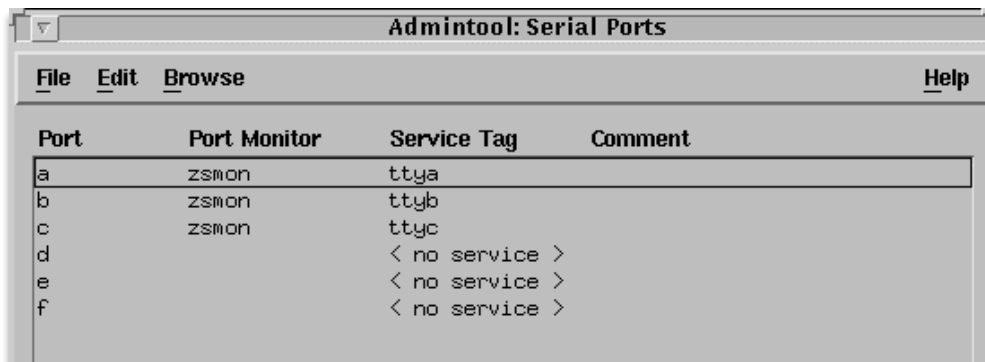


Figure 52: TTYX on-board mode ports in admintool

Example of newly added serial ports in **smc** on Solaris 10:

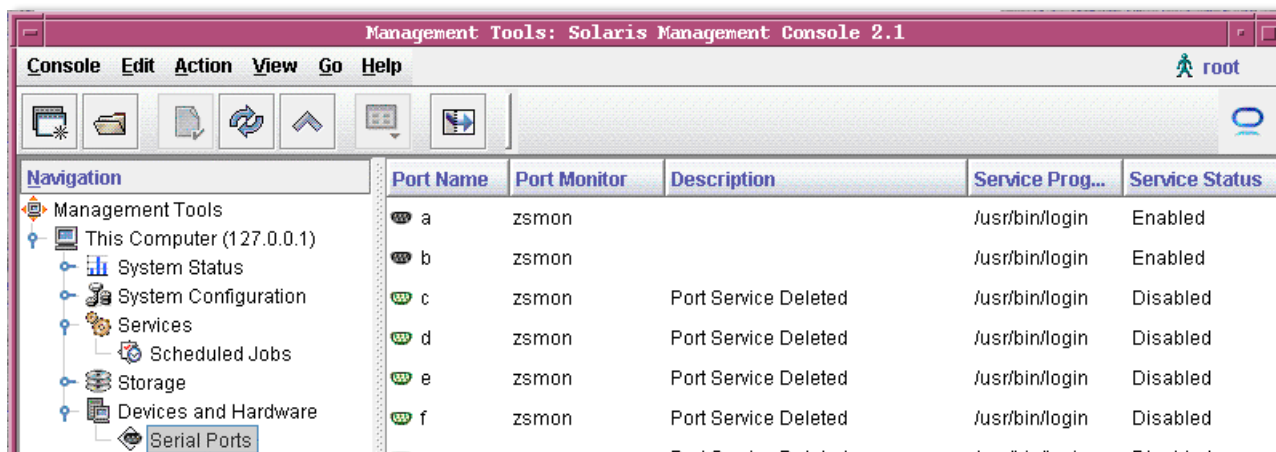


Figure 53: TTYX on-board mode ports in smc

7.5.3.13.3 DIGI AccelePort Mode on Charon-SSP/4U(+)

To add, change, or remove additional serial ports in DIGI board mode, click on TTYX in the **Virtual Machine Settings** window. A window like the following screenshot opens.

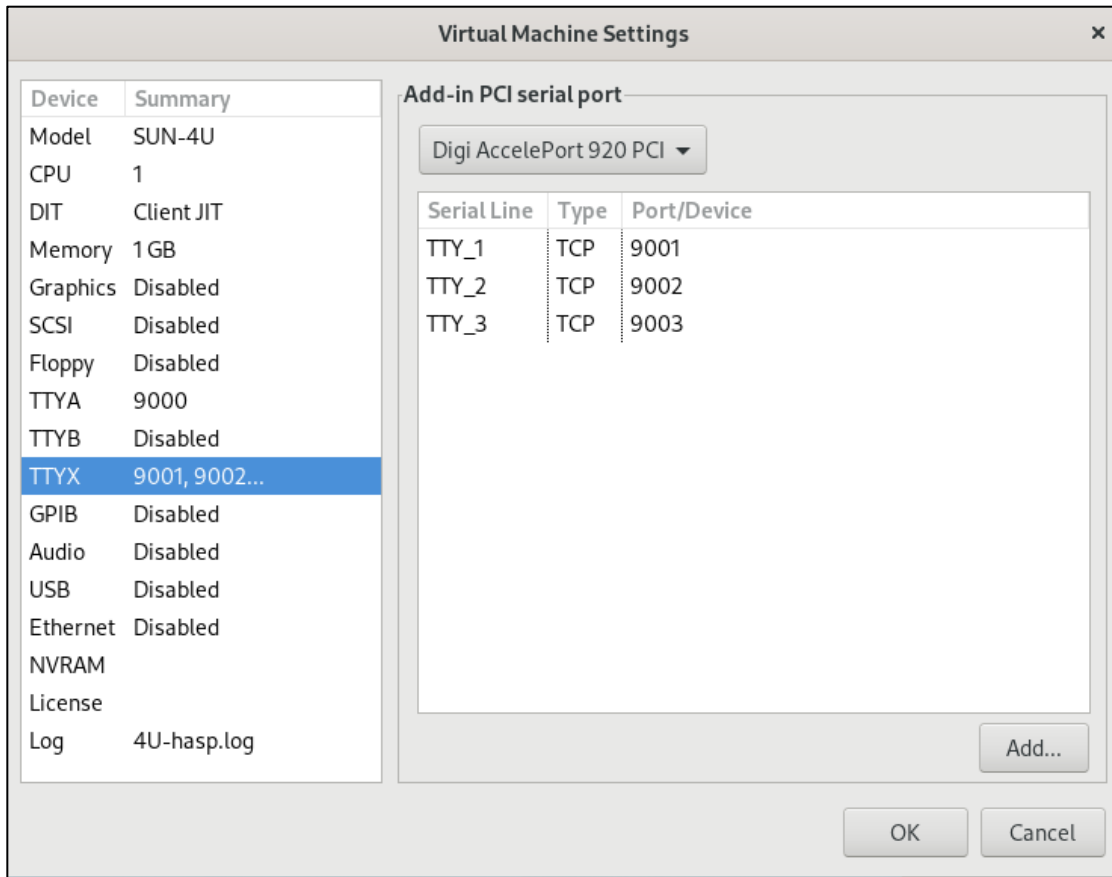


Figure 54: TTYX configuration window (DIGI AccelePort mode)

Note that the option **Digi AccelePort 920 PCI** is selected. The configured serial ports in the example are network ports (TCP sockets).

Select serial port mode as the first step. If you configure serial ports first and then change this option, your configured serial ports will be deleted.

7.5.3.13.3.1 Adding Serial Ports in DIGI AccelePort Mode

Adding serial ports in DIGI AccelePort mode works in the same way as for ports in TTYX on-board mode. Please refer to the section [Adding Serial Ports in TTYX On-Board Mode](#).

Note that the created ports have different names than in on-board mode: They are called TTY_1, TTY_2, etc.

7.5.3.13.3.2 Modifying or Removing Ports in DIGI AccelePort Mode

Modifying and removing ports in DIGI AccelePort mode works in the same way as for ports in on-board mode. Please refer to the section [Modifying or Removing TTYX On-Board Mode Ports](#) for details.

7.5.3.13.3.3 Solaris Driver Installation for DIGI AccelePort Emulation

Charon-SSP/4U emulates a **Digi AccelePort 8r 920-PCI** board.

For Solaris to be able to use these ports, the appropriate driver must be installed. If you do not have the drivers and the associated documentation, please contact Digi support.

The driver packages are data-stream packages. Hence, the installation command is similar to the following example (driver names will vary depending on Solaris version):

Sample installation command for DIGI board driver (Solaris 10)

```
# pkgadd -d 40002543_A.bin
```

The prompts and information texts during the installation in conjunction with the installation guide will lead you through the driver installation.

However, there are **two important items** to note during the installation:

- Select the correct DIGI board PCI adapter (option 0 for the AccelePort Xr 920 PCI adapter emulated by the DIGI AccelePort emulation). Later, you will have to choose the number of ports (8) for it.
- Make sure that interrupts for the board are disabled (answer: **y**) because interrupts are not supported by the emulated device or the PCI pass-through device.

The following image shows these two steps during the installation:

```
Configuring adapter 1.

Adapters supported:

PCI Adapters
  0) AccelePort Xr PCI / Xr 920 PCI
  1) AccelePort Xem PCI
  2) AccelePort C/X PCI
  3) AccelePort EPC/X PCI

What type is adapter 1? 0

In order to reduce response time to small packets (latency), it
may be helpful to enable interrupts on the adapter. However,
doing this will significantly increase driver CPU usage on your
Solaris system. By default, interrupts are disabled.
To enable interrupts, answer no.

Do you want to keep interrupts disabled on the adapter? (y/n)? █
```

Figure 55: DIGI driver installation

Select the AccelePort Xr PCI board (option 0) and leave interrupts disabled on the board (answer: **y**).
Note: if you install the driver for DIGI PCI pass-through devices, the Digi Accele C/X is also supported.

Device special files: the driver installation creates new terminal devices under **/dev/dty**.

7.5.3.13.4 Managing DIGI AccelePort Ports on Solaris

The system does not have to be rebooted after the driver installation. The serial ports are available immediately after a successful installation.

As with the ports in on-board mode, the easiest way to manage the created ports is the graphical management tool that comes with Solaris. In older versions, this is the **admintool**. In Solaris 10, this is the Solaris Management Console **smc**. The following example shows how the ports (a001-8) are displayed in the **smc** tool on Solaris 10. Details about the port configuration can be found in the appropriate Solaris system management guides.

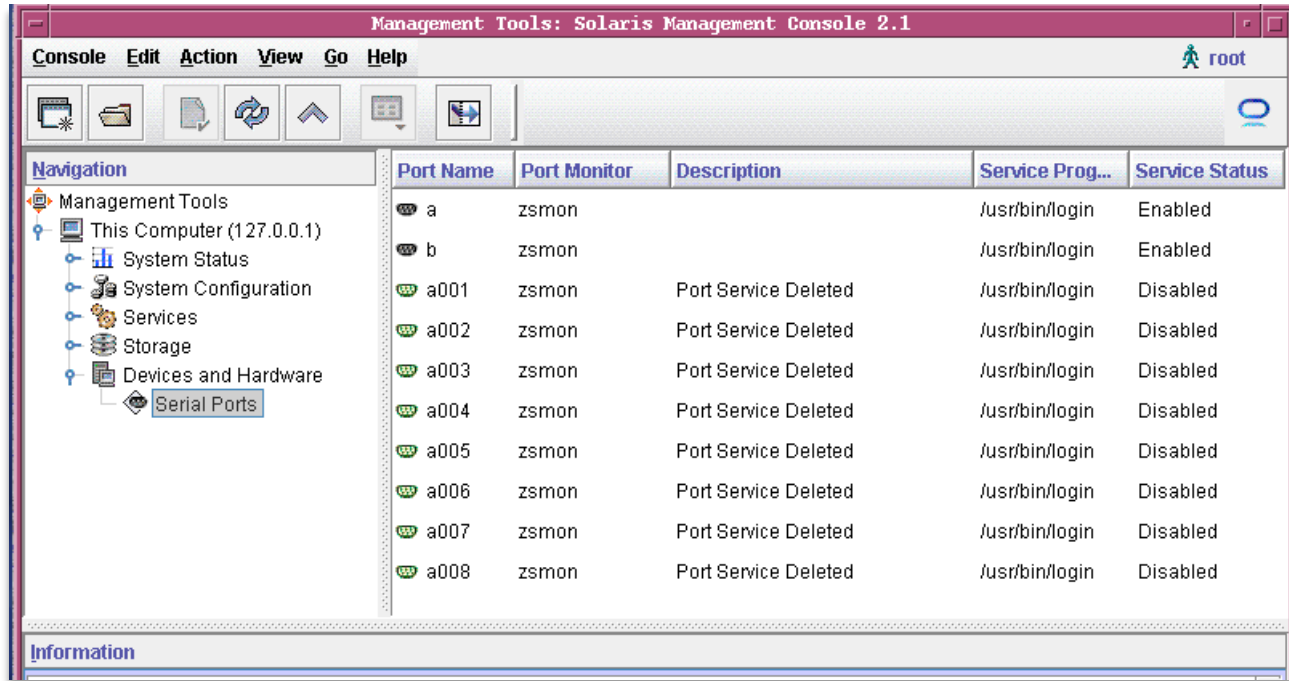


Figure 56: DIGI board ports in smc

7.5.3.13.4 Adding a DIGI PCI Pass-Through Device on Charon-SSP/4U(+)

To add a DIGI PCI pass-through device, select the appropriate option on the configuration screen, then select the device by clicking on the file browser symbol (the devices provided by the Charon-SSP pass-through driver are named **kdigi***):

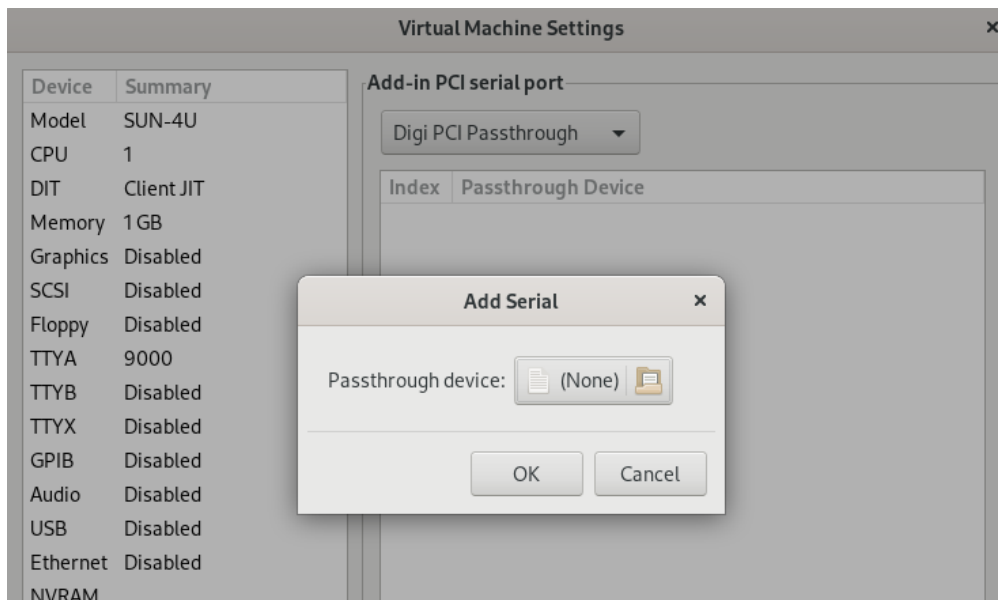


Figure 57: DIGI Board PCI Pass-Through

On Solaris, the driver supporting the device must be installed. See section [DIGI AccelePort Mode](#) for more information. Make sure to select the correct adapter during the driver installation.

If any Linux host operating system driver has been loaded for the PCI board to be used in PPT mode, it must be unloaded (and possibly blacklisted). The device is handled by the Charon-SSP pass-through driver.

The management of the new serial devices in Solaris is the same as described above.

Please note: DIGI PCI pass-through is not available on all platforms and Linux versions. Please refer to [Host System Software Requirements](#). On supported non-Baremetal installations, the kernel module must be loaded manually by the user.

7.5.3.13.5 TTYX Ports on Charon-SSP/4M

Charon-SSP/4M emulates a STC SBus card with eight serial ports and one parallel port. This card is supported by the standard drivers.

7.5.3.13.5.1 Adding Serial Ports in TTYX Mode on SUN-4M

Adding serial ports in TTYX mode on Charon-SSP/4M works in the same way as for ports in TTYX on-board mode on Charon-SSP/4U/4V. Please refer to the section [Adding Serial Ports in TTYX On-Board Mode](#).

Note that the created ports have different names than in TTYX on-board mode: They are called TTY_1, TTY_2, etc.

7.5.3.13.5.2 Modifying or Removing Ports in TTYX Mode on SUN-4M

Modifying and removing ports TTYX mode on Charon-SSP/4M works in the same way as for ports in TTYX on-board mode on Charon-SSP/4U/4V. Please refer to the section [Modifying or Removing TTYX On-Board Mode Ports](#) for details.

7.5.3.13.5.3 Managing SUN-4M TTYX Ports on Solaris

After booting Solaris with the **reconfigure** option (**boot <disk> -r**), the new serial ports should be available in Solaris (`/dev/term/[0-7]`, `/dev/cua/[0-7]`). The easiest way to manage them is the graphical management tool that comes with Solaris.

- In older versions, this is the **admintool**.
- In Solaris 10, this is the Solaris Management Console **smc**.

The following example shows the **admintool** output of a Solaris 2.6 system:

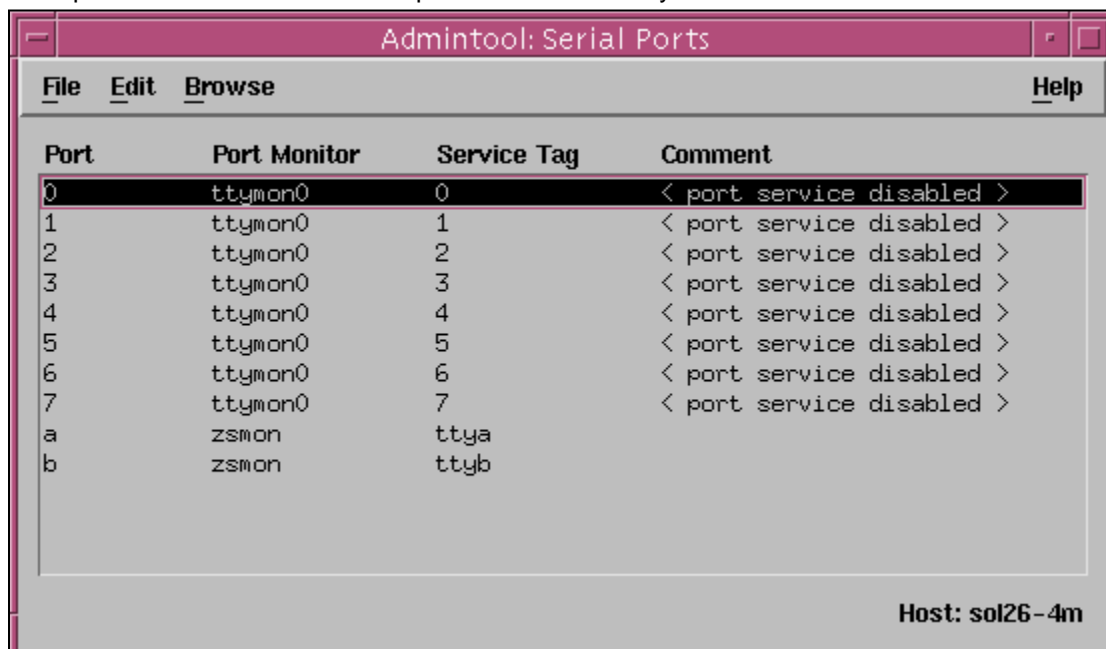


Figure 58: TTYX ports on Solaris 2.6 in a Charon-SSP/4M instance

7.5.3.14 GPIB Configuration on Charon-SSP/4U(+)

Please note: GPIB PCI pass-through is not available on all platforms and Linux versions. Please refer to [Host System Software Requirements](#). On supported non-Baremetal installations, the kernel module must be loaded manually by the user.

Charon-SSP can pass a PCI or PCIe GPIB device to the guest Solaris system as an NI-488.2 GPIB device. To select an existing device, select the GPIB configuration option in the left-hand pane of the Settings window. This will open the configuration window, where you can add a new GPIB device, or edit and remove an existing one. The sample output below shows the add dialog:

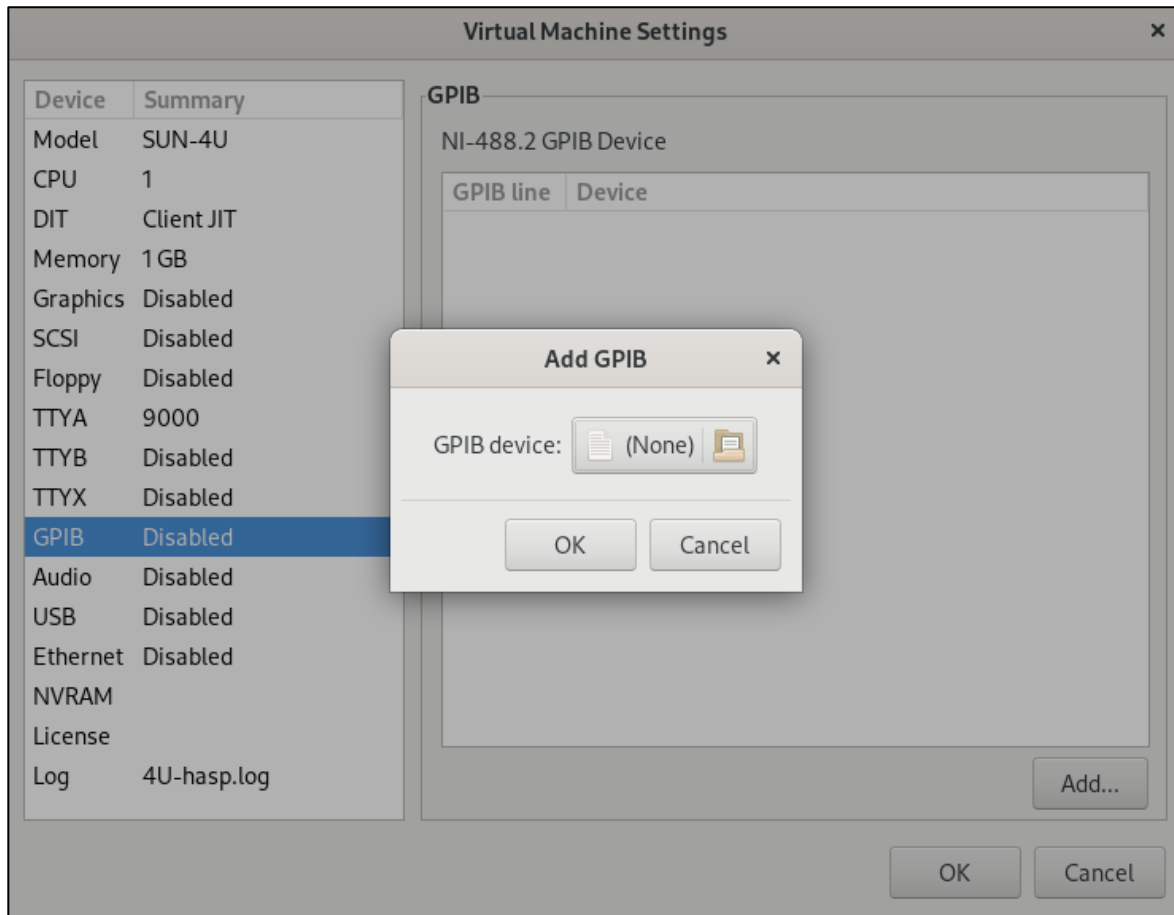


Figure 59: GPIB configuration:

Use the GPIB device file browser button to select the desired device (the devices provided by the Charon-SSP pass-through driver are named *kni**). Then press **OK**.

To edit or remove an existing device, select it. This will activate the **Edit** and **Remove** buttons.

Prerequisites:

- Solaris: use the original device driver for the device.
- Host operating system: unload operating system specific drivers for the device (if present). The device is handled by the Charon-SSP pass-through driver.

7.5.3.15 Parallel Interface Configuration

Please note: this feature is only supported on conventional and Baremetal non-cloud Charon-SSP/4M installations.

Charon-SSP/4M emulates a STC SBus card with eight serial ports and one parallel port. This card is supported by the standard drivers.

To configure the parallel port in the Charon-SSP Manager, click on the option **Parallel** in the left-hand pane of the Settings window. This opens the configuration screen for the parallel port:

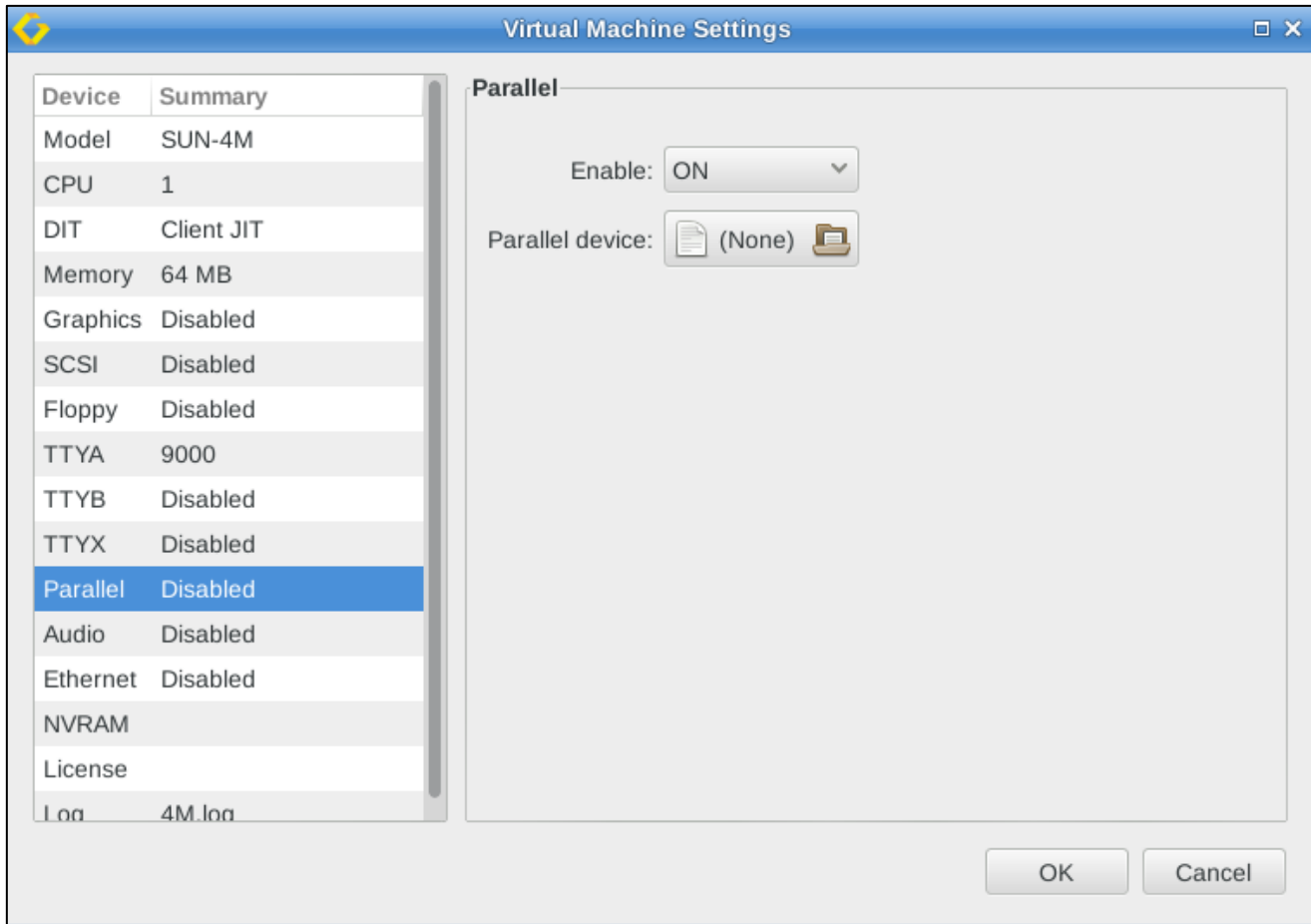


Figure 60: Parallel port configuration on for Sun-4M

After **enabling** the parallel port, you can select a parallel device or file on the host system.

The device name on Solaris is **/dev/bpp0**. To make sure the device is available, boot the guest Solaris system with the reconfigure option (**boot <device> -r**).

7.5.3.16 Audio Configuration

Note:

- The audio configuration is not applicable to Charon-SSP/4V.
- The audio feature is not supported across NAT.
- Currently, only PulseAudio on Linux is a supported audio server.

To enable, disable, or change an audio server for the emulated Solaris system, click on the option **Audio** in the left-hand pane of the Settings window:

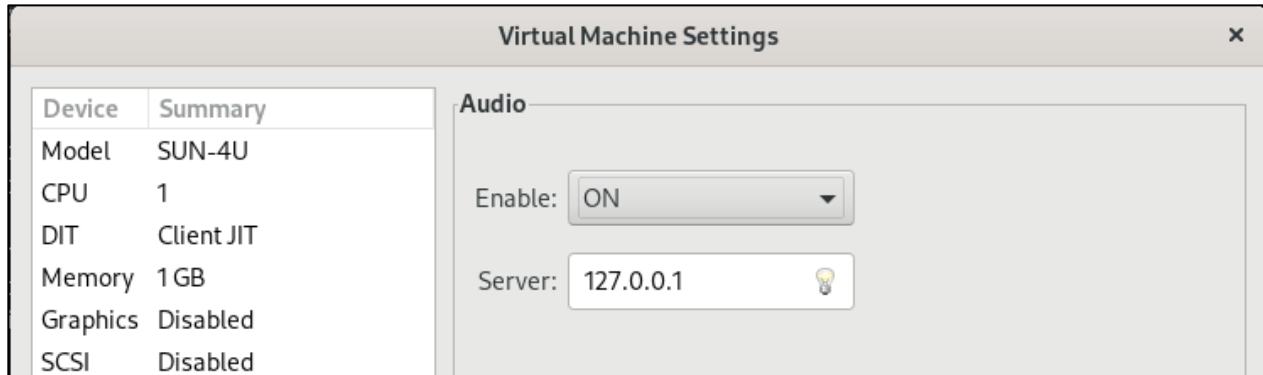


Figure 61: Audio configuration screen

The audio stream is mapped from the emulated DBRle device to a PulseAudio Server accessed remotely on TCP port 4713. After enabling the functionality, you can set the **IP address** of the audio server and click on **OK** to save the settings. The default is the local host system. The address of the audio server must be reachable directly (no NAT). **Currently, only PulseAudio on Linux is a supported audio server.**

The audio emulation emulates a DBRle SBUS adapter and supports the following features:

- CS4215 16-bit multimedia codec for mono and stereo audio playback and recording
- Audio data encoding: uLaw, aLaw, 8/16 bit linear
- Sample rates from 5513Hz to 48000Hz (Voice to DAT quality)
- Speaker volume, recording volume and MIC/speakers muting

In addition to providing an emulated sound card to the Solaris guest operating system, the audio configuration also enables the **Keyboard Buzz** feature, i.e., it allows Solaris applications to create keyboard beeps. If only the Keyboard Buzz feature is used, the Solaris guest system does not require an active sound card.

Prerequisites:**Audio server:**

On the **audio server**, PulseAudio must be enabled for network access. **On Baremetal and Barebone distributions, this is preconfigured.**

On **other Linux** systems, perform the following steps to enable network access to PulseAudio:

| Step | Description | Command |
|------|---|---|
| 1 | Check if PulseAudio is installed on the system. | <pre># rpm -qa grep -i pulseaudio</pre> <p>If the software is not installed, use</p> <pre># yum install pulseaudio pulseaudio-utils</pre> <p>to install it.</p> |
| 2 | Enable network access to the PulseAudio server. | <p>Edit the PulseAudio configuration file:</p> <p><i>If PulseAudio runs under the non-root account of the current desktop user (normal case, recommended):</i></p> <pre># vi /etc/pulse/default.pa</pre> <p><i>If PulseAudio is run as root user (system mode, not recommended, only useful in special cases – e.g., embedded use where no real local users exist):</i></p> <pre># vi /etc/pulse/system.pa</pre> <p>Add the following line if it does not already exist:</p> <pre>load-module module-native-protocol-tcp auth-anonymous=1</pre> <p>Save the file.</p> |
| 3 | Restart the PulseAudio server. | <p>If the default.pa file was modified, the following commands must be run as the non-root user under which PulseAudio was originally started.</p> <p>Stop the PulseAudio server:</p> <pre>\$ pulseaudio -k</pre> <p>Start the PulseAudio server:</p> <pre>\$ pulseaudio --start</pre> <p>If PulseAudio was started in system mode and the system.pa file was modified, the system-wide service must be restarted.</p> <p>If “autospawn = yes” is set in /etc/pulse/client.conf, the process will be restarted automatically after stopping it.</p> |
| 4 | Check if the server is listening to its port. | <pre># netstat -an grep 4713 OR</pre> <pre># netstat -a grep -i pulse</pre> |

Make sure access to the PulseAudio server port is not blocked by a firewall. However, access to the port should only be allowed as required in order to minimize security risks.

Solaris guest system:

On **Solaris**, the audio driver is part of the standard Solaris installation kit. No additional driver should be needed. However, after configuring the audio server (e.g., in Charon-SSP Manager),

- the Charon instance must be restarted, and
- the Solaris guest must be booted with the **reconfigure** option (**boot <device> -r**) to create the **/dev/audio** device.

Charon host system (if different from audio server):

If the pulse-audio server is not the Charon host system itself, the **Charon host system** still requires the **pulseaudio package** and the **alsa-plugins-pulseaudio** package.

Testing the audio functionality:

After configuring the audio function and rebooting the Solaris guest system, use the command-line utilities **audioplay/audiorecord**, the GUI-based Java **media player**, **sdtaudio**, or **audiotool** depending on the Solaris version used. These tools allow you to record and play back audio.

Note: If the connection to the PulseAudio server is interrupted (e.g., configuration changes or network problems), the audio device in the guest stops working. Even if the connection is then restored, the audio device will not start working again until the emulator instance has been restarted.

7.5.3.17 USB Configuration

Please note: this feature is only supported on conventional and Baremetal non-cloud Charon-SSP/4U and Charon-SSP/4V installations.

This feature emulates a USB port for the emulated SPARC system. To enable or disable the USB configuration, select **USB** in the left-hand pane of the settings window. The image below shows a sample:

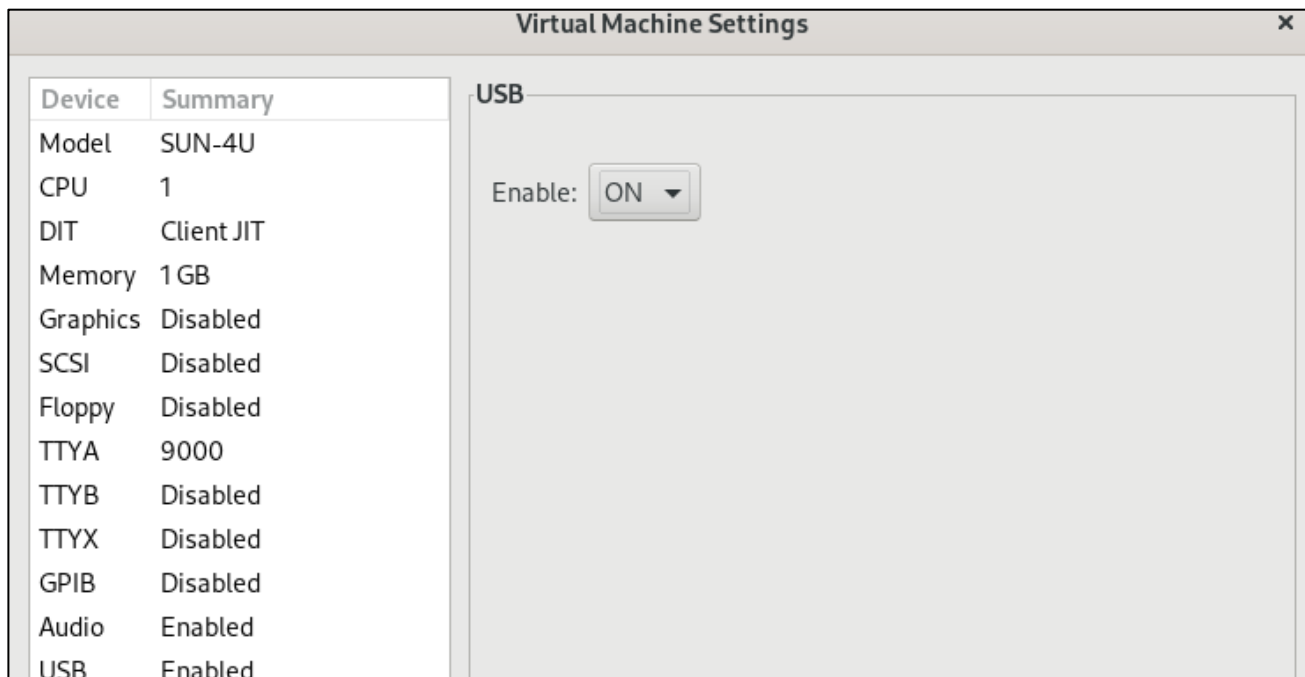


Figure 62: USB configuration

Select **ON** to enable the emulated USB port, select **OFF** to disable it.

Once the USB emulation has been enabled, removable USB storage devices that are attached to the host system can be attached to a running emulated SPARC system. To do this in the Charon Manager, select the running emulated SPARC system, and then **Virtual Machine > Removable Storage**.

Then select the USB device to be attached and the **Attach to VM** option as shown in the sample below:

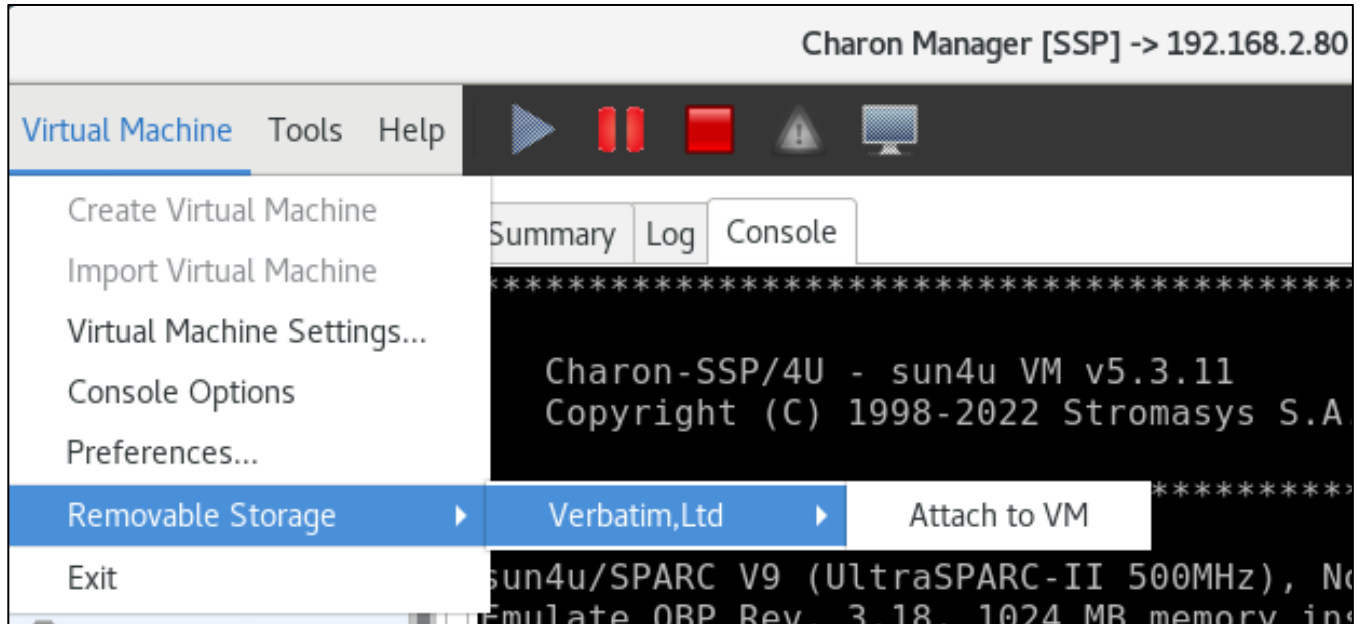


Figure 63: Adding a removable device

Note that the **Removable Storage** option is inactive if there are no running emulated SPARC system with USB devices enabled is selected, or if there are no USB storage devices attached to the host system.

7.5.3.18 Ethernet Configuration

To view or change the virtual machine Ethernet configuration, select **Ethernet** in the left-hand pane of the Settings window as shown in the non-cloud sample below:

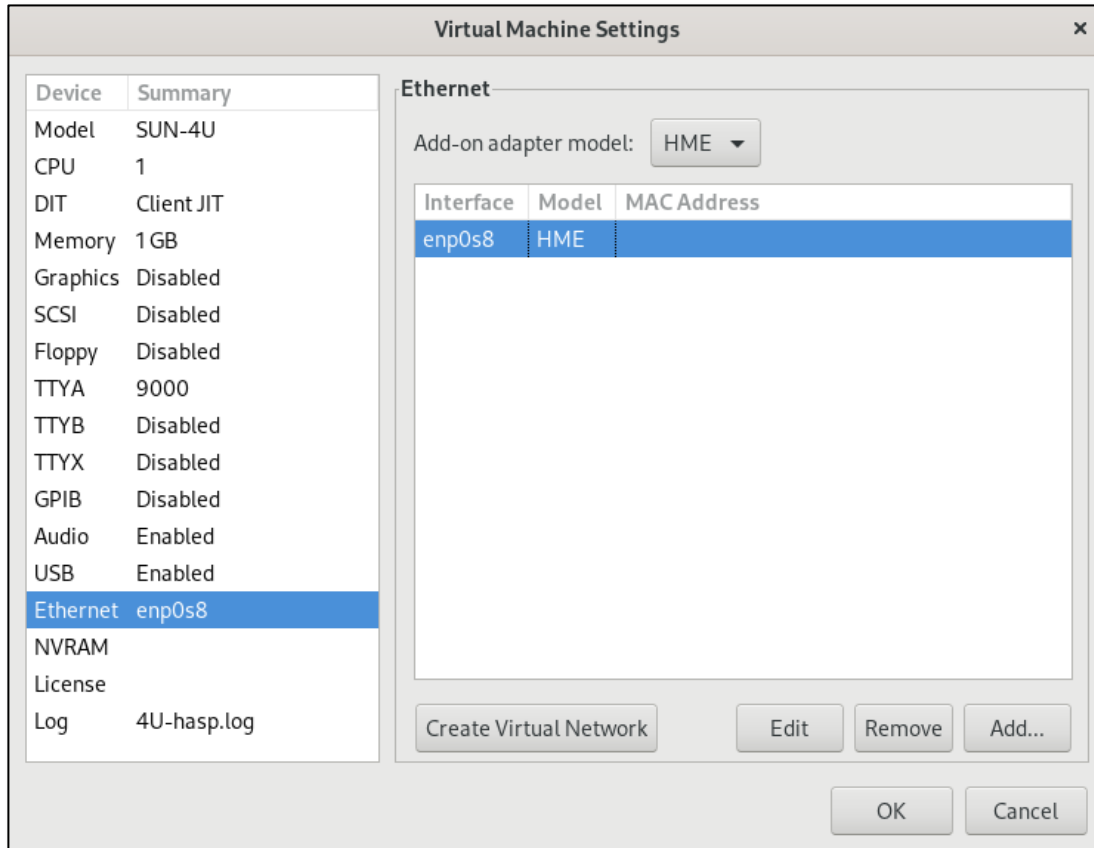


Figure 64: Virtual machine Ethernet configuration

7.5.3.18.1 Supported Adapter Models

The different Charon-SSP variants support different adapter models:

- Charon-SSP/4U: **HME**, **QFE** (4-port Fast Ethernet), and **BGE** (starting with versions 5.2.5 and 5.3.4)
- Charon-SSP/4M: **LE**
- Charon-SSP/4V: **BGE** and **QFE** (4-port Fast Ethernet)

The selection of the adapter model applies to all emulated Ethernet controllers configured for the emulated system. Exception: for Charon-SSP/4U **the first configured Ethernet interface** in the Charon-SSP Manager represents the SPARC on-board device and must be of type HME. It will always show model HME even if another type has been selected.

Prerequisites for QFE controllers on Solaris:

After newly configuring one or more QFE Ethernet ports, boot the guest system with the reconfigure flag (**boot <disk> -r**). To support the QFE controller, Solaris needs the *Sun Quad FastEthernet Adapter Driver* (SUNWqfed). This package is bundled with the Solaris operating environment starting with Solaris 2.6 Hardware: 5/98. For earlier versions of Solaris, the vendor provided a driver CD with the adapter.

After installing the driver, the interfaces should become visible in the **ifconfig** output upon entering the command **ifconfig qfeX plumb**. X denotes the interface number. Use **prtconf** to identify the correct interface numbers. To assign an address to the qfeX interface, create a `/etc/hostname.qfeX` file with the hostname for the interface and add the address for the hostname to `/etc/hosts`.

On Solaris 11 different commands are required to configure the interface:

```
ipadm create-ip netX and ipadm create-addr -T static -a <ip-address>/<netmask> netX/v4
```

QFE configuration notes for Charon-SSP/4U:

To configure a QFE Ethernet ports:

- Select **QFE** in the **Add-on adapter model** drop-down menu.
- Configure the on-board Ethernet device. This will be of type HME even if adapter model QFE is selected.
- Configure the desired number of emulated QFE ports (the number does not have to be a multiple of four).

If the guest system on Charon-SSP/4U does not use the HME controller, you can create a virtual network without an external interface and assign one of the bridge interfaces to the controller as a dummy interface. Alternatively, you could assign the localhost interface (lo) to the unused emulated Ethernet device

7.5.3.18.2 Charon-SSP Ethernet Configuration Screen Functions

The Ethernet configuration screen offers several functions:

- To **create** a virtual network, **click** the **Create Virtual Network** button. For further details on creating, changing, and removing a virtual network, see the section [Host System Network Configuration](#).
If the Charon-SSP host system is installed in a VMware environment, it is recommended to use vNICs to provide Ethernet interfaces to guest systems. Connecting a virtual bridge with a vSwitch is not recommended as this can lead to serious network problems in the LAN unless the configuration is very well understood. Similar considerations apply to other hypervisor environments.
- To **modify** an existing virtual Ethernet adapter, **select** it from the list of configured devices and **click** on **Edit**.
- To **remove** an existing virtual Ethernet adapter, **select** the adapter from the list of configured devices and **click** the **Remove** button.
- To **add** a new virtual Ethernet adapter, **click** the **Add** button. Please note that on version 8.x of the supported Linux distributions only interfaces that are under NetworkManager control will be offered.

After selecting to **add** or to **edit** an adapter, a window like the one below will open:

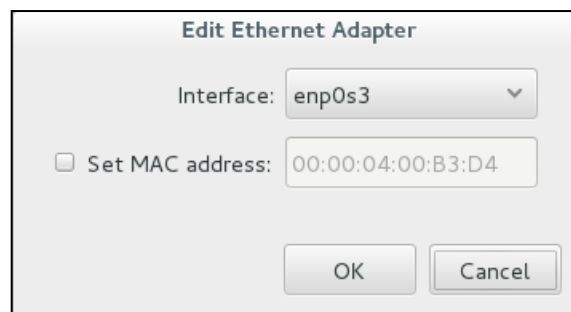


Figure 65: Edit Ethernet adapter

The following table lists and describes the fields in the **Add/Edit Ethernet Adapter** configuration window:

| Field | Description |
|-----------------|---|
| Interface | <p>Select the host attached Ethernet device to be connected to the virtual device. This field is a drop-down list of all the network adapters available on the host system. Important points:</p> <ul style="list-style-type: none"> • The interface must allow promiscuous mode unless the configuration described below (under Special configuration considerations) is used. • It is permitted to assign the localhost interface (lo) to an emulated device (if only a dummy device is required in the guest). • It is also permitted to add the same physical device to multiple emulated Ethernet devices of the same instance. However, this is not recommended for performance reasons. • Sharing a NIC between emulator and host (not recommended for performance reasons) is possible but requires promiscuous mode and the MAC addresses of host and emulated system to be different (normally, this is automatically taken care of by Charon-SSP toggling the locally-administered bit of the MAC address for the interface assigned to the guest system). It also requires mutual host routes via a default gateway for the Charon host and guest system if they should be able to communicate with one another on a shared NIC. Not an option in cloud environments. • VMware and the Solaris MAC address: VMware has several parameters to protect the environment from forged MAC addresses (e.g., the forged transmits and the address change parameters). If a MAC address is to be used for a Charon instance that is different from the host NIC MAC address, these parameters must allow such a configuration. • Assigning the same physical interface to more than one Charon-SSP instance is possible but not supported for production operation. It requires promiscuous mode and manual setting of unique MAC addresses for the Charon instances. I/O performance will be significantly degraded. For Stomasys testing purposes only. <p>Special environment-specific configuration considerations:</p> <ul style="list-style-type: none"> • On VMware ESXi and other hypervisors promiscuous mode is disabled by default on virtual adapters. The best solution when running the Charon-SSP host in a VM is to dedicate a vNIC to the emulator and to set the MAC address of the emulated adapter to the same address as the MAC address of the ESXi vNIC. See Set MAC Address parameter below. If a dedicated vNIC is not possible, the interface must support promiscuous mode and the Hypervisor must allow multiple MAC addresses for this interface. • In Cloud environments, promiscuous interface mode is generally not an option. In such environments, if the guest system needs a dedicated NIC, the emulator and guest configuration must use the MAC and IP address values assigned to the interface by the cloud provider. The MAC address of the cloud instance NIC is assigned to the emulated NIC and the IP address of the cloud instance NIC is configured not on the Charon host, but on the guest. Please refer to the cloud-specific Getting Started Guides and to the dedicate NIC example in the appendix for more information. |
| Set MAC Address | <p>To force the MAC address of the virtual Ethernet device to a specific value, select the checkbox and enter the address in groups of two-character hexadecimal digits, separated by a colon, e.g., 08:00:2b:aa:bb:cc.</p> <div style="border: 1px solid black; background-color: #ffffcc; padding: 5px;"> <p>This option can be useful in cases where licensing is tied to a network adapter MAC address. It can also be used to avoid having to set a VMware virtual network adapter to promiscuous mode, or in cloud environments where promiscuous mode is typically not possible. If this configuration is used, the emulator needs a dedicated NIC on the host system.</p> </div> |

7.5.3.19 NVRAM Configuration

To view or change the NVRAM configuration of the emulated system, select **NVRAM** in the left-hand pane of the **Virtual Machine Settings** Window:

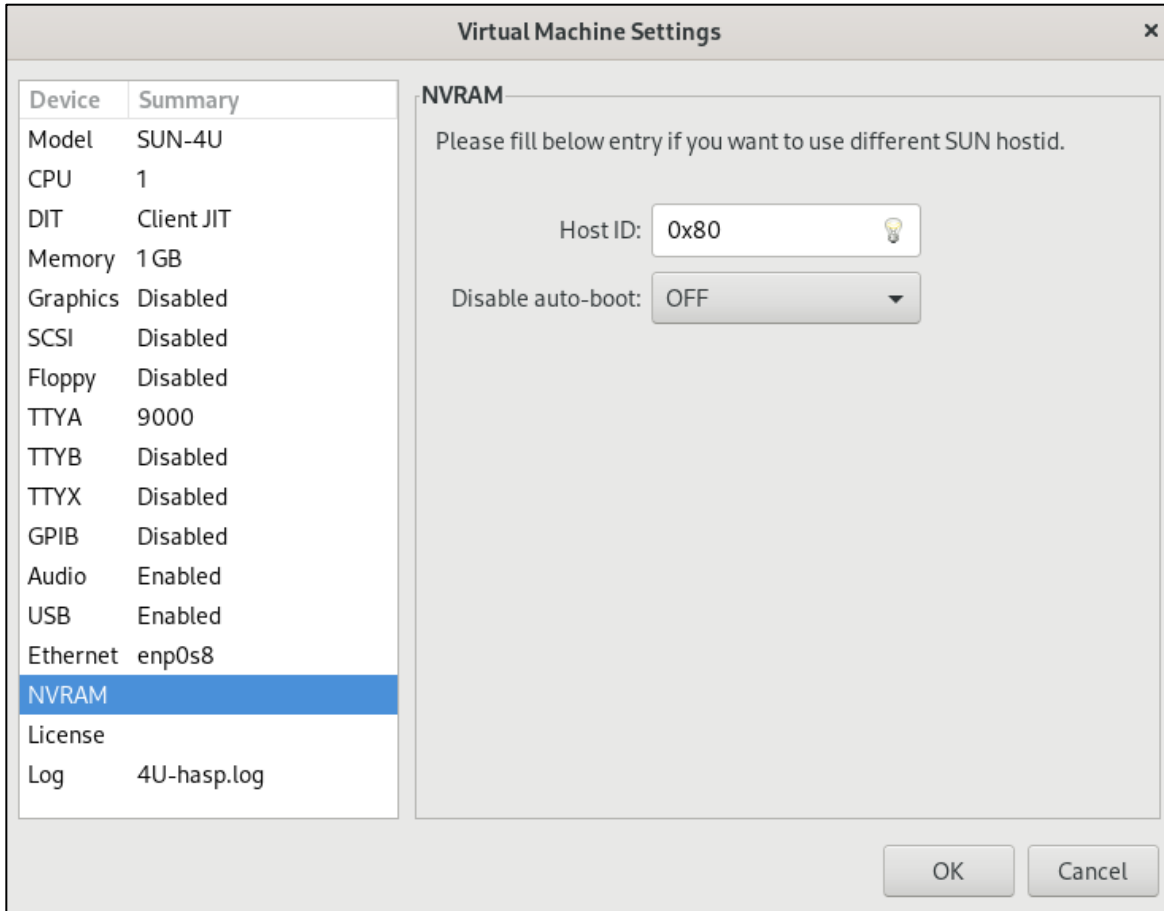


Figure 66: NVRAM configuration

On this screen, two NVRAM parameters can be configured:

| Field | Description |
|--------------------------|---|
| Host ID | This option can be useful in cases where licensing is tied to the host ID of the physical system. |
| Disable auto-boot | Default: OFF. The automatic boot of the emulated system can be disabled. |

7.5.3.20 License Settings

Please note: this section is **not applicable** to Charon-SSP AL images with automatic licensing or AutoVE licensing (the public license server does not require configuration; the configuration for the AutoVE server must be added to the cloud instance before first instance launch). Please refer to the VE license server guide (see [Charon licensing documentation](#)).

The license configuration window enables specific, per-VM license configuration options for Sentinel/Gemalto and VE licenses. Open it by clicking on **License** in the left-hand pane of the window. Example:

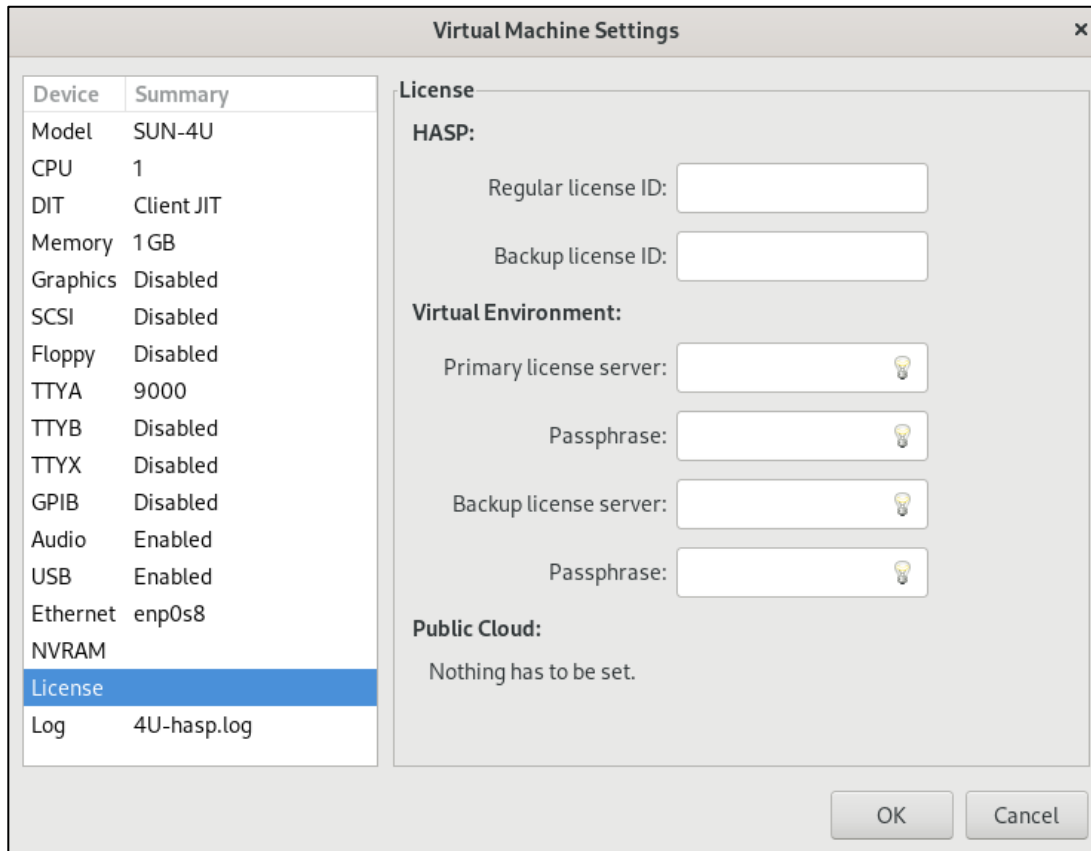


Figure 67: License configuration for HASP and VE licenses

The following table describes the configurable parameters.

Configurable parameters:

| Parameter | Description |
|------------------------------------|--|
| HASP section | <p>This setting helps to manage a situation where multiple Sentinel HASP license keys are available on the system. Using the license settings, you can define</p> <ul style="list-style-type: none"> • a primary (Regular License ID) and • a backup (Backup License ID) license key. <p>This is useful if there is a production license and a backup license limited to a certain number of hours runtime. In this case, the parameters in this section avoid the unintentional depletion of the backup-key hours. The parameters can also be used to define the correct key should Charon-SSP identify the key for a different product as the default key by mistake.</p> <p>Identifying the HASP license ID:</p> <ul style="list-style-type: none"> • You can use the following command to identify the license ID of all installed Sentinel/Gemalto licenses (run from a local terminal): <pre>\$ /opt/charon-agent/ssp-agent/utils/license/hasp_srm_view -all</pre> • Or you can use the Charon-SSP Manager HASP Management tools (HASP Viewer) to identify the license ID. <p>Refer to the license management section for more detail.</p> |
| Virtual Environment section | <p>If using a VE license server, the following configuration options are available:</p> <ul style="list-style-type: none"> • Primary license server: format <code><ip-address>[:<port>]</code> where <ul style="list-style-type: none"> ○ <i>ip-address</i> is the name or address of the primary license server, ○ <i>port</i> is the TCP port on which the license is served (if not specified, the default port 8083 will be used). <p>The primary license server is a mandatory parameter for a VE scenario.</p> • Backup license server: optional; the format is the same as the primary license server definition. You can define a backup server that can be used in case the primary server becomes unavailable. The license on the backup server is typically limited to a certain number of hours runtime. • Passphrase: passphrase(s) to authenticate the client to the primary and/or backup server. The passphrase is defined when the license is created by Stromasys. Please let Stromasys know if you require a license with or without a passphrase (can be selected per product section). The use of a passphrase requires emulator versions 4.3.x and higher, and license server versions 1.1.x and higher. If the license contains more than one product section, there will be a passphrase per product section. Use the passphrase defined for the product section that the emulator will use. To identify the passphrase, use the command /opt/license-server/license_viewer on the primary and backup license servers. <p>Please note:</p> <ul style="list-style-type: none"> • To access the license, the client must have access to the port on the license server used to serve the license (default: TCP 8083). • To define a remote Sentinel HASP license server, use Tools > HASP Tools > HASP Manager. |
| Public Cloud section | <p>No user accessible configuration. This is applicable to cloud-specific Charon-SSP AL images that use a public, Stromasys-operated license server. The licensing system of such instances is built in and works without configuration.</p> |

7.5.3.21 Log Configuration

To view or change the virtual machine logging configuration, select **Log** in the left-hand pane of the Settings window.

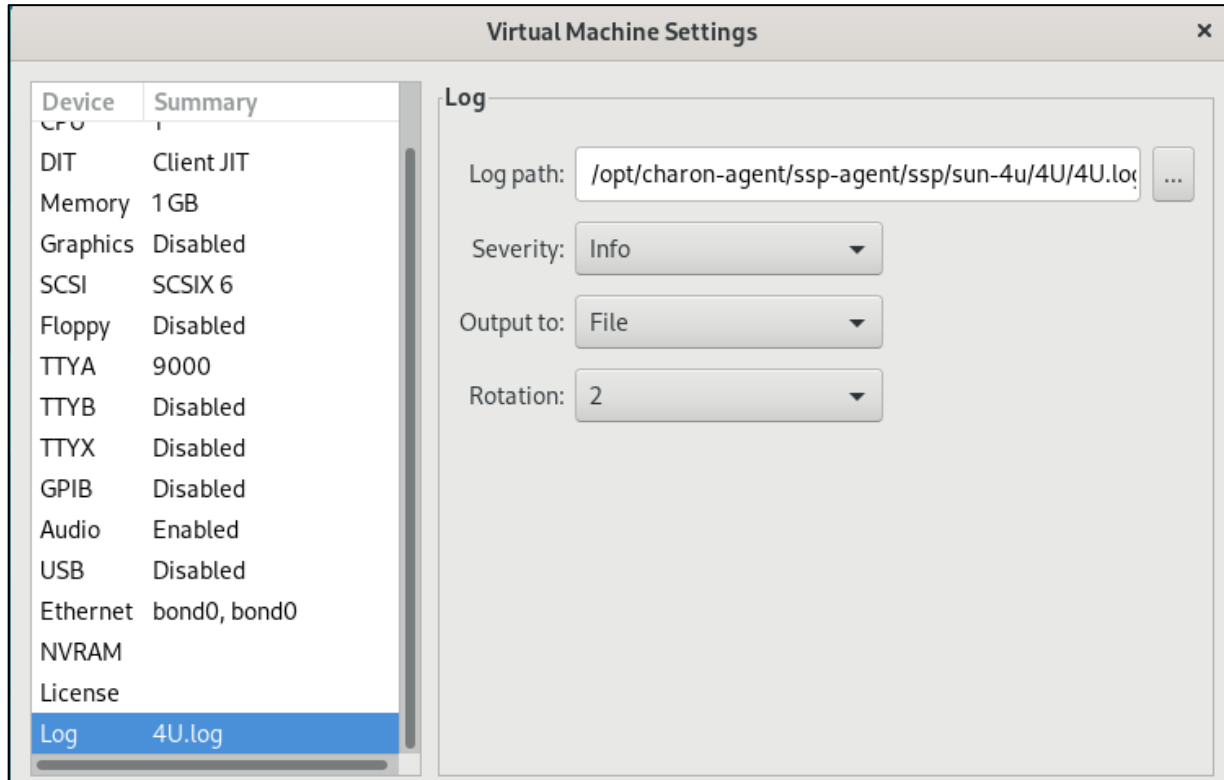


Figure 68: Log configuration window

The following table lists each of the **fields in the log configuration window** and describes their operation.

| Field | Description | |
|------------------|--|---|
| Log path | Specify the path and name of the log file. Configuration option not available for Charon-SSP AL images. | |
| Severity | Set the minimum level of messages that should be reported. Legal values are debug , info , warning , error and fatal . The default is info . | |
| Output to | Indicate the location to which virtual machine logging information should be written. The default is file. | |
| | Option | Description |
| | file | Write virtual machine logging information only to the file configured in Log path . |
| | console | Write virtual machine logging information only to the virtual machine console. |
| | all | Write virtual machine logging information to both the file configured in Log path and the virtual machine console. |
| Rotation | Select the number of old versions of the log files to be saved. The Charon-SSP log files are rotated when the virtual machine starts and, during operation, based on the number of lines written to the log. Once the number of log lines reaches 800.000, the log is rotated. | |

7.5.3.21.1 Viewing the Charon-SSP Log Files

Currently, Charon-SSP writes three types of instance specific log files:

- **Emulator log** (*<vm-name>.log*): it documents the operation and potential problems of the Charon-SSP instance in question. For example, if no valid license is available, this is logged here.
- **Console log** (*<vm-name>_<serial-line>.log*): if configured, Charon-SSP keeps a console log for TTYA and TTYB.
- **Crash log** (*<vm-name>.crash.log*): should the Charon-SSP instance terminate unexpectedly, trace-backs and similar information are found in this log file. The contents help Stromasys engineering to identify and repair the problem. In addition, this file contains the standard error output of the Charon-SSP emulator (for example, missing shared libraries). This can be a helpful troubleshooting tool in case of problems. This file is only available if the emulator is started via the Charon Manager. If you start the emulator manually from the command-line, review/capture the command-line output in case of problems.

The log files can be viewed using **Log** tab of the Charon-SSP Manager:

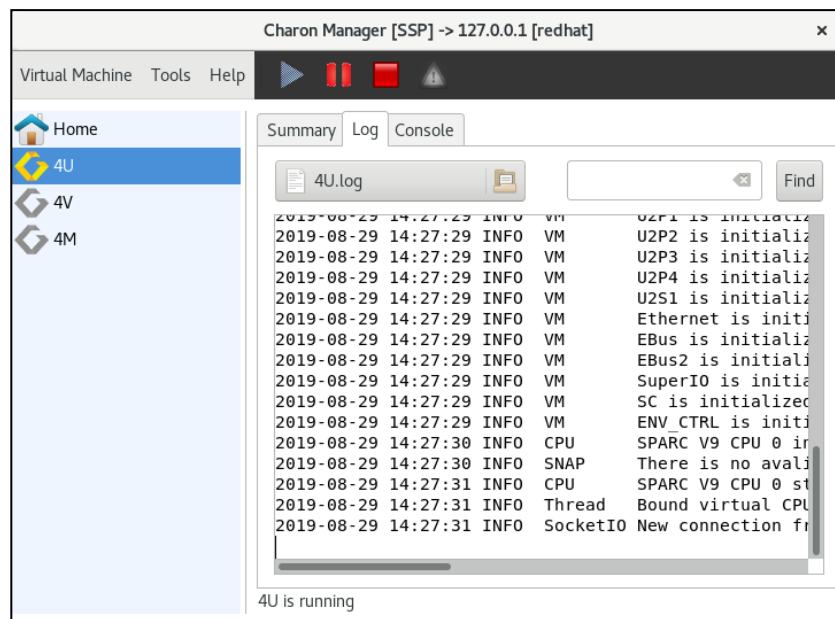


Figure 69: Log tab of the Charon-SSP Manager

To select a log file, click on the file-browser button. This shows all available logs in the default (or configured log path) and lets you select a file. You can also select a different file path to display log files in other locations.

Entering text into the search field and pressing find will filter the log contents according to the search string.

7.5.4 Starting, Stopping, and Suspending the Emulated System

7.5.4.1 Starting the Emulated System

Once the emulated SPARC system has been configured, you can start the emulated system.

An emulated system can be started

- interactively via the Charon Manager, or
- as a service during host system startup.

Please note: the emulator will not run (or not run properly) if Linux CPU accounting is enabled. In such cases, you will see an error like “ERROR Thread Unable to create I/O thread” in the emulator log. Should you encounter this problem, refer to [this troubleshooting article](#) for more information.

7.5.4.1.1 Interactive Start

An emulated system can be started interactively from the Charon Manager. There are **three different options** inside the Charon Manager to start an emulated system:

1. Click on the **blue triangle** at the top of the Charon Manager window, or
2. right-click on the virtual machine and select **Run Virtual Machine** from the context menu, or
3. select the virtual machine. Then select the Summary tab and click on the **Run Virtual Machine** button at the bottom of the summary page.

The image below shows the three options:

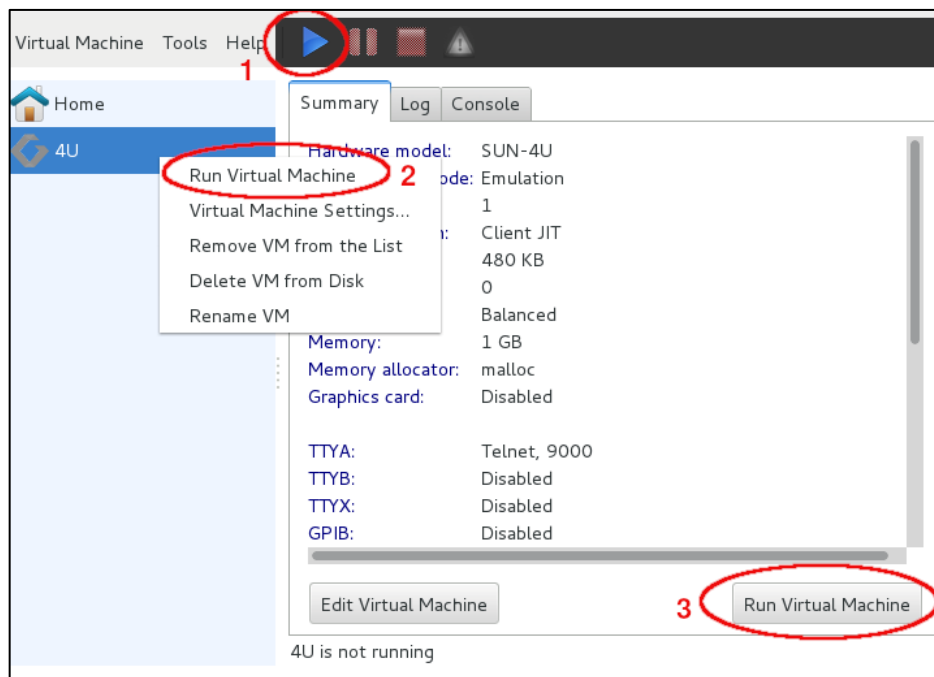


Figure 70: Starting the emulator from Charon Manager

After the system has been started, the built-in console and the emulator log are displayed in Charon Manager.

The image below shows the console prompt of an emulated SPARC system:

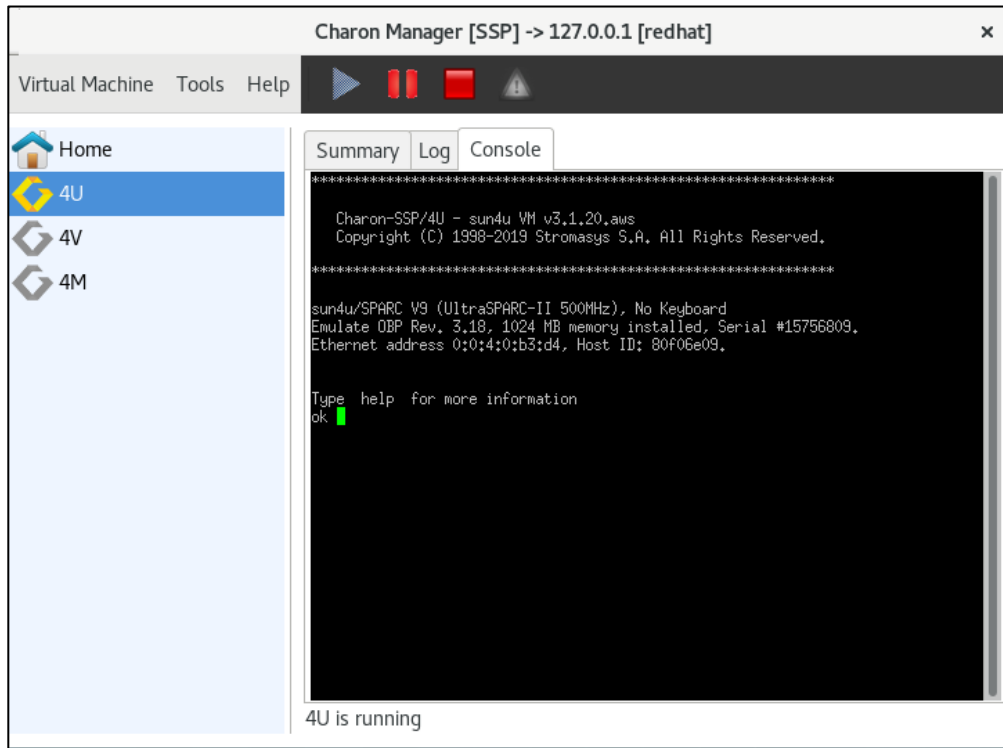


Figure 71: Charon Manager built-in console

The **Log** tab allows the user to view the different log files produced by the emulator. The example below shows a view of the emulator log file:

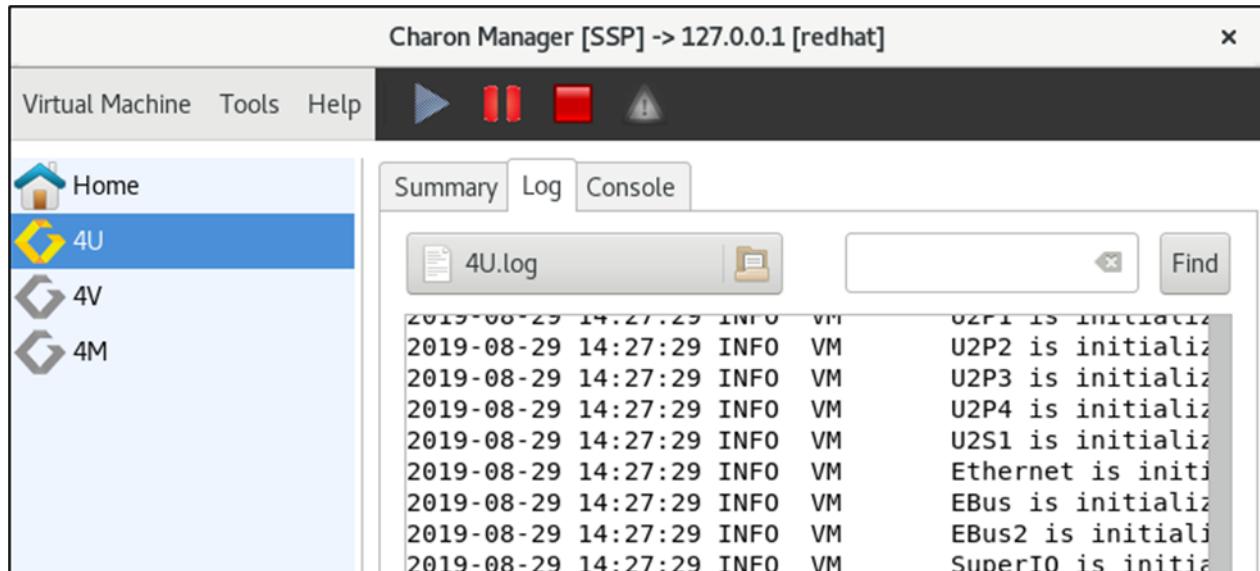


Figure 72: Charon Manager log display

7.5.4.1.2 Start with Host System Startup

The model configuration screen in Charon Manager allows you to automatically start the emulated system (optionally with a delay) when the host system starts, as shown below.

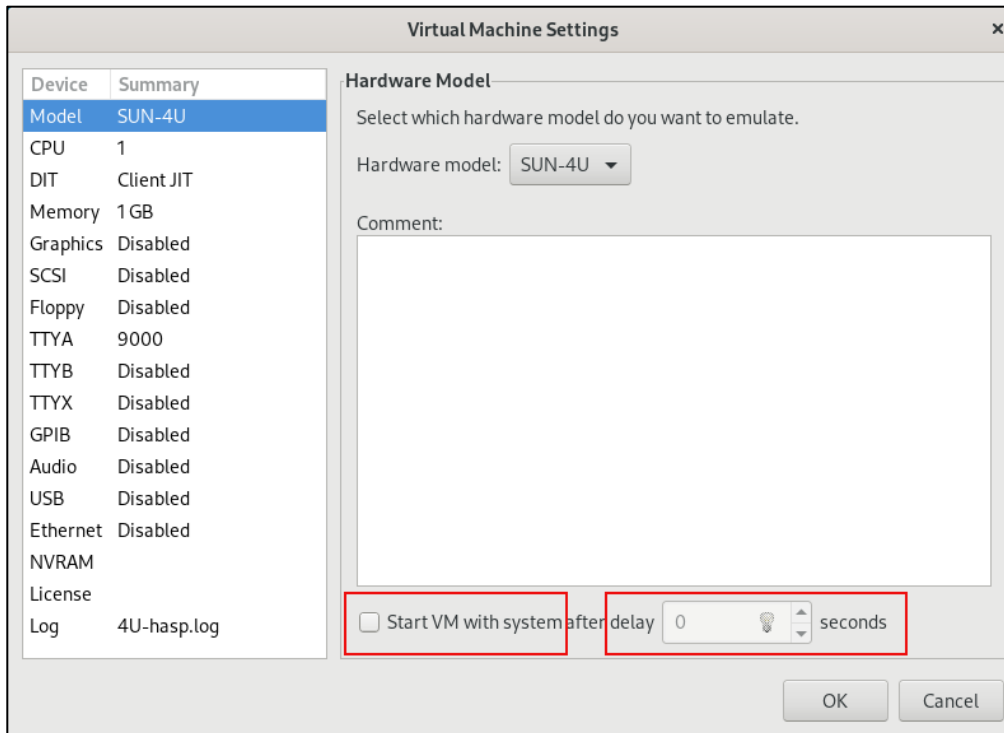


Figure 73: Emulator start with host system

Important: the guest operating system must be shut down cleanly or be suspended before stopping the emulator when the host system is shut down. Failing to do so may cause corruption of the guest operating system.

7.5.4.2 Stopping the Emulated System

After shutting down the guest operating system cleanly, you can stop the emulator in Charon Manager:

- Select the emulated system you want to stop.
- Click on the **red square** at the top of the window.

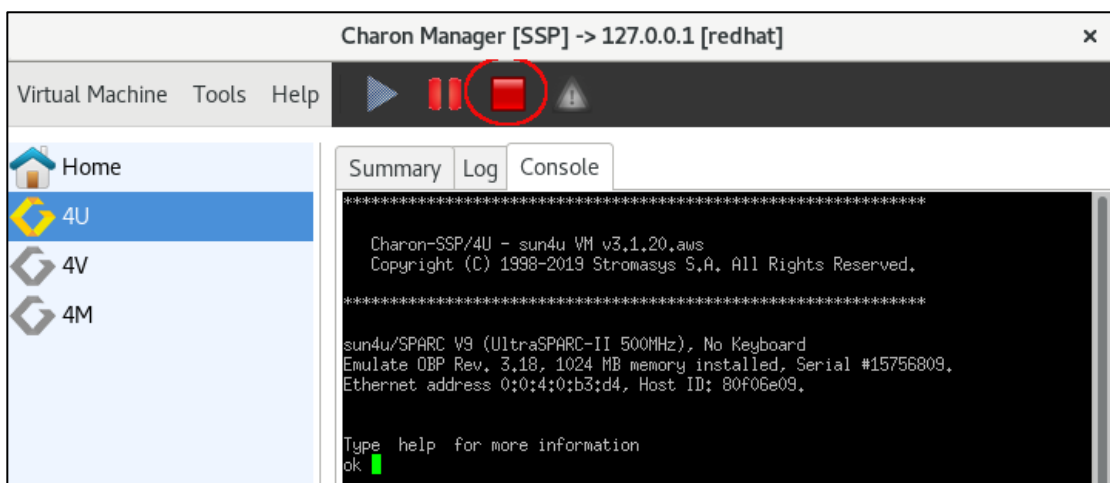


Figure 74: Stopping the emulator

Alternatively, you can also stop the emulator by entering the command `poweroff` at the console prompt.

7.5.4.3 Suspending the Emulated System

The emulated system can be suspended. This means that the memory content of the system is saved.

Please note: the files created during the suspend operation and used for the resume operation are not suitable for creating a backup copy of the emulated SPARC system.

To suspend the system, use the **pause symbol** at the top of the Charon Manager window as shown in the image below:

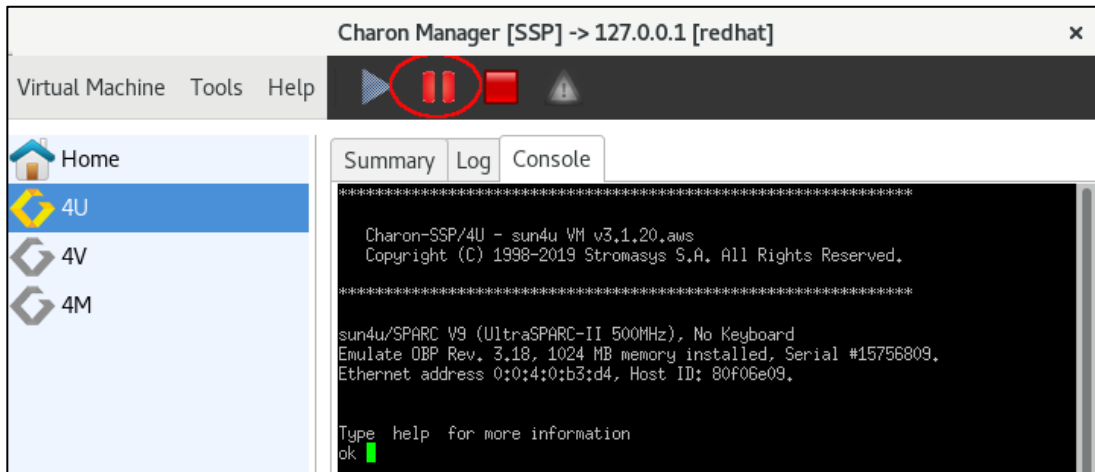


Figure 75: Suspending the emulated system

At the next start, the emulated system will resume with the status it had when it was suspended.

The created files are saved in the `/charon/storage/ssp-snapshot` (Baremetal and cloud-specific marketplace images) and `/home/ssp-snapshot` (conventional product) directories by default. The directory can be changed (**Virtual Machine > Preferences**) when using a product other than a cloud-specific Charon-SSP AL image. The files are deleted automatically after the first regular shutdown of the guest system and emulator after resuming the suspended system.

You can suspend a running Charon emulator instance from the command-line using the command

```
# kill -SIGTSTP <ssp-pid>
```

where `ssp-pid` is the PID of the Charon emulator process. If you do not use the Charon Manager to start the emulator process, note that you must specify the location for storing the snapshot files on the command-line when starting the emulator.

7.5.5 Virtual Machine Context Menu

Each configured virtual machine in the Charon-SSP Manager has a context menu that is opened by **clicking** on the virtual machine with the **right mouse button**.

This context menu has the following options:

- Run Virtual Machine
- Virtual Machine Settings
- Remove VM from the List (non-AL only)
- Delete VM from Disk
- Rename VM
- Backup VM (Charon-SSP AL images only)

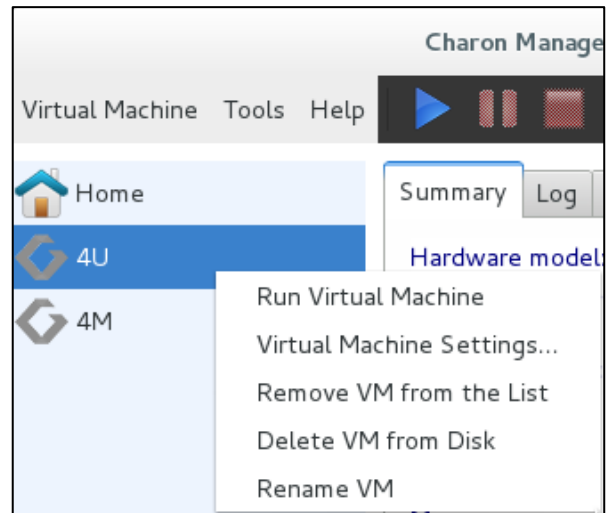


Figure 76: Virtual machine context menu (non-AL)

The different options are described in the following sections.

Right-clicking into the virtual machine list pane when no virtual machine is selected opens an additional small context menu with options to create or import a virtual machine.

7.5.5.1 Run the Virtual Machine

The option **Run the Virtual Machine** starts the virtual machine. The Charon-SSP icon next to the machine name changes from gray to multi-color to indicate a running instance. After starting the virtual machine, all options in the context menu apart from **Virtual Machine Settings** (and, in the case of a Baremetal system, **Backup VM**) are inactive until the virtual machine is stopped again.

The **Run the Virtual Machine** option is equivalent to

- clicking on **Run Virtual Machine** at the right-hand bottom of the virtual machine summary page, or
- clicking on the blue triangle at the top of the Charon-SSP Manager window.

This option is inactive while the virtual machine is running.

7.5.5.2 Virtual Machine Settings

The option **Virtual Machine Settings** leads to the configuration options that are described in section [Configuring a Virtual Machine](#).

7.5.5.3 Remove Machine from the List (non-AL only)

The option **Remove Machine from the List** removes the virtual machine from the machine list in the Charon-SSP Manager. The associated configuration file and virtual storage container files are not deleted. If needed, the machine can be re-imported into the Charon-SSP Manager using the **Import** option under the **Virtual Machine** menu item of the Charon-SSP Manager.

This option is inactive while the virtual machine is running.

7.5.5.4 Delete VM from Disk

The complete removal of a virtual machine must be performed in several steps:

1. Shut down the guest operating system and stop the virtual machine if it is running. The menu option to delete a virtual machine is inactive while the virtual machine is running.
2. **Right-click** on the name of the virtual machine in the left-hand pane of Charon-SSP Manager.
3. The context menu opens. Select **Delete VM from Disk**. You will be prompted to confirm your choice.
4. Any configurations and log files related to the system are removed and no longer exist. Associated virtual storage container files are not deleted.

Important: In older versions, Charon-SSP Manager may not ask you to confirm this action and the configuration and log files **are immediately deleted**.

7.5.5.5 Rename VM

The option **Rename VM** allows you to rename your virtual machine. When you click on the option, you will be prompted for the new VM name. Enter the new name and confirm your input by clicking on **OK**.

The virtual machine appears in the Charon-SSP Manager with the new name. This action renames the configuration directory of the virtual machine and the associated configuration file.

This option is inactive, while the virtual machine is running.

7.5.5.6 Backup VM (Charon-SSP AL Images only)

Use this function to create a ZIP-file of the configuration file, log files and other VM information. When this option is selected, a window opens where storage location and ZIP-file name can be selected. The resulting backup can be copied to a remote system via SFTP (via the user **charon**).

This function does not backup the virtual and physical disks used by the Charon-SSP instance.

7.5.6 Host System Network Configuration

7.5.6.1 General Information

Charon-SSP Manager provides features to configure the following **host system network configuration** aspects:

- Configuring host system IPv4 network interface settings.
- Adding a virtual bridge, i.e., a collection of virtual network tap (TAP) devices that constitute a host-attached virtual LAN. A virtual bridge can be connected to the customer LAN or be internal to the host system.
- Adding VLAN interfaces to a parent Ethernet interface. This allows the host system to participate in the specified VLAN in the customer network.

Please note: The host system network management features of the Charon Manager can be used to prepare a host interface for use by the guest system (for example, by creating a TAP interface or by disabling the IP address configuration). However, **it cannot be used to specify the Solaris network configuration. This must be done on the Solaris level, once the guest system has booted.**

7.5.6.2 Network Settings Tool – Additional Notes and Caveats

Charon-SSP on version 7 of the supported Linux distributions

The host system network configuration functionality requires the *network-scripts* package. Configuring a virtual bridge requires the *bridge-utils* package to be installed on the system. If these packages are not yet installed on your Charon-SSP system use the **yum** command to install them. The packages are preinstalled on Baremetal and Barebone systems and Charon host systems based on prepackaged cloud marketplace images.

Charon-SSP on version 8 of the supported Linux distributions

The *NetworkManager* must be installed and enabled.

NetworkManager requirements

Please read the chapter [NetworkManager Considerations](#). It is important to understand the different behavior of Charon-SSP on Linux 7.x and Linux 8.x.

Charon-SSP in VMware environments

It is recommended to use individual vNICs to provide Ethernet interfaces to guest systems. Connecting a virtual bridge with a vSwitch is not recommended as this can lead to serious network problems in the LAN unless the configuration is very well understood. This does not apply to internal bridges (i.e., virtual bridges not connected to an external interface). Similar considerations apply to other hypervisor environments.

Interface IPv6 configuration

Any IPv6 configuration of the host system network interfaces is not covered by the Network Settings tool of the Charon Manager. IPv6 configurations must be performed manually via the Linux host operating system options. Enabling or disabling IPv6 globally or for specific interfaces can be done using the **sysctl** command and the *sysctl.conf* file.

Sample entries in the *sysctl.conf* file to disable IPv6 for all interfaces:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
```

Sample entry in the *sysctl.conf* file to disable IPv6 for interface eth0:

```
net.ipv6.conf.eth0.disable_ipv6=1
```

Activate the entries in the *sysctl.conf* file:

```
# sysctl -p
```

Please refer to the documentation of your host operating system for more information.

Charon-SSP in a cloud environment

Every cloud environment has specific characteristics that could conflict with interface configurations made manually or via the Charon Manager. Please refer to the documentation provided by the cloud provider and the network-specific sections in the *Getting Started Guide* of your cloud-specific product to understand the networking behavior of your cloud instance **before you change any interface settings**.

Important points that apply to most cloud environments:

- If the Charon host is configured with more than one active IP interface, asymmetric routing can cause connectivity problems. In such cases, policy-based routing (per interface routing tables with associated IP rules) is required.
- Only IP unicast traffic is supported. Non-IP traffic or multicast/broadcast traffic is not supported and requires traffic tunneling.
- Promiscuous interface mode is not supported.
- Only traffic with the MAC address assigned by the cloud provider is allowed across an interface.
- Routing requires special configuration steps (source/destination check disabling) on the cloud instances. Enabling IP forwarding on the Linux host is not enough.
- Cloud specific security rules must allow the relevant traffic. Configuring the Linux firewall correctly is not enough.
- If a host NIC is dedicated to a guest system, the MAC address and IP address assigned to the interface by the cloud provider must be used by the guest.

Please note: older versions of the prepackaged Charon-SSP images available on the different cloud marketplaces were based on CentOS 7.x. Newer versions are based on Rocky Linux 8.x. Hence, please also refer to the corresponding [NetworkManager Considerations](#) section.

7.5.6.3 Network Settings Overview

To open the network settings window, click on **Tools > Network Settings**. This will open a window like the ones shown below. The window on **Linux 7.x** looks slightly different, depending on whether an `ifcfg`-file (in `/etc/sysconfig/network-scripts`) exists for the selected interface or not.

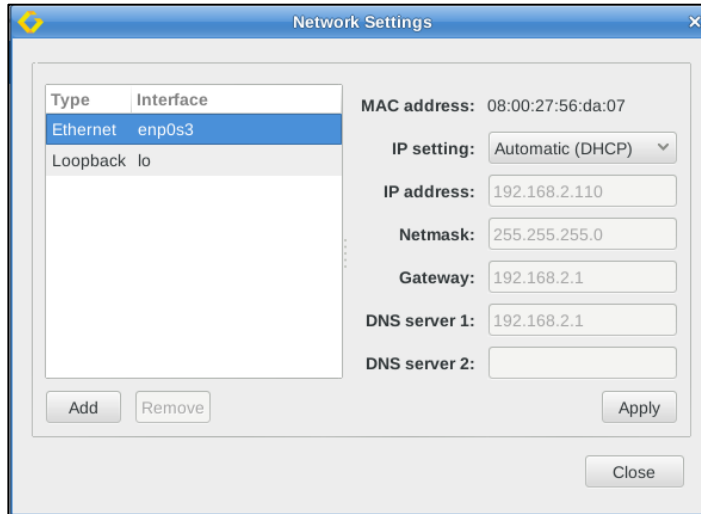


Figure 77: Host network configuration - `ifcfg`-file exists

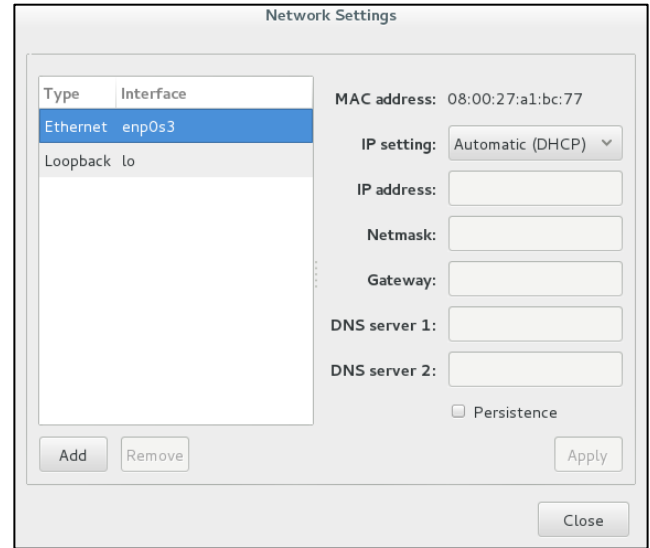


Figure 78: Host network configuration - no `ifcfg`-file

Overview of the network settings window:

- Left-hand side:** list of available host system network interfaces (including bridge and VLAN interfaces created previously).
Please note: **For version 8.x of the supported Linux distributions only interfaces that are under the control of the NetworkManager are shown here.** If an existing interface is missing, it is likely excluded from NetworkManager control. This needs to be changed on the Linux management level, before the interface can be used by the Charon Manager.
- Right-hand side:** settings of the currently selected interface.
- Apply** button: confirms any configuration changes made for the selected interface. On Linux 7.x, greyed-out if no `ifcfg`-file for the interface exists (see option **Persistence**).
- Add** button: opens a submenu where you can select to add a virtual bridge or a VLAN interface.
- Remove** button: allows to remove the selected virtual bridge or VLAN interface.
- Persistence:** Only available on Charon host systems running Linux version 7.x. If no `ifcfg`-file for the selected interface exists in `/etc/sysconfig/network-scripts`, selecting this option creates such a file and removes the interface from NetworkManager control. If an `ifcfg`-file for an interface already exists, the option Persistence is not visible.
Please note: At the time of writing, this option was not available in Charon-SSP VE and cloud-specific AL images based on CentOS 7.x. Therefore, on instances based on such images, the `ifcfg`-file for a new interface must be manually created to ensure that the interface can be properly managed by the Charon Manager. If such a file does not exist, clicking on the **Apply** button may cause an error.

Please refer to the next section for a detailed description of the network configuration options.

7.5.6.4 Managing Host System Network Interfaces

Open the network settings window as described above by clicking on **Tools > Network Settings**.

Using the network settings window, you can set up the existing host system network interfaces according to your requirements. The window also contains previously created bridge and VLAN interfaces.

Important information regarding the default gateway configuration option:

The Network Settings option allows you to configure a default gateway when configuring an interface with manual IP information. Please use this option with care.

- Incorrect settings can make the Charon host unreachable. If running Charon-SSP in a cloud environment, a wrong configuration of this parameter can make your instance permanently unreachable.
- When configuring an *internal* virtual bridge (a bridge with binding interface OFF), you must leave the gateway field empty.
- Version 7.x of the supported Linux distributions: using this option to define a default route on a second NIC, or a bridge connected to a second NIC, overwrites an existing default route via another interface. To restore the original gateway, remove the definition for the second NIC in Charon Manager and restart the network.
- Version 8.x of the supported Linux distributions: using this option to define a default route on a second NIC, or a bridge connected to a second NIC, adds a second default gateway. Removing the definition for the second NIC in Charon Manager also removes it from the active configuration.

To configure an interface, **select the interface** that is to be configured.

After selecting an interface, you can then set the following **host system network interface parameters**:

| Field | Description |
|-------------------|--|
| IP setting | Specify the method used for the IPv4 addressing of the interface. Options are Automatic (DHCP) , Manual , and None . If an interface is to be used for a guest Solaris system running in an emulated SPARC system, use None . This will enable the interface at host system boot without configuring an IP address (the Solaris guest will configure its IP address on the emulated NIC to which this interface will be assigned). It is possible that both, host system and guest system, assign their own IP address to the same interface. This is not recommended, as it will affect network performance. Sharing an interface requires promiscuous mode and the MAC addresses of host and emulated system to be different. It also requires mutual host routes via a default gateway for the Charon host and guest system if they should be able to communicate with one another on a shared NIC. Not an option for cloud environments. |
| IP address | If manual addressing is selected, the host IP address can be added in this field. The field is inactive if DHCP or None is selected in IP settings. |
| Netmask | If manual addressing is selected, the netmask for the host IP address can be added in this field. The field is inactive if DHCP or None is selected in IP settings. |
| Gateway | If manual addressing is selected, a default gateway for the host can be added in this field. The field is inactive if DHCP or None is selected in IP settings. This parameter will set a default route via the interface currently being configured. If left empty, a currently active default route is used. Please read the important information about this option at the beginning of the <i>Managing Host System Network Interfaces</i> chapter! |

| Field (cont'd) | Description |
|---|---|
| DNS server 1 and DNS server 2 | <p>If manual addressing is selected, enter the IP address of one or two DNS name servers. Inactive if DHCP or None is selected in IP settings.</p> <p>If the host system runs Linux version 8.x, and the content of the first DNS server field is not manually configured, this field is populated automatically if an /etc/resolv.conf file with a nameserver exists (created automatically by NetworkManager).</p> |
| Persistence | <p>Only for host systems running version 7.x of the supported Linux distributions: If no ifcfg-file for the interface exists, selecting this option will create the ifcfg-file for the interface and remove the interface from NetworkManager control. If an ifcfg-file for an interface exists (whether under NetworkManager control or not), this option is hidden. For Charon hosts running Linux 7.x, the Charon-SSP Manager can only manage an interface if an ifcfg-file exists. The Apply button is inactive if management is not possible.</p> |

The **Apply** button confirms any changes made and **Close** discards them.

7.5.6.5 Creating a Virtual Network (Virtual Bridge)

A virtual network in Charon-SSP is a virtual bridge on the host system with one or more virtual network interfaces (TAP interfaces) attached to it. The virtual interfaces can be used to provide network interfaces for use by Charon-SSP instances. A virtual network can be connected to the external network using a so-called binding interface, or it can be internal to the host system.

To create a new virtual network, open the network settings window via **Tools > Network Settings**. Then follow the steps shown below:

1. Click on the **Add** button to open the submenu for selecting between virtual networks and VLANS.
2. Select **Virtual Network**.

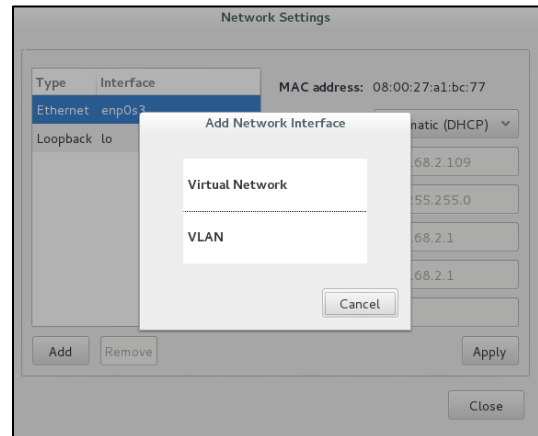


Figure 79: Network type selection

3. This will open the virtual network configuration window.
4. Configure the virtual bridge. The configuration settings are described below.

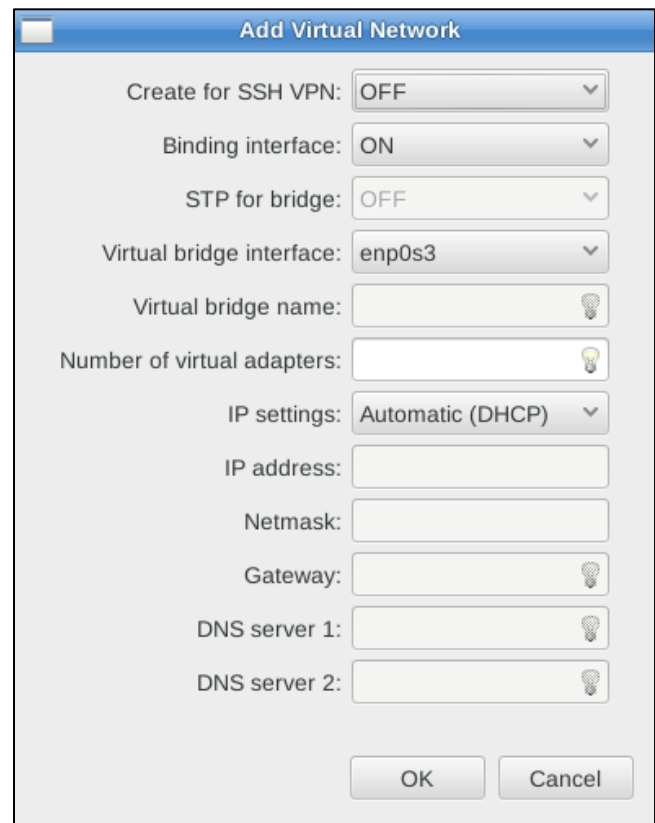


Figure 80: Virtual network configuration

Virtual bridge (i.e., virtual network) configuration options:

| Field | Description |
|--------------------------------------|---|
| Create for SSH VPN | If set to ON , a special virtual network will be created to be used as the basis for creating an SSH VPN tunnel as described in <i>SSH VPN Operation</i> . |
| Binding interface | <p>ON: select a physical interface from the Virtual bridge interface dropdown menu on which the bridge is configured. The bridge is connected to the host system LAN.</p> <p>OFF: enter a user-defined name in the Virtual bridge name field. This name is used in naming the bridge and TAP interfaces instead of using the physical interface name. The bridge is internal to the host system. Always OFF if Create for SSH VPN is enabled.</p> <p>Using a binding interface is not suitable for cloud environments as it requires that multiple MAC addresses are accepted across one physical NIC.</p> |
| STP for bridge | Enable or disable the Spanning Tree Protocol on the virtual bridge. Always OFF if binding interface is set to ON or the selection Create for SSH VPN is enabled. |
| Virtual bridge interface | Dropdown menu to select a physical interface that will provide an external network connection to the bridge. Inactive if the binding interface is disabled. |
| Virtual bridge name | Set a user-defined bridge name if the binding interface is disabled. This name is used in place of the physical NIC name when creating the bridge and TAP interfaces. Inactive if binding interface is enabled. Fixed name vpnX for SSH VPN configuration (X = 0, 1, ...). |
| Number of virtual adapters | Specify how many virtual adapters (TAP interfaces) are needed. |
| IP settings | Specify the method used to set an IPv4 address on the bridge interface. This interface is used to connect the Charon host to the virtual bridge. Options are Automatic (DHCP) , Manual , and None . If binding interface is disabled, manual configuration is mandatory (to assign an IP address for the Charon host to communicate on the internal virtual bridge). |
| IP address | If manual addressing is selected, the host IP address can be added in this field. The field is inactive if DHCP or None is selected in IP settings. |
| Netmask | If manual addressing is selected, the netmask for the host IP address can be added in this field. The field is inactive if DHCP or None is selected in IP settings. |
| Gateway | <p>If manual addressing is selected, a default gateway for the host can be added in this field. The field is inactive if DHCP or None is selected in IP settings. It is also inactive if Create for SSH VPN is selected. This parameter will set a default route via the interface currently being configured. If left empty, a currently active default route is used.</p> <p>Please read the important information about this option at the beginning of the <i>Managing Host System Network Interfaces</i> chapter!</p> |
| DNS server 1 and DNS server 2 | <p>If manual addressing is selected, you can add the IP address of one or two DNS name servers. Inactive if SSH VPN configuration, or DHCP or None in IP settings, is selected.</p> <p>If the host system runs Linux version 8.x, and the content of the first DNS server field is not manually configured, this field is populated automatically if an <code>/etc/resolv.conf</code> file with a <code>nameserver</code> exists (created automatically by NetworkManager).</p> |

The virtual network **connected to a binding interface** consists of

- a bridge device called br_<physical interface>, and
- a series of TAP devices named tapX_<physical interface>.

If the **binding interface is disabled**, the virtual network consists of

- a bridge called br_<bridgename>, and
- a series of tapX_<bridgename> TAP devices.

If **SSH VPN is enabled**, the first virtual network created consists of

- a bridge called br_vpn0,
- a tap0 interface (VPN tunnel endpoint), and
- a series of tapX_vpn0 interfaces

X is a number from 0 up to the number of virtual adapters (0 to configured number minus 1) specified in **Number of the virtual adapters**. These devices can then be configured for use as virtual Ethernet controllers.

Should you create a virtual bridge manually for any reason, please adhere to the naming conventions outlined above. Otherwise, Charon-SSP will not recognize the correct interface type.

7.5.6.6 Deleting a Virtual Network

To delete a virtual network, follow the instructions listed below.

- Follow the menu path **Tools > Network Settings** to open the network settings window.
- **Select the bridge** you want to delete and click on the **Remove** button. This will open a confirmation window.
- To delete **all** virtual network interfaces associated with the selected bridge, **click on YES**.

Following the instructions above will immediately delete all TAP devices and the bridge.

Please note: if the bridge was connected to a binding interface (i.e., a host system NIC), the configuration of the bridge will be transferred to the NIC when deleting the bridge.

7.5.6.7 Resizing a Virtual Network

To resize a virtual network, follow the instructions listed below.

| Step | Description |
|------|---|
| 1 | Shut down any running guest operating systems and stop all virtual machines connected to the virtual network TAP devices. |
| 2 | Delete the current virtual network, using the instructions detailed in Deleting a Virtual Network . |
| 3 | Re-create the virtual network using the instructions detailed in Creating a Virtual Network . Make sure to specify the new virtual network size in the Number of the virtual adapters field. |
| 4 | Reconfigure the Ethernet configuration of the virtual machines. This step is only necessary if shrinking the virtual network and only if the virtual machines are configured for TAP devices that no longer exist. |
| 5 | Start the attached virtual machines. |

Shrinking a virtual network may make it necessary to adjust several virtual machine configurations because the name of their virtual Ethernet interface has changed.

7.5.6.8 Adding a VLAN interface

To add a VLAN interface to a parent Ethernet interface of the host system, open the network settings window via **Tools > Network Settings**. Then follow the steps shown below:

1. Click on the **Add** button to open the submenu for selecting between virtual networks and VLANs.
2. Select **VLAN**.

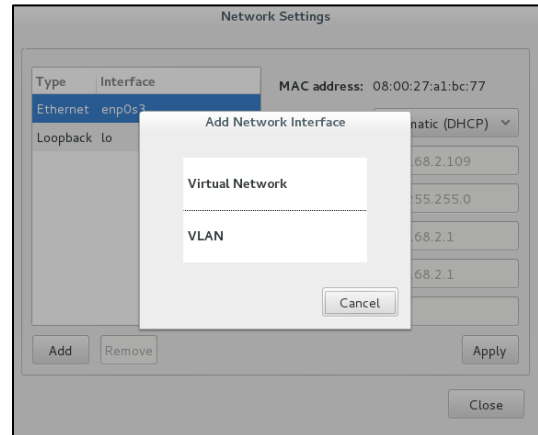


Figure 81: Network type selection

3. This will open the VLAN configuration window.
4. Configure the VLAN interface. The configuration settings are described below.

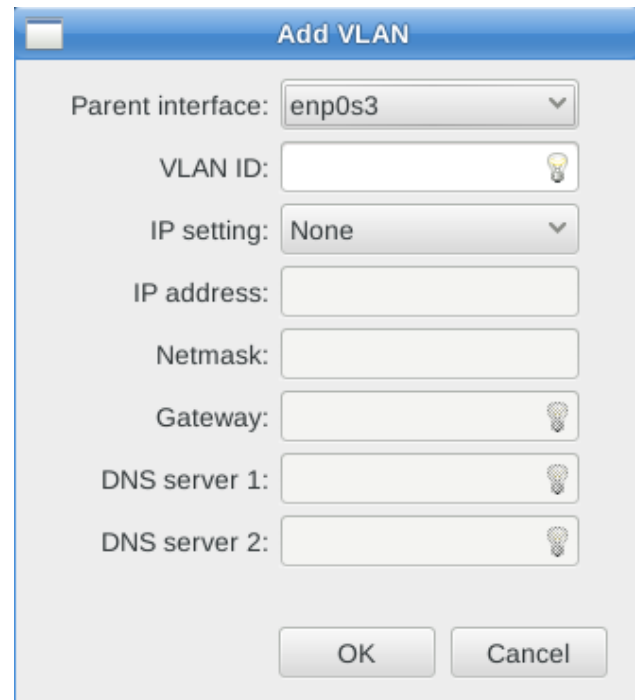


Figure 82: VLAN configuration

VLAN configuration options:

| Field | Description |
|--------------------------------------|--|
| Parent interface | Select the host system Ethernet interface that will serve as the base interface for the LAN connection. |
| VLAN ID | Enter the VLAN number matching the customer's LAN configuration. Values: 2-4094. The interface name of the new interface has the format: <i><parent-interface>.<vlan-id></i> |
| IP settings | Specify the method used for addressing the interface used to connect the host to the external network. Options are Automatic (DHCP) , Manual , and None . |
| IP address | If manual addressing is selected, the host IP address can be added in this field. The field is inactive if DHCP or None is selected in IP settings. |
| Netmask | If manual addressing is selected, the netmask for the host IP address can be added in this field. The field is inactive if DHCP or None is selected in IP settings. |
| Gateway | If manual addressing is selected, a default gateway for the host can be added in this field. The field is inactive if DHCP or None is selected in IP settings. This parameter will set a default route via the interface currently being configured. If left empty, a currently active default route is used. Please read the important information about this option at the beginning of the <i>Managing Host System Network Interfaces</i> chapter! |
| DNS server 1 and DNS server 2 | If manual addressing is selected, you can add the IP address of one or two DNS name servers. |

7.5.6.9 Deleting a VLAN Interface

To delete a VLAN interface, follow the instructions listed below.

- Follow the menu path **Tools > Network Settings** to open the network settings window.
- Select the VLAN interface you want to delete and click on the **Remove** button. This will open a confirmation window.
- To delete the VLAN interface, click on **YES**.

Following the instructions above will immediately delete the VLAN interface.

7.5.7 Miscellaneous Management Tasks

The following sections describe some additional functions provided by the Charon-SSP Manager interface that may be useful in certain instances.

- Gathering Host Information
- Adding an Existing Virtual Machine to Charon-SSP Manager
- Determining the Charon-SSP Manager Version
- Modifying the Charon-SSP Agent Preferences

7.5.7.1 Displaying Host Information

To view the details of the system hosting the Charon-SSP instance, follow the menu path **Tools > Host Information** to open a window like the one below.

This window provides details of the host system's hardware configuration and operating system version.

Please note: for a correct display of the Ethernet interface list, the **pciutils** package must be installed on the Charon host.

See example on the right:

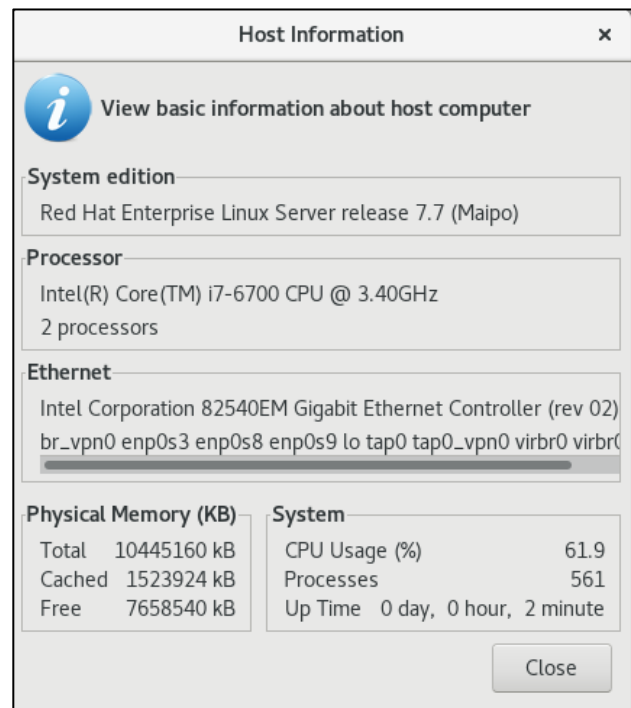


Figure 83: Host information display

7.5.7.2 Adding an Existing Configuration to the Charon-SSP Manager

To add an existing virtual machine configuration to the Charon-SSP Manager, you can use the **Import** function. This function is available in the **Virtual Machine** menu (when **Home** is selected), on the **Welcome** page of the Charon-SSP Manager, and in the context menu of the virtual machine pane when no Charon-SSP instance is selected.

The **Import** function lets you select an existing Charon-SSP virtual machine configuration and a name for the newly added system. Alternatively, you can use the **import-config** command-line tool (see [import-config utility](#)).

Please note:

- The imported configuration may have to be adapted to the possibly different environment on the new host system. For example, the path to the virtual storage container files or the names of network devices may be different when compared to the previous environment.
- Server JIT and use of the plus-version (4U+/4V+) cannot be imported via the GUI in this version. Please set them manually after import or use the **import-config** command-line tool. The same is true for the autoboot settings.
- An existing nvram file (<configname>.dat) is not imported. NVRAM settings may have to be corrected after import, or the file has to be copied manually to the correct location in the Charon Agent hierarchy.

7.5.7.3 Determining the Charon-SSP Manager Version

To display the version of Charon-SSP Manager currently running, select **Help > About** from the menu bar. This will open a window displaying the version of the software.

7.5.7.4 Modifying the Charon-SSP Agent Preferences

To modify the preferences maintained by the Charon-SSP Agent software, follow the menu path **Virtual Machine > Preferences** to open a window like the one shown below.

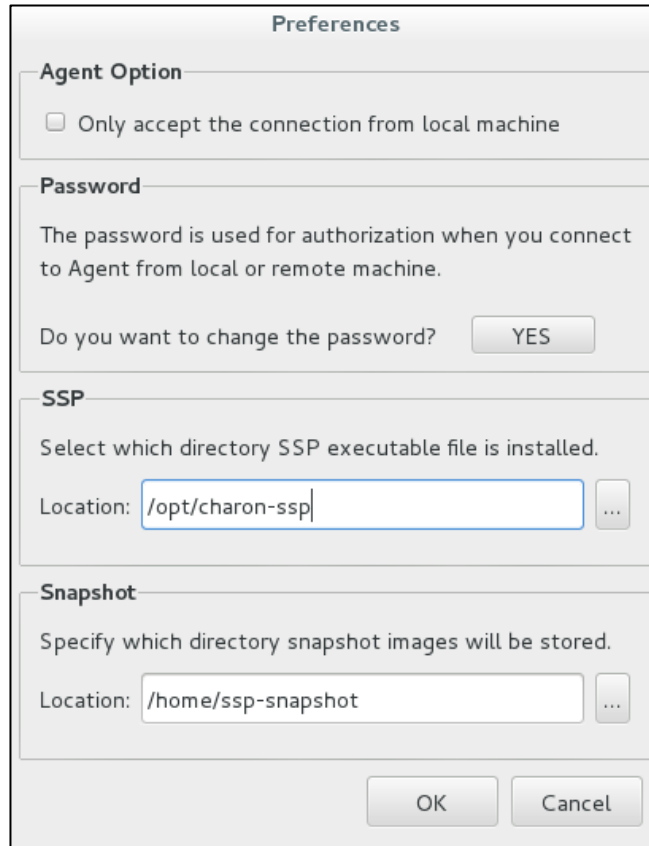


Figure 84: Agent preferences window

The preferences window offers the following configuration options:

- To limit the access to the Charon-SSP agent to the local system, check the box under **Agent Option**.
- The **password** to be used by the Charon-SSP Manager to connect to the current Charon-SSP Agent can be modified by **clicking** on the **YES** button next to **Do you want to change the password?** This will open a change-password dialog in which you can enter and confirm the new password. On **Baremetal systems**, this will change all management and user passwords of the host system.
- **Products other than Charon-SSP AL:** It is possible to alter the root location of the Charon-SSP executable images. This might be useful, for example if multiple versions of Charon-SSP have been installed. In this case, this can be used to switch between them. To change the root location, alter the pathname in the field **Location**. This configuration does not actually move any files. It only changes the pointer to the directory.
- The **Snapshot** parameter shows where currently the resulting files are stored if a Charon-SSP virtual machine is suspended. On Charon-SSP AL cloud images, this path cannot be changed.

7.5.7.5 Setting Console Options

The way the Charon-SSP Manager displays the built-in console can be influenced by using the console options configuration.

To open the configuration window, select **Virtual Machine > Console Options**. This displays a window like the following:

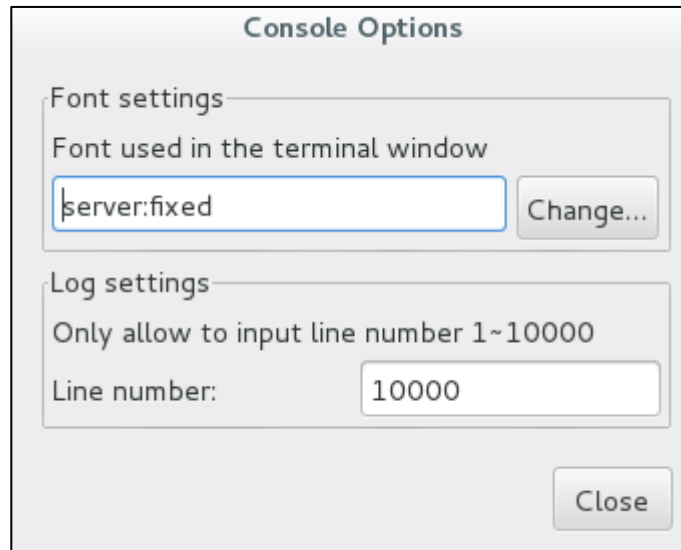


Figure 85: Console options configuration

The configuration window contains two configuration options for the built-in console:

- **Font settings** allow selecting a different font to use for displaying the console output. Click on the **Change** button to select the desired font from a menu. To activate the new setting for any active embedded console windows, you have to restart the Charon Manager.
- **Log settings** allow selecting the number of lines cached for the console display area in the Charon-SSP Manager. When the virtual machine is stopped, the console display tab shows the cached lines of console output for this machine. This log setting is not related to defining a log file in the TTYA and TTYB configuration.

7.5.7.6 Retrieving a Charon-SSP Core Dump

If Charon-SSP instances encounter an unrecoverable problem, they may write a core dump. The user may be instructed by Stromasys support to retrieve such a core dump and make it available to Stromasys for troubleshooting.

This description **applies to systems running the Charon-SSP software including the Charon-SSP Agent** (default Baremetal and Barebone systems). Administrators of systems without a running Charon-SSP Agent who want to enable core dumps for Charon-SSP instances must configure their own environment.

At startup, the agent will configure the core-dump location (using `/proc/sys/kernel/core_pattern`) depending on the Charon-SSP variant (to verify the location, check `/opt/charon-agent/ssp-agent/agent-log/agent.log`):

- Conventional, RPM-based Charon-SSP installation and VE cloud images: `/home/`
- Barebone Charon-SSP installation: `/home/charon`
- Baremetal Charon-SSP installation: `/charon/storage`
- Charon-SSP AL images: `/charon/storage`

The name of the core dump written to the configured location consists of several components:

Example: `core-ssp4m-11-1514348663.5048`

In the above example

- `ssp4m` is the name of application,
- `11` is the signal causing the core dump,
- `1514348663` is the Unix tick time since 1970, and
- `5048` is the process ID of affected process.

Copying the file to another system or another device:

- On a **conventional** or cloud-based system:
 - Become the root user because the written core-dump file is owned by the root user and not readable by other users.
 - Either copy the file to another system using SFTP, or
 - if physical access to the system is possible, attach a USB device to the system, mount it, copy the file to the USB device, unmount, and remove the device.
- On a **Baremetal** system:

Either

 - Attach a USB device to the system.
 - Use the **Storage Manager** to mount it.
 - Use the **File Manger** to copy the core dump file to the USB device.
 - Use the **Storage Manager** to unmount the USB device.
 - Remove the USB device from the system.

Or

 - Connect to the system using SFTP (user: **charon**) and retrieve the core dump file.

7.5.8 Special Baremetal and Charon-SSP AL Tools

When the **Charon-SSP Manager** is connected to a **Baremetal system** or a **cloud-specific Charon-SSP AL (Automatic Licensing) system**, the Charon-SSP Manager **Tools** menu shows additional tools in the **Tools > Charon Baremetal** or **Tools > cloudname Cloud** menu. These tools are:

- File Manager
- Storage Manager
- Time & Date
- SFTP Server
- SSH public key (Baremetal only)

7.5.8.1 File Manager

The file manager allows the user to manage files and directories in the data area of the Charon-SSP host system. The image below shows an example of a file manager window:

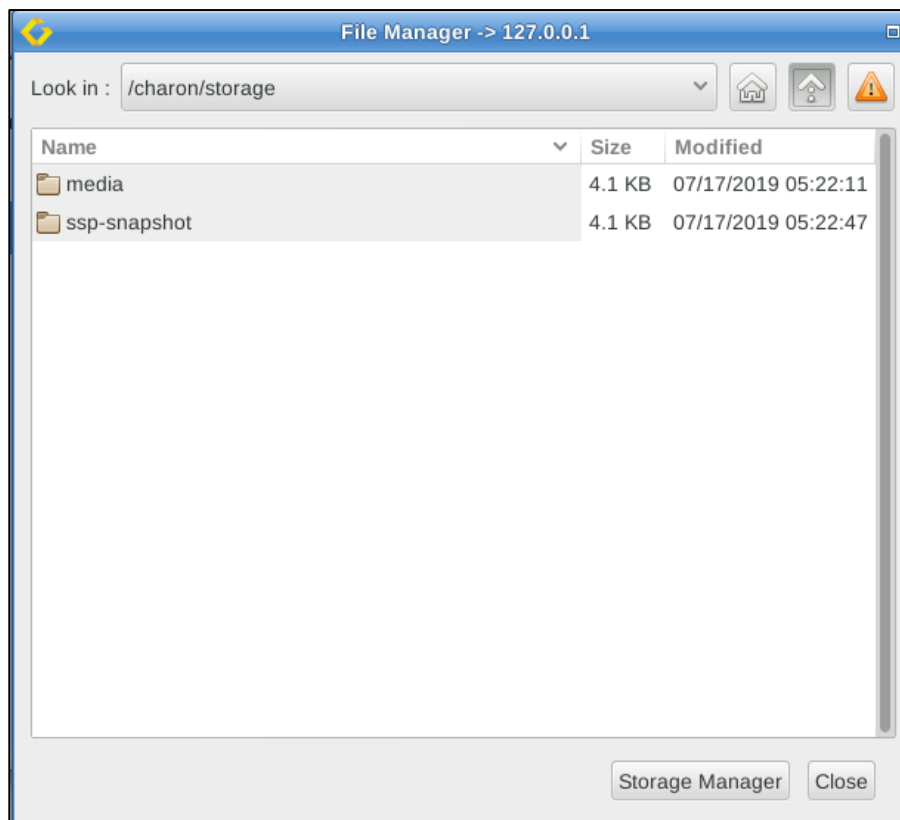


Figure 86: Baremetal file manager

A **right-click** in the window opens a **context menu** that provides access to basic file management tasks:

- Create a new folder
- Cut, copy, and paste files and folders
- Delete files and folders
- Rename files and folders

On Charon-SSP AL, the File Manager provides access to the **/charon/storage** hierarchy to which the Charon Manager on such systems has access.

The buttons at the bottom of the window allow closing the file manager or opening the storage manager. The triangle at the top right shows relevant alerts if any such alerts exist.

7.5.8.2 Storage Manager

The storage manager allows the user to manage storage devices connected to the Charon-SSP host system. The image below shows an example of a storage manager window:

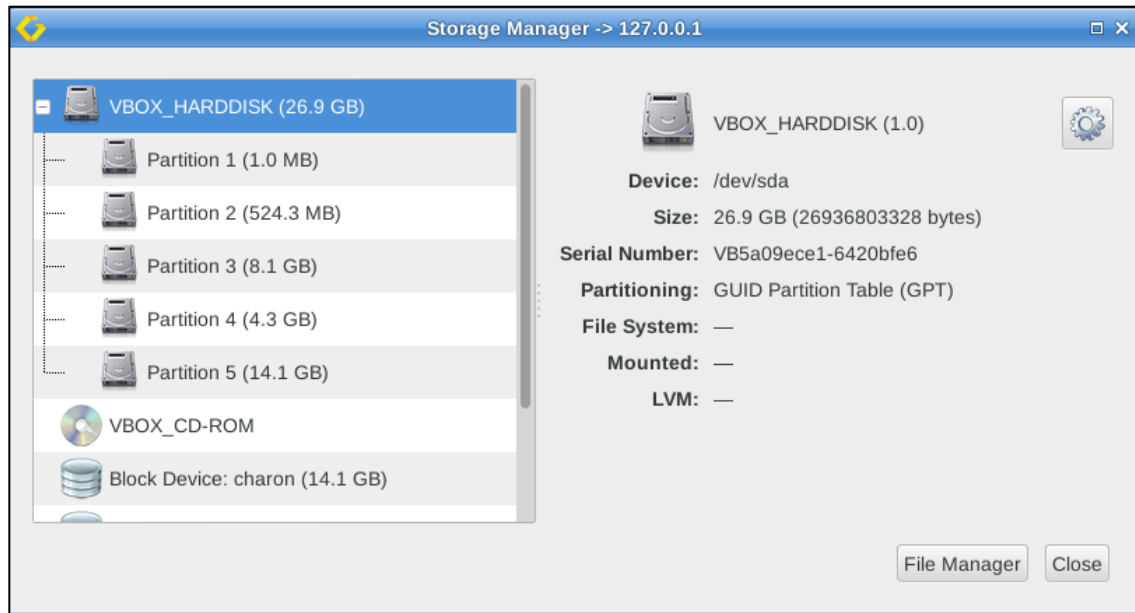


Figure 87: Baremetal storage manager

A **right-click** on a device/partition (or clicking on the cog-wheel) will open a **context menu** enabling the following tasks:

- Adding a device to the Charon-SSP host system data storage (Baremetal only)
- Detaching a device from the Charon-SSP host system data storage (Baremetal only)
- Mounting and unmounting a file system (for example, if a USB device or an additional is attached to the system). The Storage Manager mounts devices under `/charon/storage/media/<device-id>` where `device-id` is the UUID of the device or partition, or the filesystem label.
- Formatting a volume

Using the buttons at the bottom of the window, the Storage Manager can be closed, or the File Manager can be opened. Once a volume has been mounted via the Storage Manager, it will be mounted automatically after the host system has been stopped and started.

The Storage Manager looks slightly different, depending on which host system is used. The images below show some samples of cloud-specific **Charon-SSP AL systems**:

AWS-specific example:

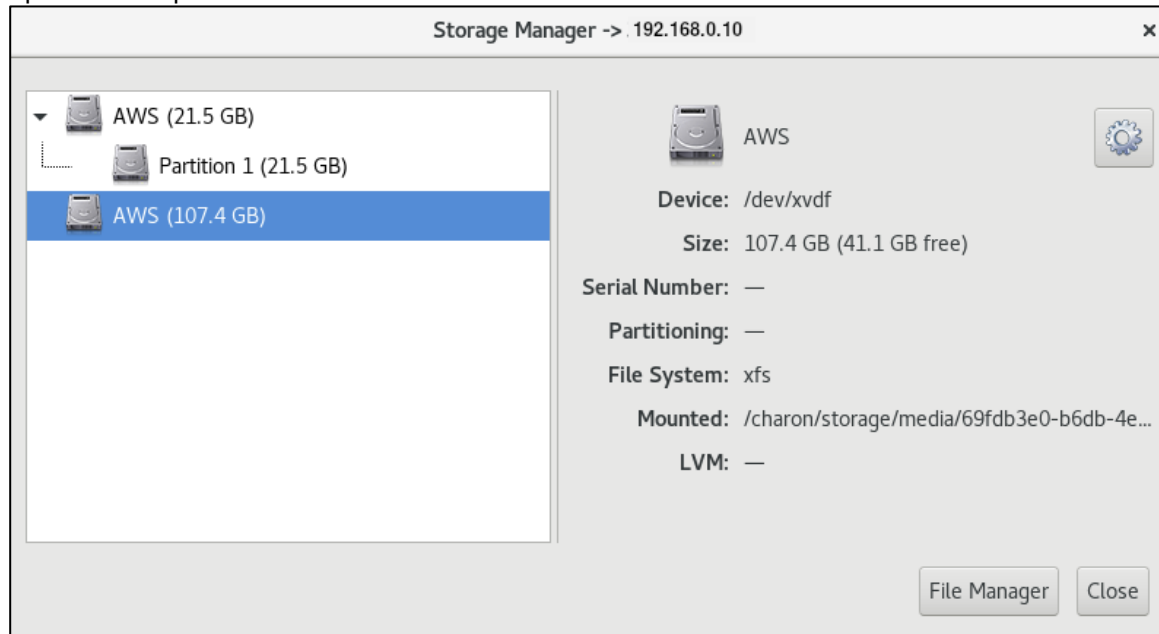


Figure 88: Storage Manger Charon-SSP AL for AWS

OCI-specific example:

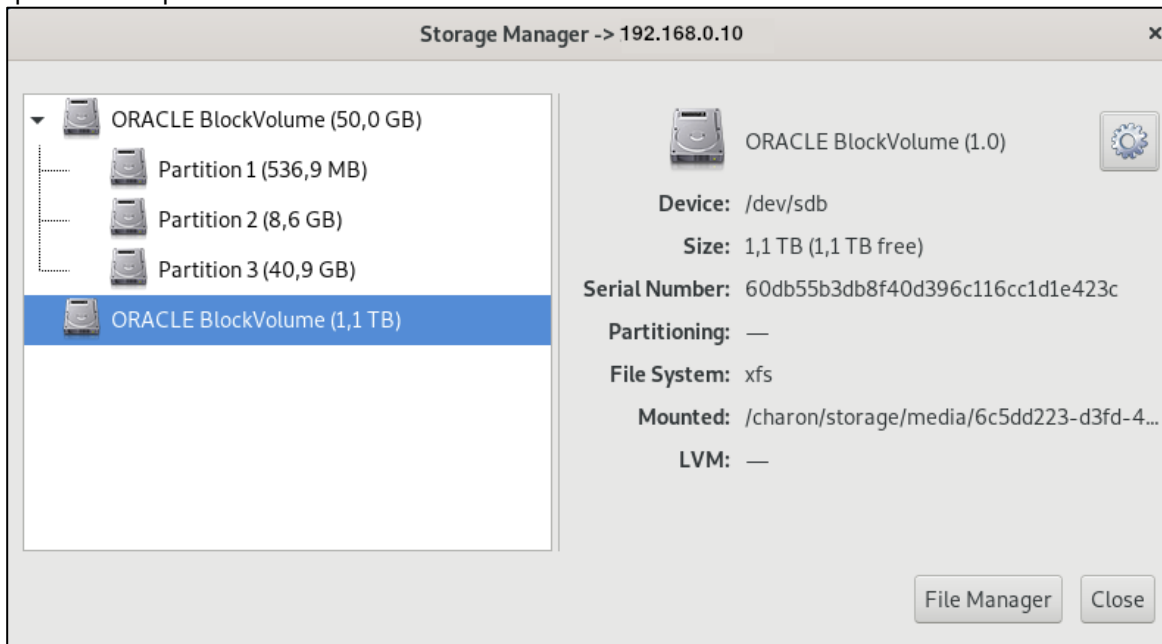


Figure 89: Storage Manger Charon-SSP AL for OCI

7.5.8.3 Setting Time and Date

The **Time & Date** option allows setting the time and date of the Charon-SSP Baremetal or cloud-specific Charon-SSP AL system via Charon-SSP Manager. The following image shows the available options:

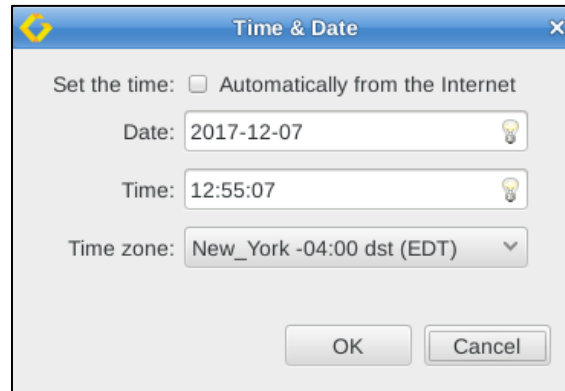


Figure 90: Baremetal time and date menu

In this window you can

- enable NTP or manually set time and date, and
- configure the time zone.

7.5.8.4 SFTP Server

This option allows to configure the login method and the host system address used for SFTP:



Figure 91: Baremetal SFTP server

In the configuration window, the following options can be selected (if supported by the host system):

- **Authentication:** Public Key or Password authentication. For Public Key authentication, the public key must first be uploaded to the Baremetal host system (see below). For cloud-specific Charon-SSP AL systems, only Public Key is supported.
- **IP Address:** if the host system has several usable IP addresses, select the correct address to connect to via SFTP.

The username is fixed to the preconfigured user **charon**.


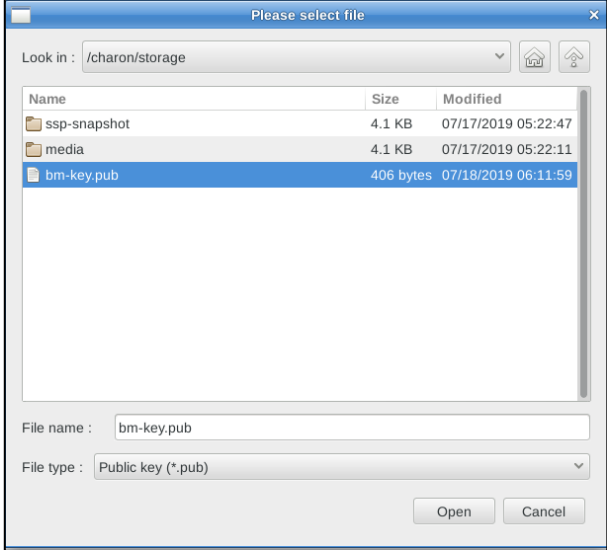

7.5.8.5 Baremetal SSH Public Key Import

The option **SSH public key** allows the user to import a new SSH public key. The key is used to create SSH VPN tunnels, and to enable SFTP (user **charon**) and SSH (user **sshuser**) logins without providing a password. The upload and import can be performed from a local or a remote Charon-SSP Manager.

If you have not created and SSH keypair yet, please refer to [Creating and Uploading the Public SSH Key](#).

If using the local Charon-SSP Manager on the Baremetal system to import the public SSH key created on the remote system, the public key must be first copied to the Baremetal host system via **SFTP** using the **charon** account. On versions before Charon-SSP 4.0.x, the uploaded file must be world-readable.

The following table shows the detailed steps required to import a public SSH key:

| Step | Steps to import SSH public key | |
|------|--|--|
| 1 | Select the menu option Tools > Charon Baremetal > SSH public key. The following screen will be opened. |  |
| 2 | <ul style="list-style-type: none"> Click on ... to open a file browser and to select the public key file (must be world-readable in older versions). Click on Open. This will take you back to the previous window where the selected key will now be shown in the Public key field. In this window, click on Apply. |  |
| 3 | After the key has been imported, you will receive a success message as shown here. Should there be a problem with importing the key, you will receive an error message. |  |

7.6 The Charon Management Password

Depending on the product variant used, the management password handling can be different.

7.6.1 General Information about the Management Password

What is the Charon management password?

This is the password required on all product variants to connect to the Charon Agent with the Charon Manager. Without it, the management GUI does not work. The default password is **stromasys**.

Additional use cases of the Charon management password:

On Baremetal product variants, this password is the general password for all system-generated, user-visible accounts and functions that require a password. The same is true for instances created using pre-packaged Charon-SSP cloud images. However, usually SSH key-based login is implemented during instance launch making the password less important.

7.6.2 Where and How to Set the Charon Management Password

The Charon management password can be set in different ways:

- **All product variants:**
 - If the default password has not been changed yet, it must be changed when first using the Charon Manager to connect to the Charon host system.
 - The password can be set via the Agent preferences ([Modifying the Charon-SSP Agent Preferences](#)).
 - The password can be set running the command `/opt/charon-agent/ssp-agent/utils/charon-passwd`.
- **Baremetal product:** the password can also be set during the installation.

On **Baremetal**, the newly set password will be valid for all system-created, user-visible accounts and functions that require a password.

Please note:

After setting the password, this management password is then valid for all subsequent logins by the same or a different user until it is changed again. It is not removed if Charon-SSP is reinstalled. Note that **older versions** of the product will not prompt for the password at first login but will use a default password (**stromasys**).

7.6.3 Resetting a Forgotten Management Password

If you forgot your management password, you cannot recover the password, but you can reset it.

7.6.3.1 Password Reset using the Command-Line

To reset a forgotten password, **root access** to the Charon-SSP host system is required.

- **On Charon-SSP version 4.0.x and higher**, perform the following steps:
 1. Log in as the **root** user.
 2. Change to the correct Charon Agent directory:
`cd /opt/charon-agent/ssp-agent/utils`
 3. Run the command to set the Charon password:
`./charon-passwd`
 4. You will be prompted to enter and confirm your new password.
- **On older Charon-SSP versions**, perform the following steps:
 1. Log in as the **root** user.
 2. Delete the file `/opt/charon-agent/ssp-agent/etc/passwd.conf`.
 3. Restart the Charon Agent:
`/etc/init.d/charon-agentd-ssp restart`
 4. Re-connect with the Charon Manager. The password should be back to default (**stromasys**).

7.6.3.2 Password Reset on a Baremetal System

On Baremetal, changing the management password using the methods described above will change the password for all system-created, user-visible accounts and functions that require a password. If the management password is lost, this also affects the root password. This makes the password recovery process more complicated. If your management password has been lost, you can use the method described below to reset the password.

This method normally preserves the existing Charon-SSP configuration and emulator data on the host system. However, to protect your data from unexpected errors, use this method only if you have up-to-date backups of your Charon-SSP configuration and any vdisks and vtapes used on the system!

If, after reading the description, you feel unsure about the procedure to be performed, do not perform it. Contact your Stromasys VAR or Stromasys support (depending on your service contract) for advice.

Password reset procedure on Charon-SSP Baremetal:

As described in the Baremetal installation and upgrade chapters, Charon-SSP Baremetal can be upgraded by using the ISO installation method. This will preserve the Charon-SSP configuration and emulator data. You can use this method also to reset your password:

1. Cleanly shutdown all running guest systems on the Charon host and stop the associated emulators.
2. Boot the Baremetal system from a Charon-SSP Baremetal installation medium of the same Charon-SSP Baremetal version currently running on the system.
3. Select the current system disk as the installation target.
4. Set the new management password (US-keyboard layout!).
5. Start the installation.

At the end of the installation, the system reboots and you should be able to start the Charon Manager with the new password. Customer-specific additions and configuration changes are lost using this method and must be restored. Please refer to the Charon-SSP Baremetal installation and upgrade chapters for more information.

7.7 Using Charon-SSP from the Command-Line

If you want to start a Charon-SSP instance from the command-line and later use Charon-SSP Manager to manage the instance, you must give the Charon-SSP instance an alias that is the same as the virtual machine name in the Charon-SSP Manager. Otherwise, the Charon-SSP Manager will not recognize the running Charon-SSP instance.

7.7.1 Emulator Command-Line Options

7.7.1.1 Program Name

The **executable programs** to run Charon-SSP from the command-line are listed below:

- **ssp4m:** Charon-SSP/4M 32-bit SPARC V8, Sun-4m architecture virtual machine
- **ssp4u** or **ssp4u-plus:** Charon-SSP/4U(+) 64-bit SPARC V9, Sun-4u architecture virtual machine
- **ssp4v** or **ssp4v-plus:** Charon-SSP/4U(+) 64-bit SPARC V9, Sun-4v architecture virtual machine
- **ssp4u-jit, ssp4u-plus-jit; ssp4v-jit, ssp4v-plus-jit:** these images implement the *second level of DIT optimization* available in Charon-SSP/4U(+) and Charon-SSP/4V(+).

All the executable programs are in `/opt/charon-ssp` under the respective architecture directory:

- `/opt/charon-ssp/ssp-4m/<imagename>`
- `/opt/charon-ssp/ssp-4u/<imagename>`
- `/opt/charon-ssp/ssp-4v/<imagename>`

Make sure you use the correct image for your emulator configuration.

7.7.1.2 Syntax

The syntax of the commands is specified below:

```
# <image-name> <options and values>
```

7.7.1.3 Description

The Charon-SSP emulator instance can be started in four different modes from the command-line:

- In **utility** mode, it is possible to specify the **-l**, **-s**, and **-k** options to list and terminate running instances.
- In **both foreground and background** mode, the Charon-SSP executable can be used to run a SPARC virtual machine. The difference between the two is that in background mode the virtual machine runs as a daemon releasing the controlling terminal.
- In **interactive** mode, a virtual machine and an interactive session are started for debugging purposes. **This mode is for use by Stromasys only. Its options are not documented in detail in this guide.**

Important notes when using Charon-SSP from the command-line:

- Some older versions of Charon-SSP do not work correctly with a relative path to the configuration file. In such cases, use **absolute file paths** when specifying the configuration file.
- Use absolute file paths inside the configuration file.
- Starting the emulator while logged in over the network may prevent access to the license (local dongle). In this case start the emulator in **daemon mode**, or use `ssh root@localhost <emulator-command> -f <path-to-configuration-file>` as a workaround.

The following table describes all the **options that can be passed to the emulator image** from the command-line.

| Option | Description |
|---|---|
| -a <i>alias_name</i> --alias= <i>alias_name</i> | Assign an alias to the new virtual machine instance. This option can be very useful when attempting to locate a specific instance in the list reported by the -l option. If this option is not specified, a name of the form instance-NN will be assigned automatically, where NN consists of two digits from 0 - 9. A virtual machine instance started via the Charon-SSP Manager shows its configured name as the alias. |
| -d --daemon | Run the virtual machine as a daemon. This option cannot be combined with -i . This can be used to start the emulator when logged in over the network which may prevent proper license access otherwise. |
| -f <i>config_file</i> --config= <i>config_file</i> | When starting a new instance, use this option to specify the path and name of the virtual machine configuration file. For further details about the format of this file, see the <i>Configuration File Reference Appendix</i> . This is a mandatory argument, unless one of the options -h , -k or -l is used. Please specify the absolute path of the configuration file because not all versions of Charon-SSP understand relative paths. |
| -h --help | Displays a brief usage message. |
| -i --interactive | Start emulator in interactive mode. This option provides access to the SSP CLI. This CLI provides some commands to manage the emulator that are interesting to end-users. There are also commands that are reserved for use by Stromasys support. Do not use these commands unless advised to do so by Stromasys support. See Charon-SSP CLI for more information. This option cannot be combined with -d . |
| -k <i>pid</i> --kill= <i>pid</i> | Stop the virtual machine instance specified by <i>pid</i> . Use the -l to determine the process id of the relevant instance. This option cannot be combined with any of the other options. Bear in mind that this command will not shut down the guest operating system cleanly. |
| -l --list | This option lists the currently running Charon instances. It cannot be combined with any other option. The list consists of the following columns: <ul style="list-style-type: none"> • pid – process id of the virtual machine, • alias – instance alias specified by the -a option at startup, • start time – timestamp indicating when the virtual machine instance was started, • log time – timestamp indicating last event, • log code – descriptive code indicating the type of event, and • name – name of Charon-SSP image. |
| -p --pause | Start the Charon-SSP instance, but do not boot the guest operating system (no autoboot). |
| -s <i>name</i> --stop= <i>name</i> | Stop the virtual machine instance specified by <i>name</i> . Use the -l to determine the name of the relevant instance. This option cannot be combined with any of the other options. Bear in mind that this command will not shut down the guest operating system cleanly. |

| Option | Description |
|---|--|
| -S path --snapshot=path | Specify path to store snapshot files resulting from suspending a running emulator. A running emulator instance can be suspended via the Charon Manager or the command-line (using the command <code>kill -SIGTSTP <ssp-pid></code>). |
| -U bus:port --usb=bus:port | Connect USB storage device to guest operating system by specifying the USB bus and port number |
| -v or --version | Displays the Charon-SSP version. |
| --auto-boot?[=value] | Set parameter auto-boot? to true or false . When no value specified, and this is the last command-line parameter, the command displays the current setting of the parameter (the virtual machine will not be started). |
| --boot-device[=device] | Set default boot-device. When no value is specified, and this is the last command-line parameter, the command displays the current value of the parameter (the virtual machine will not be started). |

If **no option** is provided, the command will show the help page.

If the command is started with only the filename parameter, it will start in **foreground mode**.

7.7.1.4 Exit Status

The virtual machine executable images exit with **0** on success and **255** if an error occurs.

7.7.1.5 Examples

If the configuration file is set up without using Charon-SSP Manager, the template files provided (*.cfg under `/opt/charon-ssp[ssp-4u | ssp-4v | ssp-4m]`) can be used as a starting point.

The following is a very basic example configuration file.

| Example SUN-4M configuration |
|--|
| <pre>[system] machine = "SUN-4M" [ram] size = 64 [nvram] path = /opt/charon-agent/ssp-agent/ssp/sun-4m/4M/4m.dat [ethernet] interface = eth0 [ttya] type = terminal [log] severity = info destination = console path = /opt/charon-agent/ssp-agent/ssp/sun-4m/4M/4m.log</pre> |

The following command can be used to start the virtual machine and connect to the console (console type **terminal** indicates that the current terminal is to be the console).

```
# /opt/charon-ssp/ssp-4m/ssp4m -f /path/4m.config
```

This generates output like the following on the current terminal:

```

Charon-SSP virtual machine output
*****
Charon-SSP/4M - sun4m VM v4.1.18
Copyright (C) 1998-2019 Stromasys S.A. All Rights Reserved.
*****

[lines removed]

2019-01-01 09:03:09 INFO VM DIT is ON
2019-01-01 09:03:09 INFO NVRAM Initialize NVRAM with ./4m.nvram.....
2019-01-01 09:03:09 ERROR NVRAM Can not open ./4m.nvram
2019-01-01 09:03:09 INFO NVRAM Initialize NVRAM with ./ssp4m_default.dat.....
2019-01-01 09:03:09 INFO Memory Allocating 67108864 bytes memory from system...
*****

Charon-SSP/4M - sun4m VM v4.1.18
Copyright (C) 1998-2019 Stromasys S.A. All Rights Reserved.
*****

CPU_#0 TI, TMS390Z50(3.x) 0Mb External cache

CPU_#1 ***** NOT installed *****
CPU_#2 ***** NOT installed *****
CPU_#3 ***** NOT installed *****

>>>> Power On Self Test (POST) is running .... <<<<<

sun4m/SPARC V8 (1 X 390Z50), No Keyboard
Emulate OBP Rev. 2.25, 64 MB memory installed, Serial #12648430.
Ethernet address 38:fa:31:6e:b6:71, Host ID: 72c0ffee.

Type help for more information
ok

```

7.7.2 Charon-SSP CLI

The Charon-SSP CLI enables a user to send commands to a running emulator. The Charon-SSP CLI offers two types of commands:

- Commands to manage certain aspects of a running emulator. These commands can be useful to end-users.
- Debugging commands for troubleshooting severe emulator problems. **Do not use these commands without being advised to do so by Stromasys support.**

7.7.2.1 Accessing and Exiting the Charon-SSP CLI

There are two ways to access the Charon-SSP CLI:

- Starting the emulator in interactive mode. This requires the restart of the emulator once to start the emulator in interactive mode, and again to stop this mode. Hence, **this mode is mostly suited for testing purposes.**
 - To start the emulator in this mode, use the `-i` or the `-interactive` command-line switch. After successful emulator start, you will see the **Charon-SSP>** prompt.
 - To exit the CLI, you have to stop the emulator with the command `quit`.
- Access via Unix Domain Socket (UDS). The socket name is displayed in the emulator log at startup. This enables access to the SSP CLI of a running emulator without the need to restart it. Hence, this mode is suited for using the commands relevant to end-users.
 - To start the Charon-SSP CLI via UDS, use **one of the following two commands**:
 - `# nc -U /var/run/ssp-linux/sock.<vm-name>`, or
 - `# socat - UNIX-CONNECT:/var/run/ssp-linux/sock.<vm-name>`
 - To exit the CLI, press CTRL+C. The emulator will continue to run.

Should the commands above not be available, they can be installed from the standard repositories. The packages are *nmap-ncat* and *socat* respectively.

7.7.2.2 Charon-SSP CLI Available Commands

The following section shows the commands currently available in the SSP CLI.

7.7.2.2.1 Commands Relevant for End-Users

Please note: the commands described in this section can change the emulator configuration. However, these changes are not persistent, i.e., they will not be preserved across an emulator restart. To preserve them across emulator restarts, you must change the emulator configuration via the Charon Manager or by editing the configuration file.

7.7.2.2.1.1 The *help* Command

The **help** command displays the available commands with a short usage description.

Syntax:

```
help { command }
```

Parameters:

command: the command for which help should be displayed. If no individual command is specified, a list of all commands will be displayed.

7.7.2.2.1.2 The *log* Command

The **log** command allows the user to display and to change the logging severity level of the running emulator.

Syntax:

```
log { severity }
```

Parameters:

severity: possible values: empty, **debug**, **info**, **warning**, **error**, or **fatal**.

If the command is run without a severity specified, the current setting will be displayed.

Example:

The following example shows the initial log severity configuration, set the severity level to debug, and shows the result.

```
Charon-SSP> log
Log severity is: info

Charon-SSP> log debug
Log severity has been changed to debug.

Charon-SSP> log
Log severity is: debug
Charon-SSP>
```

7.7.2.2.1.3 The *hasp* Command

The **hasp** command allows the user to display and to change the currently configured license IDs and their roles.

Syntax:

```
hasp { role { license-id } }
```

Parameters:

- If run without parameters, the current configuration will be shown.
- *role*: possible values are **regular** or **backup** for the primary and backup license respectively. If run without the parameter *license-id*, the current license ID – if defined – will be removed from the primary or backup license respectively.
- *license-id*: must be used together with the *role* parameter. It contains the ID of the license key that is to be used as primary or backup license.

If running the command on a VE-enabled emulator, the error reported will be `Unsupported command`.

Example:

The following example shows how the current HASP license setting is displayed, how a license ID is configured for the primary license, and how the license ID is removed again.

```
Charon-SSP> hasp
Regular license ID:
Backup license ID :

Charon-SSP> hasp regular 362831868
New regular license ID: 362831868

Charon-SSP> hasp
Regular license ID: 362831868
Backup license ID :

Charon-SSP> hasp regular
New regular license ID:

Charon-SSP> hasp
Regular license ID:
Backup license ID :
```

7.7.2.2.1.4 The **ve** Command

The **ve** command allows the user to display and to change address and passphrase of the primary and backup license servers.

Syntax:

```
ve { role param { value } }
```

Parameters:

- If run without parameters, the current configuration will be shown.
- *role*: defines the role of the license server. Possible values are **primary** or **backup**.
- *param*: must be used together with the *role* parameter and defines which parameter of a license server should be changed. Possible values are
 - **addr** - IP address or hostname and, if applicable, non-default TCP port of the license server
Format of address value: `<ip-addr>[:port]`
 - **pass** - passphrase of the product section that is to be used.
- *value*: defines the value of *param*. Depending on the value of *param*, it either specifies the IP address (and TCP port) of the license server or the passphrase of the product section to use. If no value is specified, the current setting is deleted.

If running the command on a HASP-enabled emulator, the error reported will be `Unsupported command`.

Example:

The following example shows how to display the current license configuration and how to set address and passphrase for the backup license server.

```
Charon-SSP> ve
Primary license server: localhost, passphrase: B4RL-1V6J-HI3G-TSC6
Backup license server : , passphrase:

Charon-SSP> ve backup pass B4RL-aaaa-bbbb-cccc
New passphrase of backup license server: B4RL-aaaa-bbbb-cccc

Charon-SSP> ve
Primary license server: localhost, passphrase: B4RL-1V6J-HI3G-TSC6
Backup license server : , passphrase: B4RL-aaaa-bbbb-cccc

Charon-SSP> ve backup addr 10.10.10.1
New address of backup license server: 10.10.10.1

Charon-SSP> ve
Primary license server: localhost, passphrase: B4RL-1V6J-HI3G-TSC6
Backup license server : 10.10.10.1, passphrase: B4RL-aaaa-bbbb-cccc
```

7.7.2.2.2 Debugging Commands not Relevant for End-Users

The following commands must only be used if advised to do so by Stromasys. They are only listed here for completeness.

| Command | Description |
|--|---|
| dtlb [reset print | Clear or dump entries in D-TLB (SSP/4U only). |
| insn <i>cpu-id num</i> | Enable instruction recording (SSP/4U only). Arguments: <ul style="list-style-type: none"> • <i>cpu-id</i>: ID of the CPU for which the recording should be performed • <i>num</i>: maximum number of instructions to be recorded |
| trace [dump on off] | Dump the recorded instructions (SSP/4U only). The number of instructions is defined by the insn command. |
| modify <i>addr value</i> | Set the 8-bit location at <i>address</i> to <i>value</i> . Arguments: <ul style="list-style-type: none"> • <i>addr</i>: address of memory location to update (in hexadecimal notation) • <i>value</i>: 8-bit value to be stored at address. This value is specified in hexadecimal notation. |
| memory <i>addr { units }</i> | Dump memory content starting at the specified address. Arguments: <ul style="list-style-type: none"> • <i>addr</i>: hexadecimal address of starting memory location to display • <i>units</i>: optional; hexadecimal length of the memory region <ul style="list-style-type: none"> ○ valid range: 0x10 - 0x1000000 ○ default: 0x100 |
| cpu <i>id</i> | Dump the register data of a virtual CPU. Argument: ID of the virtual CPU. |
| dit { [show reset on off] } | Enable/disable DIT, or display/reset DIT counters. If run without arguments, the default show is used. |

7.8 Using the Charon-SSP Agent

The Charon-SSP Agent is a Linux service running on the same system as the Charon-SSP emulator software and the emulated SPARC systems that are to be managed by the Charon-SSP Manager. This service provides the interface between the Charon-SSP Manager and the emulator software. It also enables automatic detection of Charon-SSP host systems by the Charon-SSP Director and starts emulated instances at host system boot if they were configured with this option via the Charon Manager. A prerequisite for the Charon-SSP Agent is that the Charon-SSP emulator software is installed on the same system. In addition, the Charon-SSP agent package provides several command-line utilities, for example, the HASP license management utilities.

By default, the directory `/opt/charon-agent/ssp-agent/ssp` and its sub-directories contain the configuration files, the log files, and other configuration information for virtual SPARC systems created using the Charon-SSP Manager. A backup of `/opt/Charon-agent/ssp-agent/ssp` saves this information for all virtual machines created by the Charon-SSP Manager.

A user can also **create virtual machine configurations manually in other locations** on the filesystem. In addition, **virtual disk containers** are usually stored in a different location. Such configuration data and the virtual disk containers of all Charon-SSP instances must be backed up separately.

Charon Agent Log: The Charon Agent writes its log messages to the file `/opt/charon-agent/ssp-agent/agent-log/agent.log`. The log file is rotated every time the Agent is started. In addition, the Agent also logs messages to `journalctl`.

On a system where the Charon Manager is not used, the agent does not need to run. However, it must always be installed to make the required command-line utilities available. Starting with version 4.0.x, if you use the Charon Manager once to configure emulator instances to start automatically with the host system boot, the Charon Agent must run to perform this task even if the Charon Manager is no longer used. As an alternative, users can create their own start/stop script or systemd service to automatically start an emulator instance during host system boot. Charon-SSP no longer creates the start/stop scripts in `/etc/init.d` that were created by older versions.

7.8.1 Starting the Charon-SSP Agent Service

The Charon-SSP Agent service can be (re-)started by executing one of the following commands from a privileged account:

```
# systemctl start ssp-agentd
or
# systemctl restart ssp-agentd
```

If the Charon-SSP agent was not stopped cleanly using the stop command, it may not start using the **start** command. In such cases, the **restart** command can be used.

Important information: a problem exists in versions 4.0.x before version 4.0.4 that will cause the Agent to stop all active emulator instances that were started by the Charon Manager or configured via the Charon Manager for automatic startup at host system boot when the Agent itself is stopped. Please review the release notes for more details and the description of a workaround.

7.8.2 Stopping the Charon-SSP Agent Service

The Charon-SSP Agent service can be stopped by executing one of the following commands from a privileged account:

```
# systemctl stop ssp-agentd
```

7.8.3 TCP/IP Ports Used by the Charon-SSP Agent

The Charon-SSP Agent uses port 9091 (TCP and UDP) to communicate with the Charon-SSP Manager and Director. For the communication with the Charon-SSP Director, port 9101 (UDP) is also required.

7.9 User Access to the Virtual SPARC System

Depending on how the virtual machine console device has been configured, it is possible to access the console in several different ways. Access via the graphical user interface is also possible.

The configuration of the following options is shown in the remainder of this section:

Console access

- Physical serial console access
- TCP/IP based serial console access from Charon-SSP Manager
- TCP/IP based serial console access without Charon-SSP Manager
- Console access via the emulated graphics device

Graphical user interface via an X-server

- Enabling XDMCP on Solaris 2.5 and higher
- Enabling XDMCP on Solaris 10
- Starting the X-server
- Stopping the X-server
- X-server configuration parameters

Should you run Charon Manager and Charon Director on Microsoft Windows, please also refer to the appendix [Charon-SSP GUI for Microsoft Windows](#).

7.9.1 Console Access

7.9.1.1 Physical Serial Console Access

For physical console access, the virtual machine must be configured to attach the virtual serial port to a physical serial port on the host system. This configuration task is performed using the Charon-SSP Manager as shown in [Physical Console Device](#) and [Physical Vconsole Configuration](#).

Additional serial port configuration options, such as speed, parity, and stop-bits must be configured using the **ttya-mode** variable in the OpenBoot guest environment. The following example shows the default configuration values for **ttya-mode**.

| Displaying ttya-mode console variable | | |
|---------------------------------------|--------------|---------------|
| ok printenv | | |
| Parameter Name | Value | Default Value |
| auto-boot? | false | true |
| boot-file | -v | |
| boot-device | disk:a disk1 | disk net |
| ttya-mode | 9600,8,n,1,- | 9600,8,n,1,- |
| ttyb-mode | 9600,8,n,1,- | 9600,8,n,1,- |

For additional information regarding the configuration of the **ttya-mode** variable, see the *OpenBoot Command Reference* in the appendix.

7.9.1.2 TCP/IP-based Serial Console Access via Charon-SSP Manager

From the Charon-SSP Manager you can access the **serial console** via the **Console** tab. The example below shows the console of a SUN-4U system.

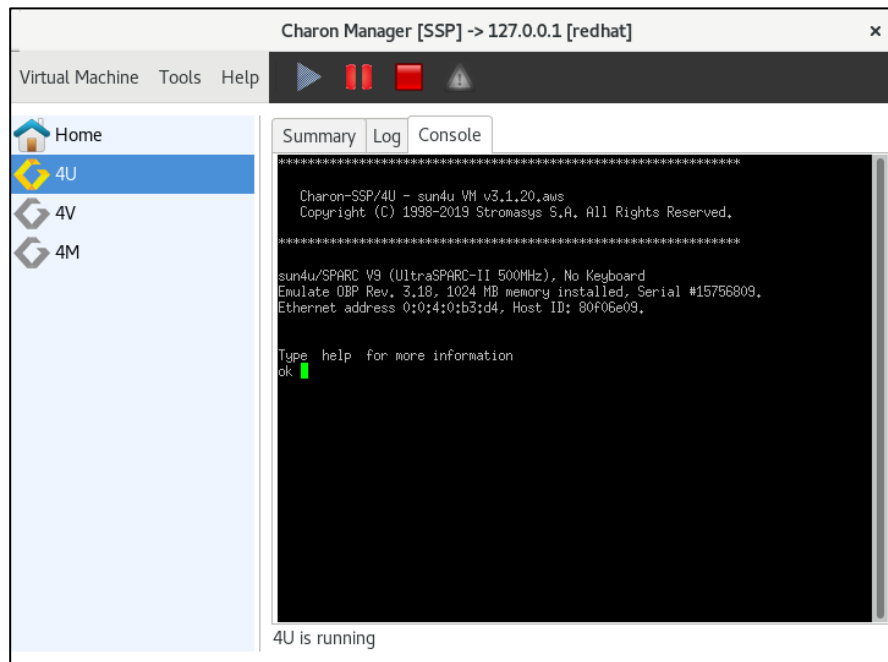


Figure 92: Charon-SSP Manager console tab

To configure the serial console access for the Charon-SSP Manager, use a **TTYA** or **Vconsole** (4V systems) configuration like the one shown below.

- The port **type** must be **TCP Raw** or **Telnet**.
- The **console** parameter must be set to **Built-in**.
- The TCP **port** specified must not be used for another application or another emulated Charon-SSP serial port on the same host system.
- Access can be set to **Local Only** to restrict access to the console to the local system, or to **Unlimited** to allow access across the network.

To access the console of a guest system across the network, make sure the port configured for the console is permitted by any intermediate firewalls (unless the embedded console is used via the integrated SSH tunnel of the Charon-Manager). If the integrated SSH tunnel is not used, please bear in mind that this traffic is not encrypted by default. Make sure to comply with your internal security requirements.

If the port is not reachable, the built-in console window seems to hang with a black screen while it keeps trying to connect to the port.

The example below shows the TTYA configuration of a Charon-SSP/4U system as an illustration. The Vconsole parameters for Charon-SSP/4V are the same as for the 4U TTYA console shown below.

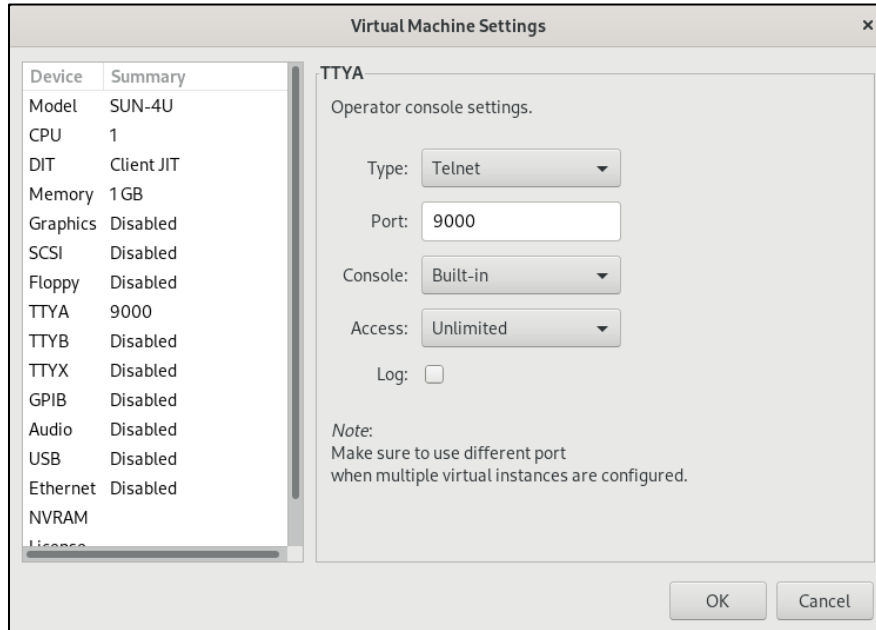


Figure 93: TTYA built-in console configuration (4U example)

Please note:

Only one connection to the console is possible at one time. If the **Charon Manager is connected via the integrated SSH tunnel** and you try to open a second connection to the console via a remote terminal emulation program, Charon Manager will terminate the second connection and re-establish the built-in console connection. If the **Charon Manager is not running or not connected via the integrated SSH tunnel**, a console connection can be established via a remote terminal program and the built-in console tab will be disconnected (if running in Charon Manager, a prompt will be shown to confirm the disconnect).

7.9.1.3 TCP/IP-based Serial Console Access without Charon-SSP Manager

The serial console can also be configured for network access only. If you configure TTYA via the Charon-SSP Manager GUI, the configuration of the serial port itself is identical to the one for the Charon-SSP Manager built-in console tab. However, the console type is **external**. This will disable the console tab of Charon-SSP Manager.

Once configured, any terminal emulator with telnet capability can be used to connect to the port on the host system and access the guest system's serial console. The console emulates a VT100 terminal. By default, Charon-SSP will try to start a PuTTY window on the Charon host system – independent of where the Charon Manager runs – if it finds a valid DISPLAY variable. For more information, please refer to the section [External Serial Console Considerations](#). If a raw connection is being used (**TCP Raw** instead of **Telnet**), there is local echo when connecting with a Linux telnet client (every input is echoed twice). To avoid this, you can enable the telnet-option **mode character** after using the telnet escape key to get to the telnet command prompt. You can also place the command in the **.telnetrc** file.

Please note:

- To access the console of a guest system across the network, make sure the port configured for the console is permitted by any intermediate firewalls. Please note that this traffic is not encrypted by default. Make sure to comply with your internal security requirements.
- Only one connection to the console is possible. If you connect to an external serial console over TCP/IP and open a second console connection to the system at the same time, the currently active console connection will be terminated.

7.9.1.4 External Serial Console Considerations

7.9.1.4.1 Serial Console Settings in Charon Manager and Configuration File

In the Charon-SSP Manager, the serial console of an emulated system can be defined as

- **built-in** (integrated console window in Charon Manager), or
- **external** (separate terminal emulation program).

This setting is directly related to the parameter **start_console** in the emulator configuration file:

- **external console:** start_console = on
- **built-in console:** start_console = off

The **start_console** parameter has an additional task: it informs the emulator whether to try to start a PuTTY window (start_console = on) or not (start_console = off). It is normally set automatically and not directly accessible via the Charon Manager.

Please note:

If a PuTTY window for an **external console** is started, the emulator always tries to start it **on the Charon host system on which the emulator itself runs** - if the emulator can determine a valid DISPLAY variable on the system. This can lead to the situation that the Charon Manager on a remote system starts the emulator but the console is displayed on the Charon host system – or is displayed nowhere if displaying it on the Charon host fails.

To avoid this, you can do **one of the following**:

- **General case:** edit the configuration file manually to set **start_console** to **off** and then start the emulator from the command line. This will prevent the emulator from starting PuTTY. Then access the console using a terminal emulation program (see next section).
- **Remote Charon Manager on Linux:** change the configuration to use the built-in console to easily display it on the remote system running the Charon Manager.
- **Remote Charon Manager on Windows:** use the PuTTY icon at the top of the window to open the PuTTY console on the local Windows system. This will terminate any other open console connection to the emulator in question.

7.9.1.4.2 Accessing the External Emulator Console from a Remote System

Please note:

- If access is across an SSH tunnel, any intermediate firewall must permit SSH access (TCP port 22) to the Charon host system.
- If access is without an SSH tunnel, any intermediate firewall must permit access to the Charon Agent (TCP port 9091) and any ports used for the Charon serial console emulation (default: TCP port 9000).

Connecting from a Windows management system:

If your remote system is a Windows system, you can use the Charon Manager to connect to the external console of a running emulator via PuTTY:

- Start the Manager and connect to the Charon host system. Use the integrated SSH tunnel for any connections across untrusted networks.
- If the emulator is not running already, start it.
- Click on the PuTTY symbol at the top of the Charon Manager window to start PuTTY and to access the serial console of the emulated system.

Connecting from a Linux management system:

- If the emulator is not running yet, use the Charon Manager to connect to the Charon host and to start the emulator, or start it on the Charon host from the command line.
- To create a secure communication path for the telnet connection to the console, you can use several methods as shown in the following examples:
 - Use SSH to log in to the Charon host and then telnet to the correct port on localhost (e.g., `telnet localhost 9000`), **or**
 - initiate a SSH connection with port forwarding.

Example:

```
ssh -i ~/.ssh/<my-private-key> -N -L50000:localhost:9000 <user>@<charon-host>
```

This example creates a port forwarding configuration where the local port 50000 is forwarded to the console port 9000 of the emulator running on *charon-host*. The example assumes a password-less login using SSH keys, but this is not mandatory. The **-N** parameter prevents that an interactive session is created. It is also optional. Please refer to the SSH manpage (`man ssh`) on your Linux system for more detail.

Once the connection has been created, you can use **telnet** to connect securely to the emulator console by using the following command (based on the example above):

```
telnet localhost 50000
```

You can also use a graphical tool such as PuTTY to connect via the port forwarding connection.

7.9.1.4.3 Changing the Default PuTTY Configuration for the Serial Console

Users may wish to change the PuTTY configuration either to make it compatible with the Linux system configuration (e.g., font used), or for preferential reasons (e.g., color scheme). This is possible, but handled differently on Linux and Windows systems.

7.9.1.4.3.1 Changing the Default PuTTY Configuration on Windows

On a Windows system, the Charon Manager provides the option to change the default parameters used for the serial console PuTTY terminal.

To adjust the settings, perform the following steps:

1. Log in to your Windows system.
2. Start the Charon Manager.
3. From the **Virtual Machine** menu at the top, select **PuTTY Setting**.
4. In the PuTTY configuration window, select the profile **OPA0** from the list of saved sessions.
5. Load the profile **OPA0**.
6. Make the required changes.
7. Go to the **Session** category and save the modified profile.

Now, when you start a PuTTY console window, your modified parameters should be in effect. The settings may be overwritten by a new installation of the Charon Manager.

7.9.1.4.3.2 Changing the Default PuTTY Configuration on a Charon Linux Host

The operation of PuTTY on a Linux Charon host system is handled by the emulator.

The emulator kit contains the default session file. For example, `/opt/charon-ssp/ssp-4u/ssp.session`. For one of the other emulator architectures replace “sun-4u” with either “sun-4m” or “sun-4v”. This default session file is copied to the `sessions` directory of the `root` user at every emulator start with external console.

To adjust the settings, perform the following steps on the Charon host system:

1. Log in to the Charon host system as the `root` user.
2. Create a backup copy of the default session file `ssp.session` (location see above)
3. Copy the default session file `ssp.session` (location see above) to the PuTTY session folder of the `root` user (`/root/.putty/sessions/`). If you have run the emulator with an external console before, the file has already been copied there by the emulator.
4. If the emulator is running, shut down the guest system and stop the emulator.
5. As the `root` user, start PuTTY (using the `putty` command).
6. Select the session `ssp.session` and **load** it.
7. Make the required changes.
8. Go to the **Session** category and save the modified profile.
9. Exit PuTTY.
10. Copy the modified session file to the Charon emulator directory (this step is required because the user-specific profile is overwritten by the emulator at every start). For example:

```
# cp /root/.putty/sessions/ssp.session /opt/charon-ssp/ssp-4u/
```

Now, when you start a PuTTY console window, your modified parameters should be in effect. The modified files in the user directory are overwritten by the emulator at every start. The files in the Charon emulator directories are overwritten by a new installation or an upgrade.

7.9.1.4.4 Common Problems

If the PuTTY window does not start despite a correct emulator configuration, you can check the following:

- The Charon emulator log shows the `DISPLAY` variable identified by the emulator (e.g., `INFO SocketIO Found valid DISPLAY :0`). The `root` user must be able to display applications there for the PuTTY configuration of Charon to work.
- Check the emulator crashlog file. For example, if the default font does not exist, PuTTY exits with an error similar to the following: PuTTY: **unable to load font "server:fixed"**

7.9.1.5 Console Access via the Emulated Graphics Device

Charon-SSP can use the emulated graphical display device as the console if this is enabled in the graphics configuration. In this version, the graphical display can only be configured in local mode.

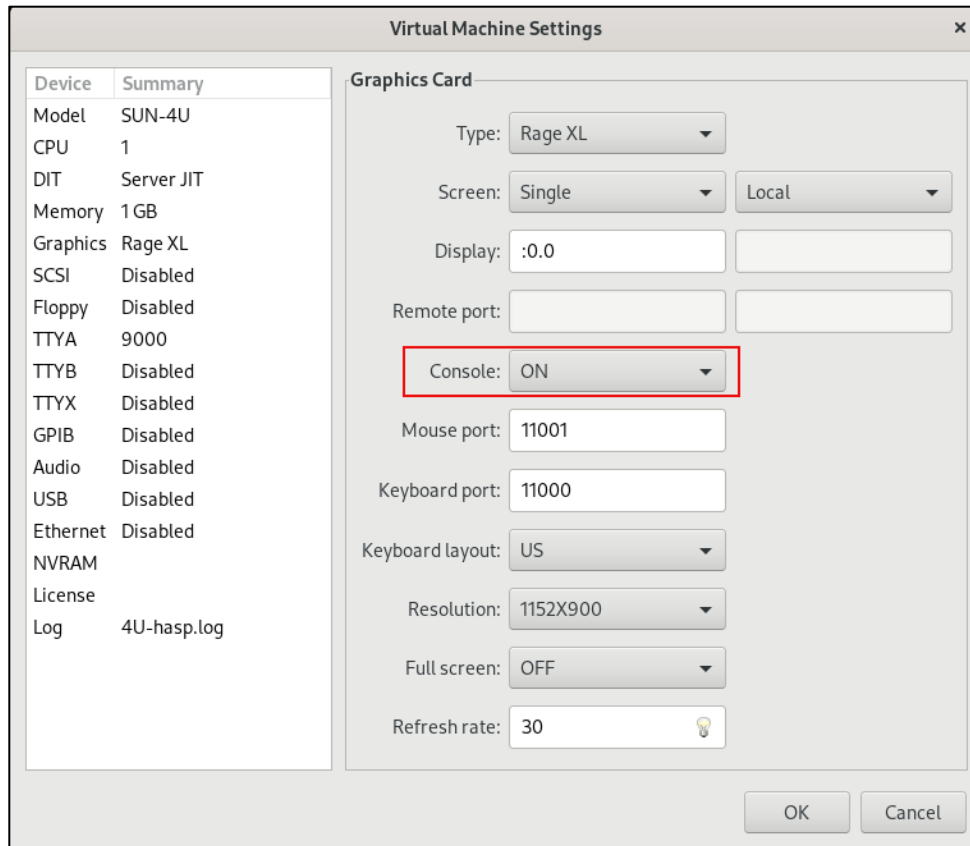


Figure 94: Graphical device configured for local display and console use

After booting, you can log into the graphical console (if dtlogin has been set up on Solaris):

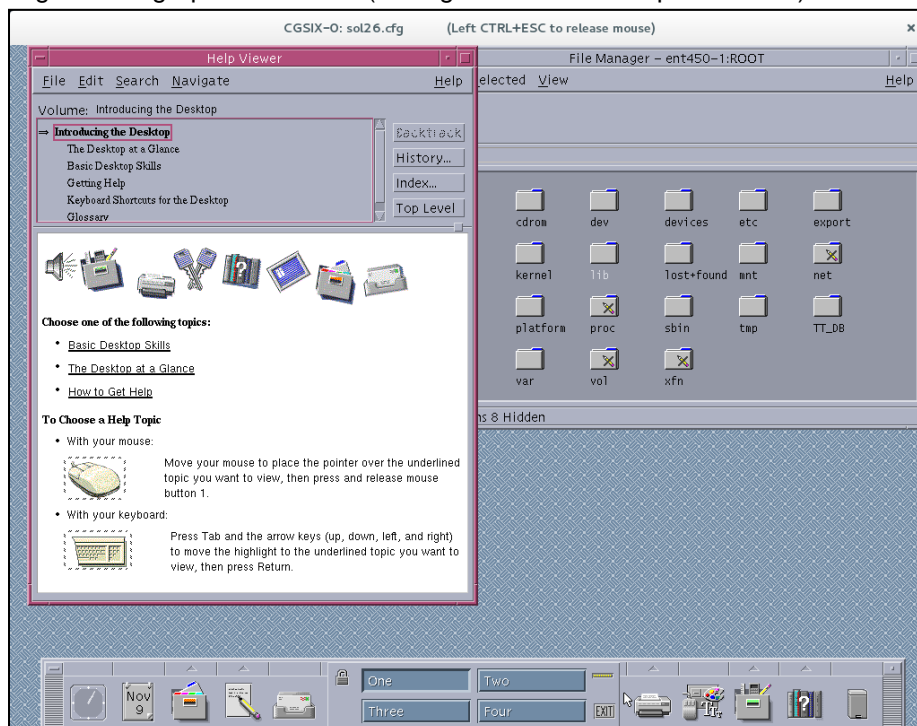


Figure 95: Graphical console after login

7.9.2 Graphical Interface via X11 Server on Linux and Baremetal

The Charon-SSP Manager can set up an X11 login session using Xephyr and the XDMCP protocol. **This feature is not supported across a NAT connection.**

The **graphical performance** depends on many parameters, for example, the performance of host system, emulated system, and network. One important requirement for a **remote display** is that the round-trip time of the network connection between display device and emulated Solaris system should not be more than 20ms

7.9.2.1 General Information

Xephyr is a nested X-server that can run within a normal Linux or Baremetal GUI-based user session. It supports the Solaris GUI (Java Desktop, Openwin, CDE, and Gnome) and can provide graphics 3D acceleration based on the OpenGL 1.4 specification.

Running an X-server to access the graphical Solaris interface, requires a network configuration that allows a TCP/IP connection between the system running the X-server and the Solaris Guest operating system (Stromasys recommends that both systems be in the same subnet). To create a network connection between the local Charon-SSP host system and the guest system, different configurations are possible:

- Using physical network adapters or VMware virtual adapters on same vSwitch, for example:
 - eth0 reserved for the Charon-SSP host system with IP address 192.168.0.100, and
 - eth1 assigned to the Charon-SSP instance as hme0 with IP address 192.168.0.120.
 This requires eth0 and eth1 physically connected to the same external switch/LAN segment.
- Using Charon-SSP virtual network (e.g., eth0 with 1 tap interface), for example:
 - br_eth0 bridged interface assigned to the host system with IP address 192.168.0.100, and
 - tap0 assigned to the Charon-SSP instance as hme0 with IP address 192.168.0.120.

If the X-server runs on a remote system, the remote system must have a working TCP/IP connection to the guest system running in the Charon-SSP instance.

The Xephyr nested X-server listens for connections on port range 6001-6100 depending on the X11 Server configuration in Charon-SSP Manager. The configured ports must be allowed if a firewall (e.g., iptables on Linux) is used. For a quick assessment, in case the X-server does not show the dtlogin screen, the following commands can be used to turn off the firewall **temporarily** (depending on what firewall is being used).

```
# systemctl stop firewalld    or    # service stop iptables
```

Ask your network system administrator to configure proper access to the required port range.

The screenshot below shows an X-session from Charon-SSP Manager to a guest running Oracle Solaris 10

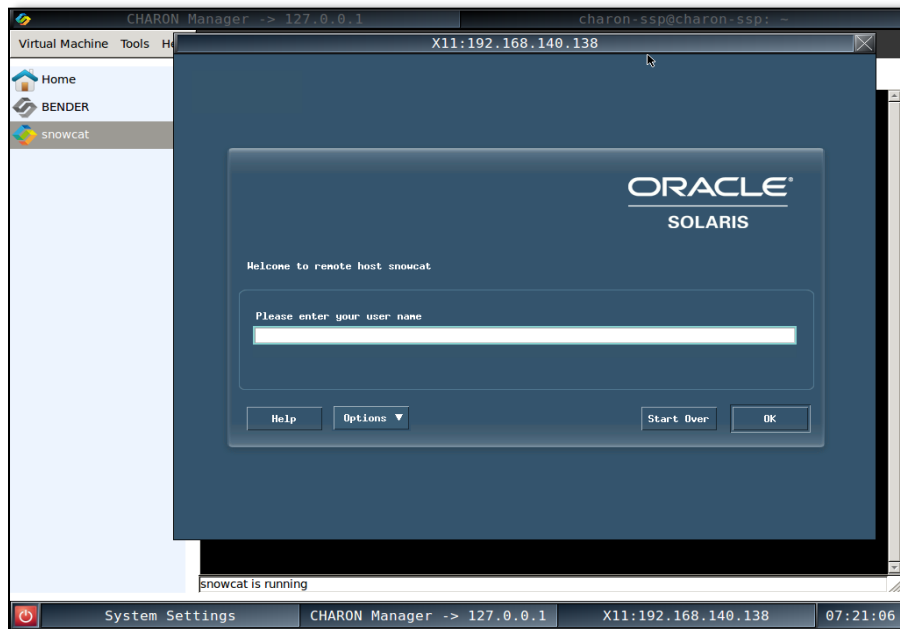


Figure 96: X11 session to a guest running Solaris 10

7.9.2.2 Enabling XDMCP

Before using the X-server, XDMCP must be enabled on the guest system. The actions for enabling XDMCP are different depending on the version of Solaris installed on the guest. Follow the relevant sub-section below to configure XDMCP on your guest.

7.9.2.2.1 Enabling XDMCP on Solaris 2.5 to Solaris 9

Use the following instructions to **enable remote login over XDMCP up to Solaris 9**.

| Step | Description |
|------|--|
| 1 | Edit the file <code>/usr/dt/config/Xconfig</code> # <code>vi /usr/dt/config/Xconfig</code> |
| 2 | Locate the following line and insert a comment character, '#', at the beginning of the line. <code>Dtlogin.requestPort: 0</code> |
| 3 | Save the configuration file and restart the X-server (if there is no <code>dtlogin</code> file in <code>/etc/init.d</code> , you must run <code>/usr/dt/bin/dtconfig -e</code> first): # <code>/etc/init.d/dtlogin restart</code> |

7.9.2.2.2 Enabling XDMCP on Solaris 10

Use the following commands to **enable remote login over XDMCP on Solaris 10**.

| Enable remote XDMCP login on Solaris 10 |
|---|
| # <code>svccfg -s cde-login setprop 'dtlogin/args=""'</code> # <code>svcadm restart cde-login</code> |

7.9.2.2.3 Enabling XDMCP on Solaris 11

Solaris 11 by default uses GDM as the display manager. To enable XDMCP, perform the following steps:

| Step | Description |
|------|--|
| 1 | Edit the file <code>/etc/gdm/custom.conf</code> and enable XDMCP: <pre>[security] DisallowTCP=false [xdmcp] Enable=true</pre> |
| 2 | Restart the GDM service: <pre># svcadm restart svc:/application/graphical-login/gdm:default</pre> |

Please note:

If the graphical environment is missing, install the `solaris-desktop` package (`pkg install solaris-desktop`).

7.9.2.2.4 Enabling XDMCP on Older Solaris Versions

The `dtlogin` command is part of the CDE environment that is not part of Solaris versions before version 2.5. These older systems use OpenWindows instead of CDE and the XDMCP protocol is implemented by `xdm`.

Basic command to start `xdm`:

```
# xdm -config /usr/openwin/lib/xdm/xdm-config
```

To start `xdm` automatically at boot, add a start script to `rc2.d` as shown in the following example (please adapt to your requirements):

```
#!/bin/sh
# xdm
#pid=`/usr/bin/ps -e | /usr/bin/grep xdm | /usr/bin/sed -e 's/^ *//' -e 's/ .*//'|`
pid=`/usr/bin/ps -e | awk '$4 == "xdm" { print $1}'`
xdm=/usr/openwin/bin/xdm

case $1 in
'start')
    if [ "${pid}" = "" ]
    then
        if [ -x $xdm ]
        then
            $xdm -config /usr/openwin/lib/xdm/xdm-config
        fi
    fi
;;
'stop')
    if [ "${pid}" != "" ]
    then
        /usr/bin/kill ${pid}
    fi
;;
*)
    echo "usage: /etc/rc2.d/S90xdm {start|stop}"
;;
esac
```

Please refer to your Solaris documentation for more information.

7.9.2.3 Additional Notes for Solaris 11

There are changes in the latest Solaris 11 versions, in particular Solaris 11.4, that prevent the X11 server configuration via the Charon Manager from working as expected. In particular, the documented version for providing remote graphical desktops is now via VNC. This configuration, and alternatively running graphical applications via X11 to a remote Xserver require additional steps. This section provides the basic steps for such configurations. However, please refer to the Solaris documentation for details.

7.9.2.3.1 Enabling the VNC server on Solaris 11.4

After enabling XDMCP as described above, enter the following commands to enable the VNC server:

```
# inetadm -e xvnc-inetd
# svcadm restart svc:/application/graphical-login/gdm:default
```

Verify that no errors exist (the output of the following command should be empty):

```
# svcs -vx
```

If root login should be allowed, the line **CONSOLE=/dev/console** in */etc/default/login* must be commented out.

The **xvnc** service can be configured with additional parameters (e.g., a non-default geometry). Please refer to the Solaris documentation for instructions.

To connect to the Solaris 11.4 system from a remote system, start the VNC viewer and point it to the IP address of the Solaris system. If configured correctly, you will be presented with a GNOME desktop login prompt.

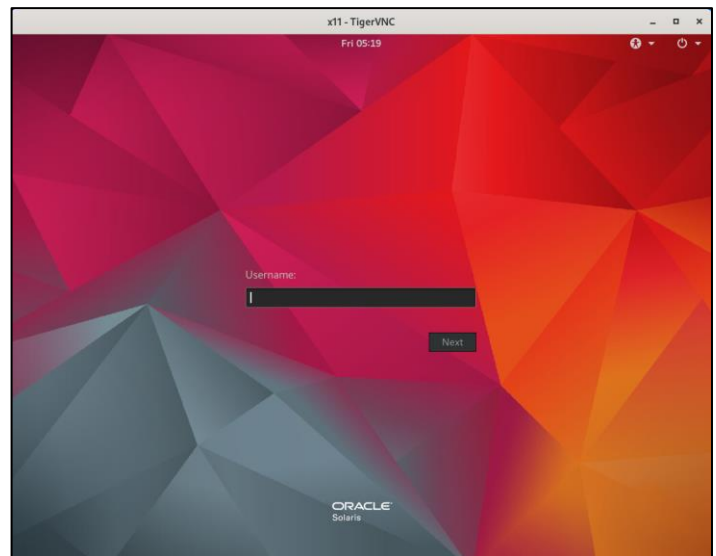


Figure 97: Solaris 11 - GNOME remote desktop

7.9.2.3.2 Enabling Connections to Remote Xservers

Alternatively, remote Xservers can be used to display graphical applications. To achieve this, a X display manager must be installed on Solaris.

You can install the **xdm** display manager using the following command (repository access assumed):

```
pkg install xdm
```

To allow remote connections, the following two files must be adapted:

/etc/X11/xdm/xdm-config:

Comment out the line **DisplayManager.requestPort: 0**

/etc/X11/xdm/Xaccess:

Allow login for specific hosts or all hosts.

After installation and configuration, the **xdm** display manager must be started. You could use the xdm sample above and adapt it to your system environment to create a startup script.

After installing, configuring, and starting the xdm display manager, you can follow the instructions for the Charon Manager X11 server configuration. However, please note that xdm provides quite basic services and that some of the commands and paths mentioned in the next sections could be different or could not exist.

To connect to the Solaris 11.4 system from a remote system, start the Xserver (for example, via Charon Manger) and send an XDMCP query to your Solaris 11.4 guest system. You should be presented with a login prompt.

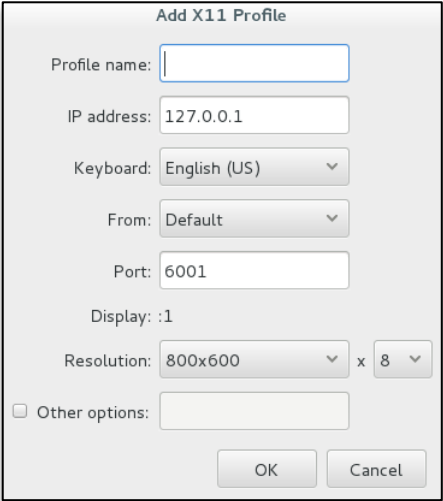


Figure 98: Solaris 11 – Xserver showing remote desktop

7.9.2.4 Configuring and Starting the X11 Server in Charon-SSP Manager

Once XDMCP has been enabled on the guest, use the following basic instructions to start the X-server display. The parameters are described in detail in the next section. You can add multiple profiles with different sets of parameters to the configuration of the Charon-SSP Manager.

Basic steps for configuring and starting the X11 server:

| Step | Description |
|------|--|
| 1 | <p>Open the X11 server Configuration window from Charon-SSP Manager (menu path Tools > X11 Server).</p> <p>Here you can start/stop already configured X11 servers and add, modify, or delete them.</p> <p>To add a new server, click on Add. This opens the Add X11 Profile window.</p> |
| |  |
| 2 | <p>Configure the X11 server by completing the fields:</p> <ul style="list-style-type: none"> • Enter a profile name • Enter the address or name of the guest in the field IP address. • Choose the keyboard layout preferred for this X-session. • Select the host IP address from which the X-server connects to guest Solaris. • Select the port to be used for the communication. • Select the X-session screen resolutions or <i>Full Screen</i> from the Resolution drop down box. • Add additional X-server parameters by checking Other options and entering the desired parameters, e.g., a non-default font-path. For possible parameters see the man-pages for Xephyr and Xserver and section X11 Server Configuration Parameters. • Click OK to save the configuration. |
| 3 | <p>Click on Start to start the selected X-server.</p> |

After **selecting an existing X-server definition**, it can be modified using the **Edit** button and deleted using the **Delete** button.

Should there be problems with the connection, check the following:

- Firewall settings
- **/var/dt/Xerrors** on the guest system
- Any restrictions for remote clients in **/usr/dt/config/Xaccess** and **/etc/dt/config/Xaccess** on the guest system
- Correct font-server configuration on the guest system

7.9.2.5 X11 Server Configuration Parameters

The parameters of the X11 server configuration are explained in the following table:

| Parameter | Description |
|----------------------|--|
| Profile name | Name to identify a specific set of configuration parameters in the list of saved configurations. |
| IP address | IP address of the guest Solaris system. |
| Keyboard | Select the required keyboard from the drop-down list. You can select from the layouts provided by the Charon-SSP host operating system. |
| From | If the system running the X-server has only one IP address, this parameter can be left at default . If there is more than one IP address configured on the X-Server host, select the address that is on the same subnet as the Solaris guest or at least reachable from Solaris. This parameter prevents older Solaris versions from choosing a random (potentially unreachable) address from multiple IP addresses available on the host running the X-Server. |
| Port | Values 6001 - 6100. The port number determines on which display the X-server is started. For example, port 6001 results in the X-server running on display ":1". |
| Display | Read-only field. Shows the display number based on the port number selected. |
| Resolution | This parameter can be adapted to specific requirements of applications with respect to the X-server capabilities ("VISUALS"). One example would be the 256-bit indexed color visual, which requires a display depth of 8 bits. It also allows users to set the X display to full screen mode. |
| Other options | When this option is enabled, X-server specific configuration options can be added. Below, some use cases are described. For a description of all available parameters, see the man-pages for Xephyr and Xserver. |

7.9.2.5.1 Use Cases for the X-Server Additional Options

The following examples show three use cases for the **Other options** parameter:

- Font-server over TCP/IP
- Dual monitor configuration
- Virtual monitor using 2-3 real monitors

Use case 1: font-server over TCP/IP

Frequently, the host operating system (X-Server) fonts do not match the fonts used by Solaris applications. This problem can be avoided by making Solaris the font-server for the X-server. XDMCP automatically tries to set the font-path to the Solaris system.

However, as shown in the table below, XDMCP on different Solaris versions behaves differently with respect to mapping the font-server if the font-path is not explicitly configured.

| Solaris Version | XDMCP Behavior |
|----------------------------|---|
| Solaris 8 and below | Solaris requests to map the font-path using its own <i>hostname+domainname</i> and port 7100. If the result of this name combination cannot be translated to a valid IP address on the system running the X-server (via <i>/etc/hosts</i> or DNS), the mapping of the font path to the Solaris font-server will fail. If the Solaris system does not have a valid domain name, the transmitted string will be in the form of <i>hostname.:7100</i> . |
| Solaris 9 and above | Solaris requests to map the font path using its own IP address and port 7100. |

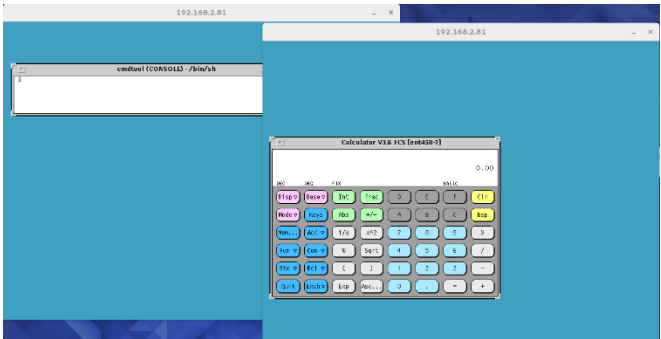
The automatic mapping only works reliably for single monitor configurations. For any **dual monitor** configuration, the font-path to the Solaris system must be explicitly set.

The following table shows the steps necessary to **use a Solaris system as the font-server for the X-server** on Linux or Baremetal:

| Step | Description |
|------|---|
| 1 | <p>On Solaris: make sure that the font-server is enabled either (Solaris up to version 9) in <code>/etc/inetd.conf</code> or via <code>inetadm</code> (Solaris 10). The example below shows <code>inetd.conf</code> where the line starting with <code>fs</code> is uncommented to enable the font-server.</p> <pre># Sun Font Server # fs stream tcp wait nobody /usr/openwin/lib/fs.auto fs</pre> |
| 2 | <p>On Solaris: verify that the file <code>/usr/openwin/lib/X11/fontserver.cfg</code> has the correct values for font-path and maximum number of clients.</p> <pre># font server configuration file # \$XConsortium: config.cpp,v 1.7 91/08/22 11:39:59 rws Exp \$ clone-self = on use-syslog = off catalogue = /usr/openwin/lib/X11/fonts/F3bitmaps/,/usr/openwin/lib/X11/fonts/Type1/,/usr/openwin/lib/X11/fonts/Speedo/,/usr/openwin/lib/X11/fonts/misc/,/usr/openwin/lib/X11/fonts/75dpi/,/usr/openwin/lib/X11/fonts/100dpi/ # in decipoints default-point-size = 120 default-resolutions = 75,75,100,100 client-limit = 10</pre> |
| 3 | <p>On the system running the X-server: use the Charon-SSP Manager to configure an X11 profile with Other options set to</p> <pre>-fp tcp/Solaris_IP_Address:7100</pre> |
| 4 | Start the X-server. |

Use case 2: dual monitor configuration

Xephyr can be configured to use two monitors. The following table shows the **Other options** for this use case:

| Step | Description |
|------|--|
| 1 | <p>On the host system: configure an X11 profile with Other options set to</p> <pre>-screen <res1>x<res2>x<depth> -fp tcp/Solaris_IP_Address:7100</pre> <p>or to</p> <pre>-screen -fullscreen -fp tcp/Solaris_IP_Address:7100</pre> <p>This configuration enables the second screen and define a font server. The resolution normally is the same as the one selected for the first screen in the drop-down menus.</p> |
| 2 | <p>Start the X-server. The attached screenshot shows a very simple example (both windows on the same physical screen) of the outcome:</p>  |

Use case 3: virtual monitor configuration

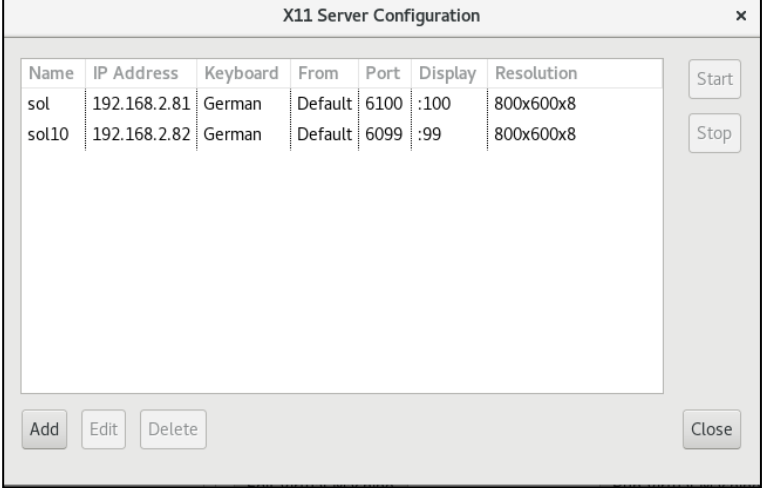
Xephyr can be configured to use two or three monitors as one virtual monitor. Using the **xinerama** qualifier, two monitors are treated as one virtual monitor where windows can be moved between these monitors.

The following table shows the configuration steps:

| Step | Description |
|------|---|
| 1 | On the host system: configure an X11 profile with Other options set to <pre>-screen <res1>x<res2>x<depth> -fp tcp/Solaris_IP_Address:7100 +xinerama</pre> or to <pre>-screen -fullscreen -fp tcp/Solaris_IP_Address:7100 +xinerama</pre> This configuration combines the first and second screen to a virtual monitor and define a font server. |
| 2 | Start the X-server. |

7.9.2.6 Stopping the X11 Server

To stop the X-server, follow the instructions below.

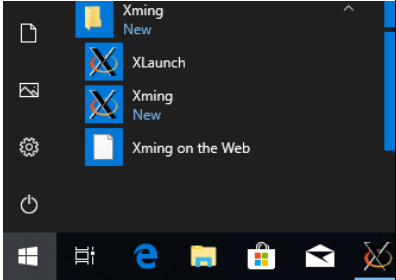
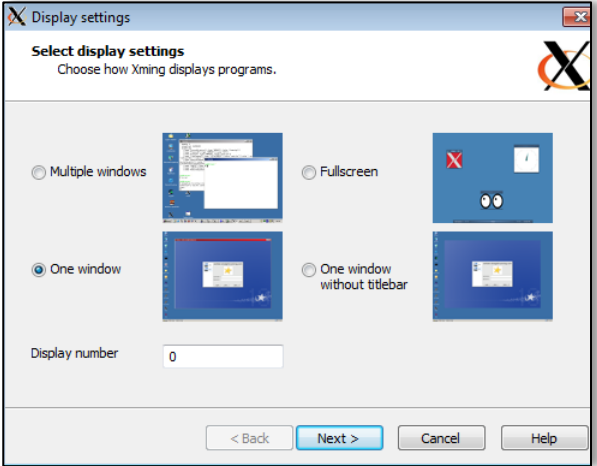
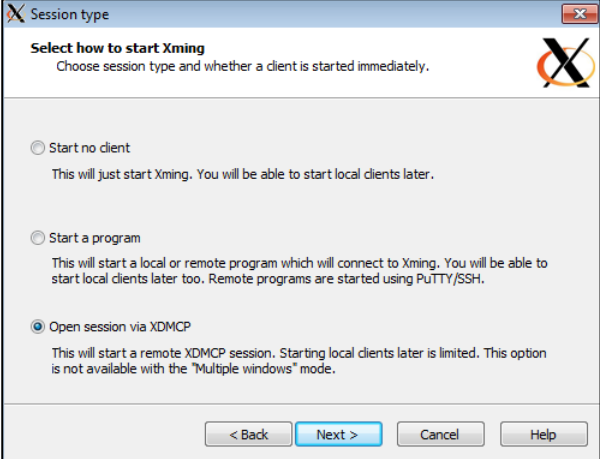
| Step | Description |
|------|---|
| 1 | <p>Open the X11 Server Configuration window from Charon-SSP Manager by following the menu path Tools > X11 Server.</p>  |
| 2 | Select the X-server you want to stop. |
| 3 | Click the Stop button to terminate the X-session. If multiple sessions to the same host are open, these steps must be repeated for each session. To delete a session definition, use the Delete button. |

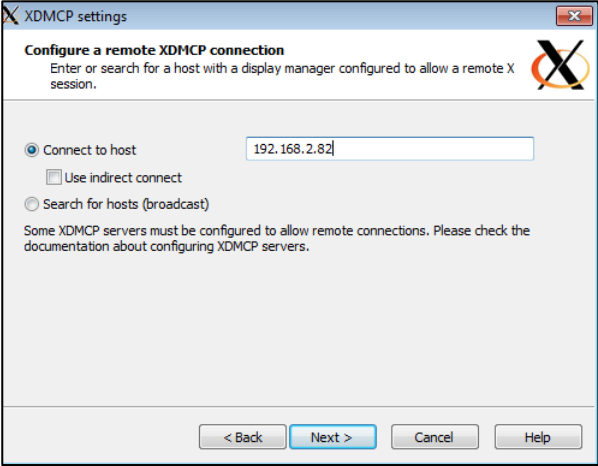
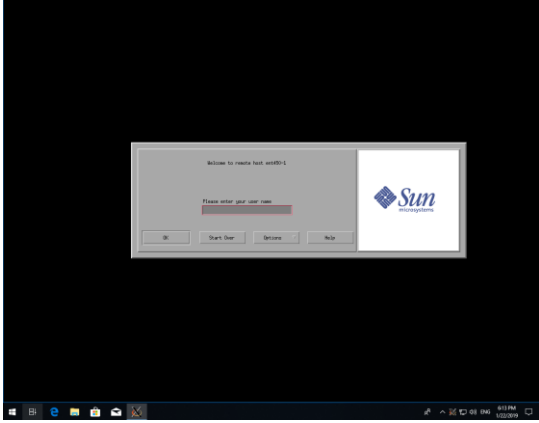
7.9.3 Using the X-Server on Windows

You can access the graphical user interface of the virtual SPARC system from Windows using an X-server. However, the steps are different from the ones used via the Charon-SSP Manager on Linux and Baremetal. First, you must install an X-server. There are several commercial products. However, there are also free X-server packages, for example the X-server integrated in Cygwin, VcXsrv, or Xming. The following examples use Xming. The installer for Xming and more product information are available on <http://www.straightrunning.com/XmingNotes/>.

The following table shows the steps needed on a Windows 10 system to access the virtual SPARC system via X using Xming as an example. Depending on the environment, a different X-server and different configuration choices may be required.

Using an X11 server on Windows 10:

| Step | Description | |
|------|---|--|
| 1 | Enable XDMCP on Solaris, as described in the section Enabling XDMCP . | |
| 2 | Start XLaunch from the Xming section in the Start menu. Make sure that Xming is not already running before you start. |  |
| 3 | <ul style="list-style-type: none"> Select the option One Window. This is one of the options supporting XDMCP. Press Next to continue. |  |
| 4 | <ul style="list-style-type: none"> Select XDMCP for the session. Press Next to continue. |  |

| | | |
|----------|--|---|
| <p>5</p> | <ul style="list-style-type: none"> • Enter the IP address or hostname of the virtual SPARC system. • Press Next to continue. |  |
| <p>6</p> | <p>Finish the setup by continuing through the remainder of the sections (for this example all the defaults were accepted) and press Finish on the last configuration screen of XLaunch.</p> | |
| <p>7</p> | <p>The Solaris graphical login screen is displayed.</p> |  |

8 Additional Charon-SSP Tools

The **Tools** menu of the Charon-SSP Manager provides additional tools not directly related to the management and configuration of virtual machines. They are described in this section. Please review the installation prerequisites section in this document unless you have a Baremetal installation.

8.1 iSCSI Initiator

Please note: this Charon Manager feature is not available in Charon-SSP AL images.

The iSCSI protocol encapsulates SCSI data into TCP packets. iSCSI allows a host to connect to storage via an Ethernet connection.

Important iSCSI terms:

| Term | Description |
|-----------------|---|
| iSCSI initiator | The initiator is the iSCSI client. The iSCSI client has block level access to the iSCSI devices. |
| iSCSI target | The target is the iSCSI server. The iSCSI server offers its devices to the clients. One device can be accessed by one client. |
| Discovery | Process, by which the initiator finds the targets. |

As the name indicates, the Charon-SSP tool represents an iSCSI initiator, i.e., an iSCSI client. This tool can be used to discover iSCSI targets and add devices. These devices can be used later for configuring SCSI storage for the virtual SPARC system.

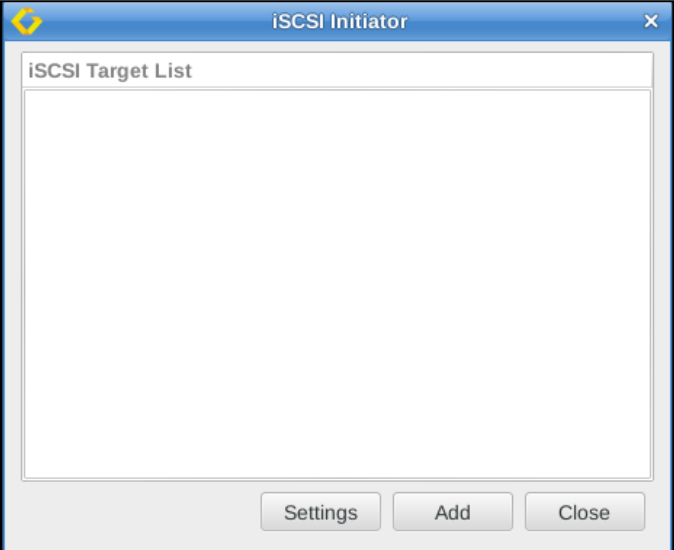
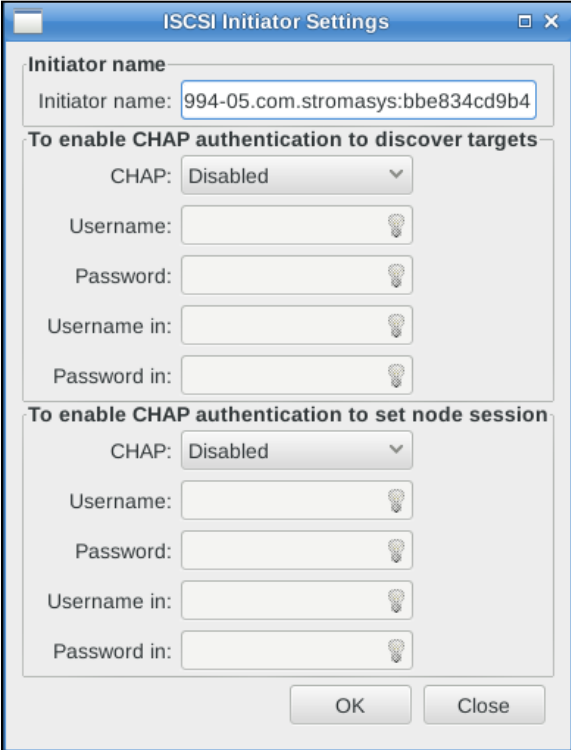
8.1.1 Prerequisites

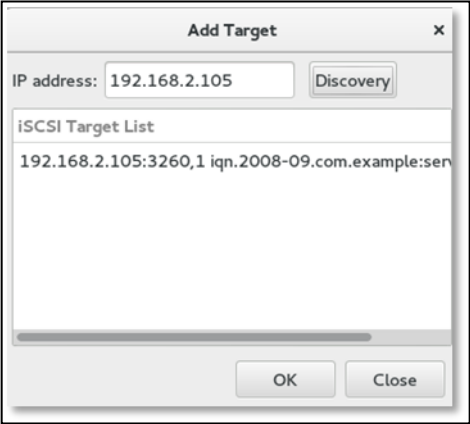
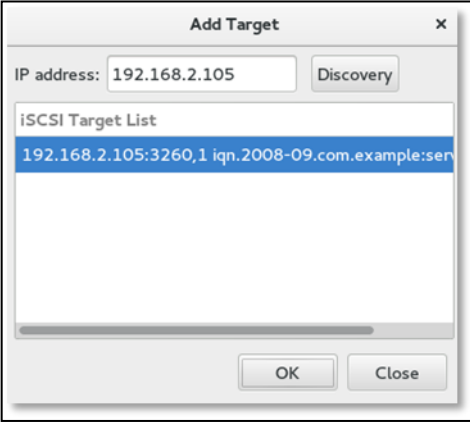
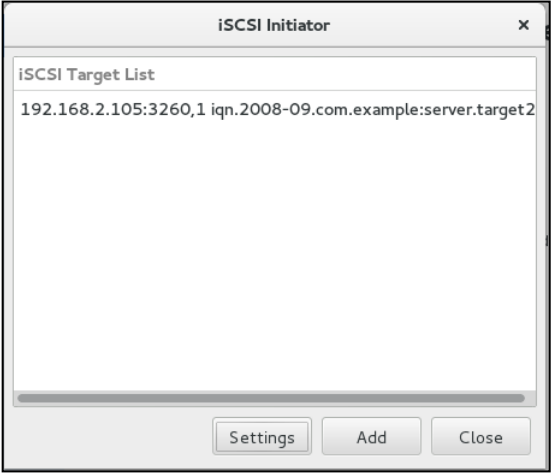
When using iSCSI on a conventional Charon-SSP installation, bear in mind that the Charon-SSP iSCSI initiator uses the standard tools available on Linux systems. These tools are provided by the *iscsi-initiator-utils* package.

The required package must be installed (using **yum** or **dnf** depending on the Linux version) prior to using the iSCSI initiator tool on Charon-SSP. It is preinstalled on the Barebone and Baremetal distributions.

8.1.2 Adding an iSCSI Target

The following table lists the necessary steps to add an iSCSI target for the use of Charon-SSP.

| Step | Description |
|------|--|
| 1 | <p>In the Charon-SSP Manager, select the iSCSI Initiator from the Tools menu. This opens the iSCSI Initiator window displaying the currently available targets. If no target has been added, the screen is empty.</p>  <p>The screenshot shows a window titled "iSCSI Initiator" with a sub-header "iSCSI Target List". The list area is empty. At the bottom, there are three buttons: "Settings", "Add", and "Close".</p> |
| 2 | <p>If authentication to/from the targets is required, click on Settings. This will open the following window where you can enable or disable CHAP authentication for discovery and session setup.</p> <p>For discovery and session setup, after enabling the configuration by selecting Enable in the CHAP drop-down menu, you can set</p> <ul style="list-style-type: none"> • Username and Password if the initiator is authenticated by the target(s). • Username in and Password in if the target is to be authenticated by the initiator. <p>The information entered will be stored in <i>/etc/iscsi/iscsid.conf</i>.</p> <p>To remove the configuration from the system, select Disabled in the CHAP drop-down menu.</p>  <p>The screenshot shows a window titled "iSCSI Initiator Settings". It has an "Initiator name" field with the value "994-05.com.stromasys:bbe834cd9b4". Below this, there are two sections for CHAP authentication. The first section, "To enable CHAP authentication to discover targets", has a "CHAP" dropdown set to "Disabled", and fields for "Username", "Password", "Username in", and "Password in". The second section, "To enable CHAP authentication to set node session", also has a "CHAP" dropdown set to "Disabled" and the same four fields. At the bottom, there are "OK" and "Close" buttons.</p> |

| | | |
|---|---|---|
| 3 | <ul style="list-style-type: none"> Click Add to open the iSCSI discovery window. To discover a new target, enter the IP address of the server and click on Discovery. This adds the target identification to the list. In the background, the Charon-SSP Manager starts the <i>iscsid</i>, if needed. |  |
| 4 | <ul style="list-style-type: none"> To add the discovered target to the system, select it and click on OK. Afterwards close the window by clicking on Close. |  |
| 5 | <p>The iSCSI Initiator window should now display the selected target.</p> <p>When you are done with adding the targets, you can close the window.</p> |  |
| 6 | <p>To verify that an iSCSI session has been established, enter the following command in a Linux terminal window: # iscsiadm -m session</p> <p>To identify the device name associated with the connected target, enter the following command in the terminal window: # iscsiadm -m session -P3</p> <p>At the end of the output, there is an entry like the following:</p> <pre>Attached scsi disk sdc</pre> <p>This example shows that <code>/dev/sdc</code> can be used to configure a disk for Charon-SSP.</p> <p>Charon-SSP Baremetal installations: you can also use the Storage Manager to identify the name of the newly added iSCSI disk(s).</p> | |

Persistent device naming for physical disks:

Linux SCSI device names in the form of `/dev/sdX` are not guaranteed to be persistent across Linux reboots.

Hence, for physical disks, it is strongly recommended to use a **persistent device name** from

- `/dev/disk/by-id`, or
- `/dev/disk/by-uuid`

instead of a non-persistent `/dev/sdX` device name.

Alternatively, the physical disk serial number can be used (device type must be **disk**). It can be determined using the following command:

```
# udevadm info -q property -n /dev/diskname | grep SERIAL
```

Either `ID_SERIAL_SHORT` or `ID SCSI_SERIAL` can be used.

Baremetal installations: you can also find the serial number using the **Storage Manager** in the Tools menu of the Charon-SSP Manager.

8.1.3 Removing an iSCSI Target

Before removing an iSCSI Target, ensure that it is no longer in use and not associated with any disks you need for your virtual SPARC systems! Charon-SSP acts on the delete request immediately and does not ask for confirmation.

To remove an iSCSI target from the system, select **iSCSI Initiator** from the **Tools** menu of the Charon-SSP Manager.

In the iSCSI initiator window, select the target you want to delete and click the **Delete** button.

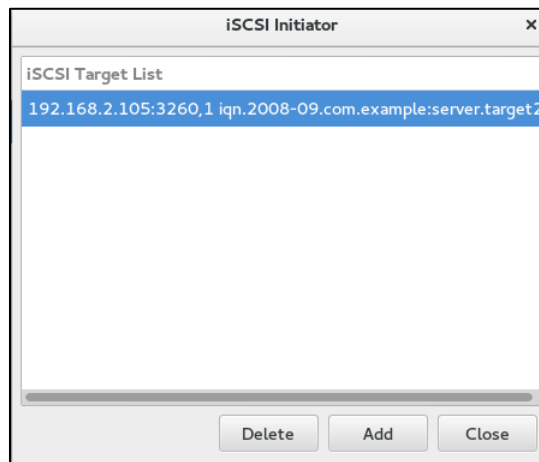


Figure 99: iSCSI target selected for deletion

8.2 NFS Server

Please note: this Charon Manager feature is not available in Charon-SSP AL images.

The **NFS Server** tool enables the Charon host to export file systems or directories to NFS client systems. Please review the installation prerequisites section in this document unless you have a Baremetal installation.

The NFS Server tool of the Charon-SSP Manager is **not** a full-featured NFS administration tool. It allows a basic NFS Server configuration with the purpose of enabling additional data transfer options during the Solaris migration from the physical system into the virtual Charon-SSP environment. It is not meant to be an additional permanent storage option for Charon-SSP.

This section shows the basic use of this tool. The section *Data Transfer to/from the Charon-SSP Host* shows an example of how this tool could be used in a migration scenario.

8.2.1 Prerequisites

When using NFS on a conventional Charon-SSP installation, bear in mind that the Charon-SSP NFS support uses the standard tools available on Linux systems. These tools are provided by the *nfs-utils*, and *rpcbind* packages.

The required packages must be installed (using **yum** or **dnf** depending on the Linux version) prior to using NFS tool on Charon-SSP. They are preinstalled on the Barebone and Baremetal distributions.

8.2.2 Adding an NFS Share

To add an NFS share, select **NFS Server** in the **Tools** menu of the Charon-SSP Manager. In the **NFS Server Configuration** window click on **Add**. This opens the following window:

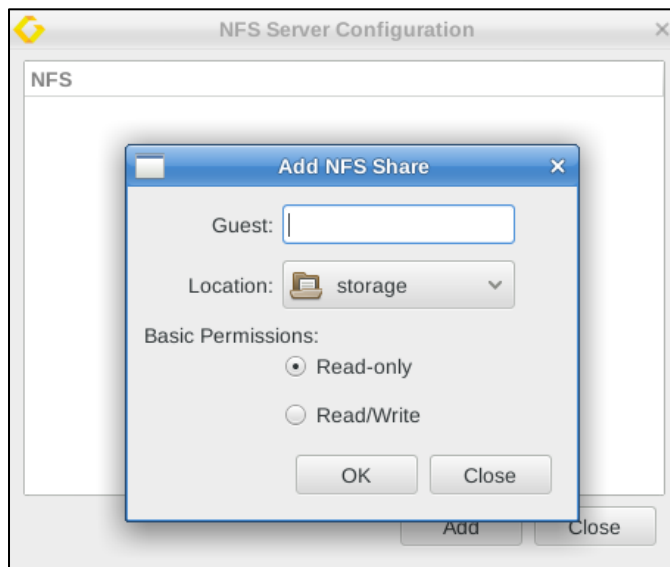


Figure 100: Adding an NFS share

Enter the host address or fully qualified domain name of the host for which access is to be allowed in the field **Guest**. Select the path to export in **Location** and specify whether write access should be allowed. Click on **OK** to accept the configuration. The NFS share is displayed in the NFS Server Configuration window. The definitions are stored in the file **/etc/exports**. Charon-SSP starts the NFS service on the host system automatically.

Any additional NFS configuration options (e.g., UID mappings) or path ownerships would have to be configured through the Linux shell. Please refer to the man-pages for **exports**. If you modify the **/etc/exports** file and open the Charon-SSP Manager NFS server configuration again, you will see these changes, but you will not be able to edit them.

SunOS 4.1.4 NFS client uses NFS V2. The Linux host NFS server uses V3/V4. You can achieve backward compatibility by adding **RPCNFSDARGS = "-V 2"** to **/etc/sysconfig/nfs** on the Linux host and restarting the **nfs-server** and **nfs-mountd** services.

Additional tasks:

Make sure any firewall between the NFS server and the NFS clients allows **rpcbind**, **nfsd**, **mountd**, **statd**, and **lockd**. To enable a proper firewall configuration, it could become necessary to provide static assignments for some ports that are dynamically assigned under normal circumstances. As a quick test to check if the local firewall causes problems with NFS you can disable the firewall temporarily. Make sure to re-enable the firewall after the test.

Example command to disable the firewall on Linux **temporarily**:

```
# systemctl stop firewalld
```

Verifying the export:

Either on the NFS server itself or a system enabled as an NFS client, you can use the following command to verify the list of exported file systems. It displays the list of exported filesystems and for which hosts they are exported.

```
$ showmount -e <ip-address-of-nfs-server>
```

8.2.3 Removing an NFS Share

To remove an NFS share, select **NFS Server** in the **Tools** menu of the Charon-SSP Manager.

In the **NFS Server Configuration** window select the NFS share in question and click on **Delete**. The share is removed from the Charon-SSP Manager and the **/etc/exports** file.

Before removing an NFS export definition, ensure that it is no longer in use. Charon-SSP acts on the delete request immediately and does not ask for confirmation.

8.3 VNC Server

Please note: this Charon Manager feature is only available on Charon-SSP Baremetal systems.

Charon-SSP Manager on **Baremetal** installations comes with a feature to configure a VNC server for the host system. This allows access to the host system desktop across the network. Please review the installation prerequisites section in this document unless you have a Baremetal installation

Firewall prerequisites:

The actual port used by VNC is determined by the VNC server configuration. The available port range is 5901 to 5910. Access to the relevant port(s) of the host system must be allowed.

8.3.1 Enabling and Disabling the VNC Server

To **configure** the VNC server, use the Charon-SSP Manager. Open the menu option **Tools > VNC Server**.

This will open a window displaying already configured VNC servers and offering to create a new server. Click on **Create** to proceed.

The next window will allow you to define the **Display** number between 1 and 10 via the drop-down menu. Click on **OK** to create the server. This will automatically start the VNC server (on port 5901 to 5910 depending on display number).

To **stop** the VNC server and remove the configuration, select the server from the configuration and press the **Kill** button

Should you not have access to the graphical console, you can start a VNC server using the following command:

```
/usr/bin/vncserver_wrapper charon :<display-number>
```

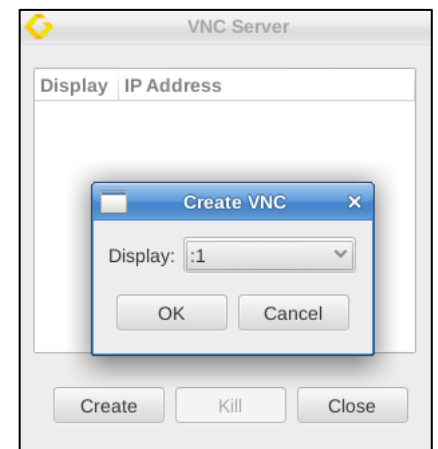


Figure 101: VNC configuration window

8.3.2 Connecting to the Charon-SSP Host via VNC

To connect to the Charon-SSP host from a remote system via VNC, you must start a VNC client with the correct parameters.

Example (using the VNC client *vinagre*):

```
$ vinagre <charon-ssp-host>::5901:1
```

This connects to the Charon-SSP host, port 5901, display “:1” (user *charon*).

The VNC server will request a password. For Baremetal systems, this is the management password set during installation or at first login of the Charon Manager (default: *stromasys*). **Non-Baremetal products only:** The password can be changed on the Charon-SSP host using the command **vncpassword** (assuming the *tigervnc* packages have been installed). On **Baremetal systems**, this password is changed together with all other management passwords using the Charon-SSP Manager (**Preferences** option in the **Virtual Machines** menu).

8.4 Using a Jumpstart Server

Please note:

- This feature is not available if Charon-SSP runs in a cloud environment.
- Supported for Solaris 2.4 to Solaris 10.

A Charon-SSP instance can use a jumpstart server to install a supported Solaris guest system. This section does not describe how to configure a jumpstart server. This is documented in the Solaris system documentation. This section only provides a short overview of important points from the Charon-SSP perspective.

Charon-SSP supports RARP and DHCP (Charon-SSP/4U and Charon-SSP/4V only) configurations.

The Charon instance must be configured with an Ethernet interface on the same LAN segment as the jumpstart server to allow the Ethernet broadcast messages sent by the client to be received by the server. Ensure that the necessary traffic (ICMP, RARP or DHCP, TFTP, Portmapper and NFS) are not blocked by a firewall.

Once the jumpstart server has been configured with the MAC address and the IP address of the client system and the desired Solaris version has been made available for installation, the Charon-SSP instance can be booted and the Solaris system can be installed.

Boot command for a RARP configuration:

```
boot net - install
```

Boot command for a DHCP configuration:

```
boot net:dhcp - install
```

Starting with Charon-SSP version 2.0.5, the `install` switch is supported when rebooting a running Solaris system with the `net:dhcp` parameter (e.g., `reboot -- "net:dhcp - install"`).

9 Data Transfer to/from Charon-SSP Host and Guest

9.1 General Information

When migrating a physical system to an emulated system, all the data of the physical system must be transferred to the emulated system. The sections below provide some hints about how data transfers can be performed. However, the section does not describe a recommended migration path. Migrating a system depends very much on the specific customer environment and the best path must be defined on a case-by-case basis.

For consulting services supporting the migration (subject to a charge), please contact your Stromasys representative or VAR. You can find the contact information on the Stromasys [web page](#).

This section shows a few data transfer options for the Charon-SSP host system and Solaris systems running on real or emulated hardware. This list is by no means comprehensive, but meant as an assistance to get started.

The behavior of the different Charon-SSP product variants differs slightly, but the basic steps in this example are applicable to all of them.

9.1.1 Direct Data Transfer Over the Network

A direct data transfer **between the Solaris system on the real hardware and a (temporary) Solaris system on the emulated hardware** is often the easiest way to migrate data from a physical system to an emulated system—especially in on-premises installations.

Technically, this is also possible if the emulated system is in the cloud. Whether this is a feasible solution in a cloud environment depends on the amount of data, the network throughput, and the stability of the VPN.

9.1.2 Transferring Backup Archives

Instead of a direct data transfer between a physical and an emulated system, you can transfer backup archives to the Linux host and provide them to the emulated system (for example, via NFS, across an internal bridge, or as a vtape device). See also below.

9.1.3 Cloud-Specific Data Transfer Options

Depending on the customer requirements, the configuration of the original system, and the amount of data, different data transfer options may have to be applied. The various cloud providers offer add-on services to facilitate the transfer of large amounts of data from the customer premises to the cloud instances. This section provides a brief list of such services. However, these services are independent of the Stromasys product offering. So always refer to the documentation of your cloud provider for up-to-date information.

- **Microsoft Azure data transfer offering:** for large data transfers, Azure offers special data transfer services. Please refer to the description of the [Azure Data Transfer Solutions](#) for more information.
- **OCI data transfer offering:** for large data transfers, Oracle offers its [Data Transfer Services](#).
- **AWS data transfer offering:** for large data transfers, Amazon offers a special service, [AWS Snowball](#).
- **Google data transfer offering:** for large data transfers, Google offers its [Data Transfer services](#).
- **IBM data transfer offering:** for large data transfers IBM offers its [Mass Data Migration services](#).

9.2 Using VTAPE Container Files

9.2.1 Convert Backup Archives to VTAPE Container Files

The Charon-SSP **mtd** command provides the option to convert Unix files to *.vtape files that can be read by the Solaris guest system.

This provides the option to convert any Solaris backup archive from the original system to a VTAPE container file which can be added to the configuration of the emulated SPARC system. If the provided archive is compatible with the target system, Solaris should be able to restore it.

Steps to convert a Solaris backup archive to a *.vtape file:

1. Copy the backup file to the Charon Linux host system (or make it available via NFS).
2. Perform the conversion:

```
mtd -u2v <unix-file-name> <vtape-file-name>
```

A full list of the **mtd** parameters is available in the [mtd command reference section](#). Emulated tape configuration and handling is described in the [SCSI Storage Configuration](#) section.

9.2.2 Using UFSdump Backup Archives as VTAPE Container Files

This method can be used to migrate UFS filesystems from a physical to an emulated SPARC using **ufsdump** backup archives.

To transfer backup archives to the Charon host system and provide them to the guest Solaris system, perform the following basic steps (details will vary depending on customer requirements and environment):

- Create **ufsdump** archives of the filesystems on the original system that are to be migrated.
Example: # **ufsdump 0f** <archive-filename> <disk-partition>
- Copy the archive files to the Charon host system.
- Add a virtual tape drive to the Charon-SSP virtual machine configuration of the emulated system (with the container file name for example <archive-name>.vtape).
- If running, shut down the Solaris guest system, stop the emulator, and restart emulator and guest system.
- Assign the backup archive as the tape container and use **ufsrestore** on Solaris to restore the files to the correct filesystems. Example:
 - If currently in use, set the tape drive offline in Solaris: # **mt -f** <tape-device> **offline**
 - If necessary, move the current *.vtape file to a backup location.
 - Move the archive file to the location of the tape container file (<archive-name>.vtape in the example).
 - In Solaris:
 - Load the tape: # **mt -f** <tape-device> **status**
 - Rewind the tape: # **mt -f** <tape-device> **rewind**
 - Use **ufsrestore** to restore the backup archive to the **correct file system**.
Example: # **ufsrestore -xf** <tape-device>

9.3 Using NFS for Data Transfer

As mentioned, the NFS Server tool provided by the Charon-SSP Manager is intended to be a support for migration scenarios and not a permanent Charon-SSP storage option. Hence, this example shows the steps to export a filesystem on the Charon-SSP host that can be mounted from Solaris (on physical and emulated systems) to **indirectly** transfer data between the two Solaris systems via the filesystem exported by the Charon-SSP host.

The following image provides an overview of the setup:

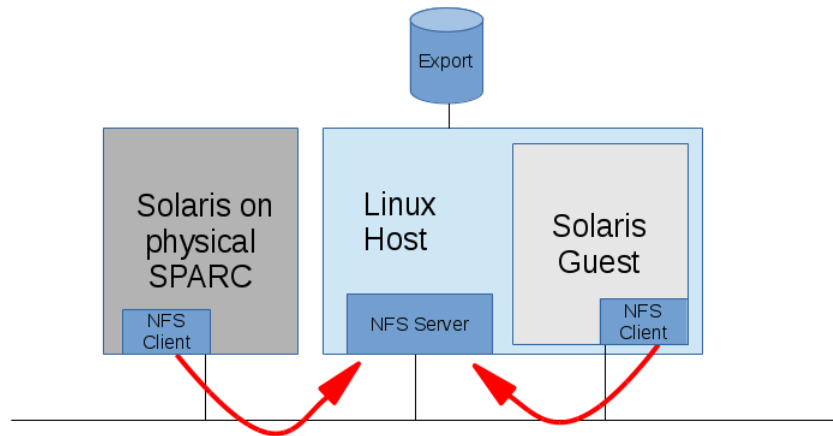


Figure 102: NFS example overview

Prerequisites:

- The Solaris guest system must be installed.
- There must be IP connectivity between the Charon-SSP host system and both the Solaris guest system and the Solaris on the physical system.
- The firewall on the host system must be configured to permit the NFS traffic, or be temporarily disabled.

Please note:

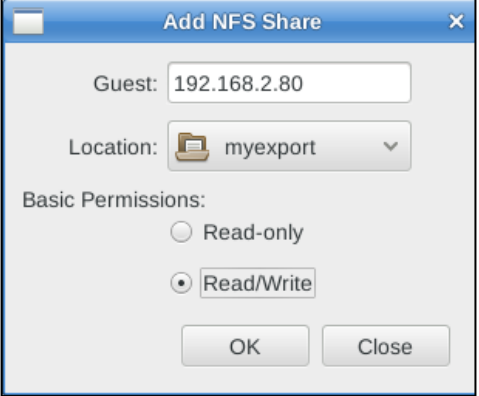
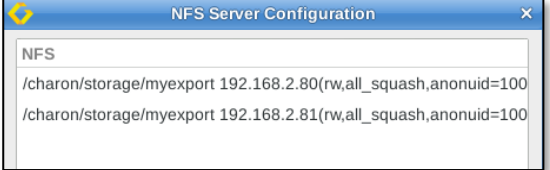
- Often a direct data transfer between the physical and the virtual Solaris system is possible. In many cases, this may be the preferred migration option (depending on network bandwidth and transmission speed).
- Instead of mounting the NFS filesystem on the physical system, you can also use a process, by which you transfer backup archives to the Linux host and provide them to the emulated system via NFS across an internal bridge.

9.3.1 Charon-SSP Host Configured as NFS Server

You can export NFS shares from the Charon-SSP host system so that Solaris systems can mount them. This allows you to share data indirectly between the NFS client systems. The following is a much-simplified example and different environments may need a more sophisticated setup. This section assumes that the NFS server tool is available on your Charon-SSP host systems.

The following example shows how to mount an NFS share exported by the Charon-SSP host system from two different Solaris systems and use it to exchange data.

First, configure filesystems to be exported by the host system. The following table shows the necessary steps to set up the NFS server:

| Step | Description | |
|------|---|--|
| 1 | Open a shell terminal window (on Baremetal, you can also use the File Manager). | |
| 2 | <p>General method:</p> <p>Create the directory that you want to export, e.g., under the current user. For example:</p> <pre>\$ mkdir ./myexport</pre> <p>If not already correct, set the owner to root (or charon on a Barebone system).</p> <p>On a Baremetal installation, using the File Manager is an alternative:</p> <p>Use Tools > Charon Baremetal > File Manager to create the new directory. The owner will be user charon.</p> | |
| 3 | <p>Configure the Charon-SSP NFS Server for the two planned clients as shown in Adding an NFS Share.</p> <p>Export the filesystem with write access as shown in the screenshot.</p> |  |
| 4 | <p>The resulting NFS Server configuration window should show exports for two NFS clients like this screenshot. Charon-SSP sets the owner of the exported directory to user</p> <ul style="list-style-type: none"> root (on non-Baremetal systems) charon (on Baremetal and Barebone systems) |  |

As the **second** step, verify the exported filesystems from Solaris and mount them:

| Step | Description |
|------|---|
| 1 | <p>Start NFS client on both Solaris systems, if necessary.</p> <pre># /etc/init.d/nfs.client start</pre> <p>Solaris 10: This script only starts <i>lockd</i> and <i>statd</i> if NFS filesystems are listed in the file <i>/etc/vfstab</i>. However, the command mount -F nfs also starts the two daemons, if needed.</p> |
| 2 | <p>Make sure the NFS export is recognized by the NFS client. On the NFS clients, enter the command</p> <pre># showmount -e <serveraddress></pre> <p>This should show if and what the server exports. The output could be like the following:</p> <pre># showmount -e 192.168.2.104 export list for 192.168.2.104: /myexport 192.168.2.81,192.168.2.80</pre> |
| 3 | <p>Mount the exported filesystem on the Solaris NFS clients as in the following example:</p> <pre># mount -F nfs 192.168.2.104:/myexport /mntnfs</pre> |
| 4 | <p>Now, when you copy data from one Solaris system to the mounted filesystem it is accessible for the other Solaris system.</p> |

After the successful configuration, both systems can access the filesystem and share data as shown in the following examples.

Sample output on Solaris system 192.168.2.81 (representing physical system). This system copies data to the NFS filesystem:

```
# showmount -e 192.168.2.104
export list for 192.168.2.104:
/myexport 192.168.2.81,192.168.2.80

# mount -F nfs 192.168.2.104:/myexport /mntnfs

# df /mntnfs
/mntnfs          (192.168.2.104:/myexport):41083952 blocks 22552716 files

# tar -cf /mntnfs/soll-etc.tar ./etc

# ls -l /mntnfs
total 8064
-rw-r--r--  1 nobody4  other    2040320 Feb 11 12:45 soll-etc.tar
```

Sample output on Solaris system 192.168.2.80 (representing Solaris guest system). This system can access the data copied to the NFS filesystem:

```
# showmount -e 192.168.2.104
export list for 192.168.2.104:
/myexport 192.168.2.81,192.168.2.80

# mount -F nfs 192.168.2.104:/myexport /mntnfs

# df /mntnfs
/mntnfs          (192.168.2.104:/myexport):41075896 blocks 22552715 files

# ls -l /mntnfs
total 8064
-rw-r--r--+  1 nobody4  other    2040320 Feb 11 13:45 soll-etc.tar
```

In a migration situation, the physical system could use the NFS filesystem as the target for **ufsdump** or **dd** output. The guest systems could access the data to run **ufsrestore** or use the **dd** of a complete disk (Slice s2) as a temporary virtual disk (vdisk).

As mentioned above, **it is often possible to transfer data directly between** the two Solaris systems. This may be the preferred way for on-premises installations in many cases.

When the NFS filesystem is no longer needed, it can be unmounted from both Solaris clients as shown below:

```
# umount <mounted-nfs-filesystem>
```

9.4 Using SCP for Data Transfer

With SCP, single files, or directory structures (recursive copy) can be copied to and from the Charon-SSP host and the Solaris guest system.

Please note: the user **charon** on Baremetal installations only supports SFTP access.

The following table shows some SCP command syntax examples:

| Task | Command |
|--|--|
| Copy single file to another system | \$ scp <local-file> <user>@<remote-host>:<remote-file-path> |
| Copy a directory recursively to another system | \$ scp -r <local-path> <user>@<remote-host>:<remote-path> |
| Copy a single file from another system | \$ scp <user>@<remote-host>:<remote-file-path> <local-file> |
| Copy a directory recursively from another system | \$ scp -r <user>@<remote-host>:<remote-path> <local-path> |

To transfer files between another system and the Charon-SSP Linux host, use the commands listed above with the appropriate username and password. The following example shows how a small directory tree is copied to the **charon** user account from another system:

```
$ scp -r ./Tmp charon@192.168.2.107:
charon@192.168.2.107's password:
file4.txt          100%  0    0.0KB/s  00:00
file3.txt          100%  0    0.0KB/s  00:00
file1.txt          100%  0    0.0KB/s  00:00
file2.txt          100%  0    0.0KB/s  00:00
```

For frequent file transfers between the same user accounts, you can create SSH keys for the users and add the key of each user to the **\$HOME/.ssh/authorized_keys** file of the respective user on the other system. This enables file transfer and SSH login without having to enter a password every time.

SSH/SCP availability on Solaris:

To use SSH/SCP on older versions of Solaris (e.g., Solaris 2.6), this software must be obtained from a provider of public domain Solaris packages, such as **unixpackages.com**. They often require a small fee.

On more modern Solaris versions (e.g., Solaris 10), packages are available on the Solaris installation media that provide this functionality.

9.5 Using SFTP for Data Transfer

SFTP is a file transfer program that can be used for secure file transfers between the Charon-SSP host system and other systems. It uses SSH as its base protocol. When used in interactive mode, you can use the **help** command to learn about the command syntax. SFTP can resume interrupted file transfers.

To connect to a remote system, use the following command if using a SSH key-based login:

```
$ sftp -I <path-to-your-private-key> <remote-user>@<remote-IP-address>
```

Use the following command to connect with username and password:

```
$ sftp <remote-user>@<remote-IP-address>
```

Please note:

- By default, on a Baremetal or Barebone Charon-SSP host, the user **charon** is configured to allow SFTP connections. The same applies to Charon host cloud instances that were created using a prepackaged, cloud-specific Charon-SSP marketplace image.
- The SSH key-based authentication is normally configured at instance launch for cloud-instances. For on-premises systems, it must be configured by the user after installation if required (see also [Baremetal SSH Public Key Import](#) and [Creating and Uploading the Public SSH Key](#)).

The following example shows how a directory structure is recursively copied from the Charon-SSP host (IP address 192.168.2.107 in the example) by another system:

```
$ sftp charon@192.168.2.107
charon@192.168.2.107's password:
Connected to 192.168.2.107.
sftp> ls Tmp
Tmp/A  Tmp/B  Tmp/C
sftp>
sftp> get -r Tmp/
Fetching /home/charon/Tmp/ to Tmp
Retrieving /home/charon/Tmp
Retrieving /home/charon/Tmp/B
Retrieving /home/charon/Tmp/B/b
Retrieving /home/charon/Tmp/B/a
Retrieving /home/charon/Tmp/A
Retrieving /home/charon/Tmp/A/b
Retrieving /home/charon/Tmp/A/a
Retrieving /home/charon/Tmp/C
sftp>
sftp> bye
```

The interactive command to copy files from another system is **get** (-r indicates a recursive copy).

The command to copy files to another system is the command **put**.

For Linux, there are also GUI-based public domain SFTP clients, for example **FileZilla**. These must be installed with the system specific package management system (pre-installed on Baremetal systems).

SFTP availability on Solaris:

Once the Solaris guest is reachable from the host system or from the customer network, SFTP can also be used to transfer data to/from the Solaris guest system.

SFTP is part of the SSH software on Solaris. To use SSH/SFTP on older versions of Solaris (e.g., Solaris 2.6), this software must be obtained from a provider of public domain Solaris packages, such as **unixpackages.com**. They often require a small fee. On more modern Solaris versions (e.g., Solaris 10), packages are available on the Solaris installation media that provide this functionality.

10 SSH VPN – Connecting Charon Host and Guest to Customer Network

If the connection between the Charon-SSP host system, including the configured Charon-SSP guest systems, and the rest of the customer's network runs over a public network, it is necessary to secure the traffic against unauthorized access.

The example in this section describes how to configure a **bridged** SSH-based VPN tunnel between the Charon-SSP host and a remote Linux system across a public network. Topologies that are more complicated will require other, more sophisticated, solutions. The advantage of a bridged connection is that L2 protocols are also supported.

The customer is responsible for ensuring that any VPN solution meets the requirements of his or her company's security guidelines. The example in this chapter is only for illustrative purposes.

The tunnel in this example has two endpoints:

- **The remote Linux system:** in this example, this system could be in the customer on-premises network and use the tunnel configuration to connect across a public network to a Charon-SSP host system in the cloud. If in conformance with the customer security policies, the configuration could be expanded to make this Linux system the router between the customer network and the Charon-SSP host system (optionally including guest systems) in the cloud.
- **The Charon-SSP host system:** in this example, the Charon host system could be in a public cloud and require a connection to other customer devices across the Internet.

10.1 Prerequisites

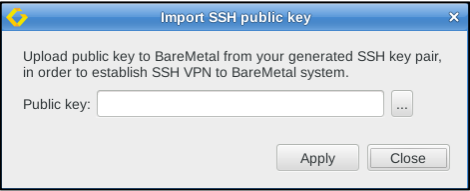

The example shows how to use the Charon Manager on the host system and a set of commands on a remote Linux system to set up an SSH VPN tunnel to the Charon-SSP host system.

Prerequisites:

- The **public SSH key** to be used for the VPN configuration must be copied to the appropriate user of the Charon-SSH host system as shown below. For systems launched from cloud-specific marketplace images, this usually happens during the first launch of the instance.
- If the VPN subnet is to be propagated in the customer network, the remote Linux system must be configured to allow IP forwarding.
- If the Linux host system runs version 7.x of the supported Linux distributions the **bridge-utils** package must be installed on the Charon host system if the Charon Manager configuration options (vs. manual configuration) are to be used.
- The **autossh** package must be installed on the remote Linux system.
- **Conventional RPM installations only** (necessary options are pre-configured on Baremetal systems and pre-packaged cloud systems): adapt the SSH configuration on the Charon-SSP host system to allow root login, enable VPN tunnels, and, if needed, enable login via SSH key.

10.1.1 Creating and Uploading the Public SSH Key

Please note: for Charon host systems based on pre-packaged cloud marketplace images, the creation of an SSH key-pair and the installation of the public key on the Charon-SSP host usually happens during the first launch of the instance. In this case, the steps in this section are not required unless you want to install a separate key for the tunnel configuration.

| Action | Command | |
|--|--|--|
| Create an SSH key-pair on the remote Linux system that will be used as the tunnel end-point. | <p>You need an SSH keypair, for example, to create the SSH based VPN and to use the embedded SSH tunnel of the Charon Manager.</p> <ul style="list-style-type: none"> The public key is installed on the target Charon-SSP host system. The private key is used by your Linux system or the Charon Manager to authenticate itself towards the Charon-SSP host system. <p>If you have not yet done so, use a command similar to the following to create a keypair: # ssh-keygen -t rsa -b 4096 -f ~/.ssh/<keyname> -q</p> <p>You will be prompted for a passphrase to protect the private key. Unless an automated application precludes it, set the passphrase as required. The command creates two files containing the public (*.pub) and the private key. Please check the manual pages of your system for available key types and up-to-date parameter information; very new key types may not yet be supported by Charon-SSP.</p> | |
| Uploading the public key to a Baremetal Charon-SSP system (for the first time or to replace an existing key): | <p>If using the local Charon-SSP Manager on the Baremetal system to import a public SSH key created on a remote system, copy the key file to the charon user first:</p> <pre>\$ sftp charon@<charon-ssp-host-ip> sftp> put <path-to-public-key> <name>.pub sftp> bye</pre> | |
| | <p>Then perform the following steps:</p> <ul style="list-style-type: none"> Start the Charon-SSP Manager (local or remote) for the Baremetal system. Select the menu option Tools > Charon Baremetal > SSH public key. | |
| | <ul style="list-style-type: none"> A window opens that allows you to select the key file. Click on the ... symbol. This opens a file browser. Select the key file and click on Open. After selecting the key file, you will be returned to the initial screen. Click on Apply. |  |
| | <p>A pop-up will open to indicate the successful import of the file.</p> |  |
| Uploading the public key to a non-Baremetal Charon-SSP system : | <p>Connect to the Charon-SSP host (e.g., user root), for example via SFTP, and copy the key file to the system:</p> <pre>\$ sftp root@<charon-ssp-host-ip> sftp> put <path-to-public-key> <keyname>.pub sftp> bye</pre> <p>Log in to the Charon-SSP host system and add the content of <keyname>.pub to /root/.ssh/authorized_keys (using a text editor).</p> <p>The file permissions of the <i>authorized_keys</i> file must be set such that only the owner can access it (e.g., chmod 400 /root/.ssh/authorized_keys).</p> <p>For more information about the possible content of the <i>authorized_keys</i> file, please refer to the SSH man page (man ssh).</p> | |

10.1.2 Adapt SSH Configuration on Charon-SSP Host System

This section is relevant only for systems not based on pre-packaged Charon-SSP cloud or ISO images (i.e., a system installed via RPM packages).

The necessary configuration is performed automatically on a Baremetal systems and cloud-specific images.

| Action | Steps |
|--|---|
| Allow root access and VPN tunnels via SSH. | As the root user edit <code>/etc/ssh/sshd_config</code> . Set the following parameters: <pre>PermitRootLogin yes PermitTunnel yes</pre> If root login with password should be prohibited, you can set <pre>PermitRootLogin without-password</pre> Restart the sshd: <pre># systemctl restart sshd</pre> |

10.1.3 PuTTY Considerations

PuTTY uses its own format for the SSH private key. To use a key created by `ssh-keygen`, it must first be converted. For SSH on Linux to use a key generated by PuTTY it must be converted to the Linux OpenSSH format first.

Linux package containing the conversion utility: **putty-tools**

Conversion using Linux command-line tools:

- Convert a **.pem** (OpenSSH format) file to a **.ppk** (PuTTY format) file:

```
# puttygen mykey.pem -o mykey.ppk -O private
```

- Convert a **.ppk** file to a **.pem** file

```
# puttygen mykey.ppk -O private-openssh -o mykey.pem
```

On Windows, PuTTYgen is normally installed as part of the PuTTY installation. It has a graphical interface that offers, among other functions, the key conversion.

10.2 Setting up the VPN Tunnel

Once the SSH-based VPN tunnel has been set up, you can use it, for example, to

- point the Charon-SSP Manager from the remote Linux system to the Charon-SSP host system,
- have the emulated Solaris graphics device open on the remote Linux system, and
- run the Solaris network connection across an encrypted connection.

The example in this section shows how to set up an SSH tunnel as shown in the overview image below:

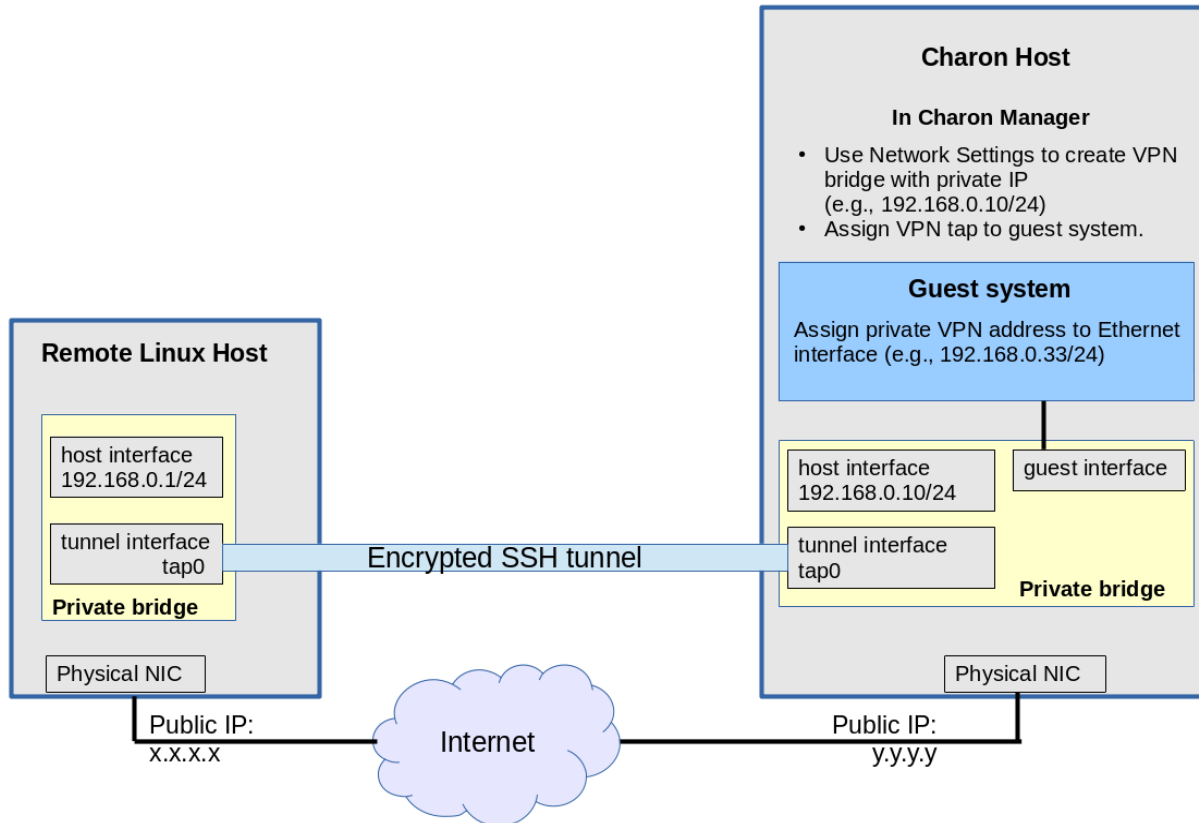


Figure 103: Sample VPN overview

10.2.1 Steps on the Charon-SSP Host System

10.2.1.1 Creating a VPN Bridge

To configure the SSH VPN connection, you must setup a private VPN bridge (called a virtual network in the Charon context) using the Charon Manager. Use the following steps to perform this task (the IP addresses used follow the example shown in the diagram above):

1. Open the Charon-SSP Manager and log in to the Charon-SSP host agent.
2. In the Charon Manager, open the Network Settings window by clicking on **Tools > Network Settings**. This will open the Network Settings window.
3. Click on **Add** and then on **Virtual Network** to open the virtual network configuration window. This will open the **Add Virtual Network** configuration window as shown below.
4. Enter the required information as shown below:

To configure a VPN bridge,

- set **Create for SSH VPN** to **ON**,
- enter the Number of virtual adapters (TAP interfaces) required,
- configure **IP address** of the Charon host on the bridge interface,
- and set the **Netmask**.

The bridge interface represents the Charon host's connection to the bridge. This interface and the interface on the remote Linux system must be in the same IP subnet.

Click on **OK** to save your configuration.

To learn more about the virtual network configuration options, refer to section [Host System Network Configuration](#).

The screenshot shows the 'Add Virtual Network' dialog box with the following configuration:

- Create for SSH VPN: ON
- Binding interface: OFF
- STP for bridge: OFF
- Virtual bridge interface: (empty)
- Virtual bridge name: vpn0
- Number of virtual adapters: 2
- IP settings: Manual
- IP address: 192.168.0.10
- Netmask: 255.255.255.0
- Gateway: (empty)
- DNS server 1: (empty)
- DNS server 2: (empty)

Figure 104: Network settings - SSH VPN

10.2.1.2 Assigning the Guest Ethernet Interface

One of the TAP interfaces created in the step above, must be assigned to the Solaris guest system to add it to the LAN that will be tunneled across SSH to the remote Linux system.

Perform the following steps:

1. Open the Charon-SSP Manager and log in to the Charon-SSP host system.
2. In the Charon Manager, select the guest system and then the Ethernet configuration category on the left. Assign one of the created TAP interfaces (tapX_vpn0) to the guest (see example below).

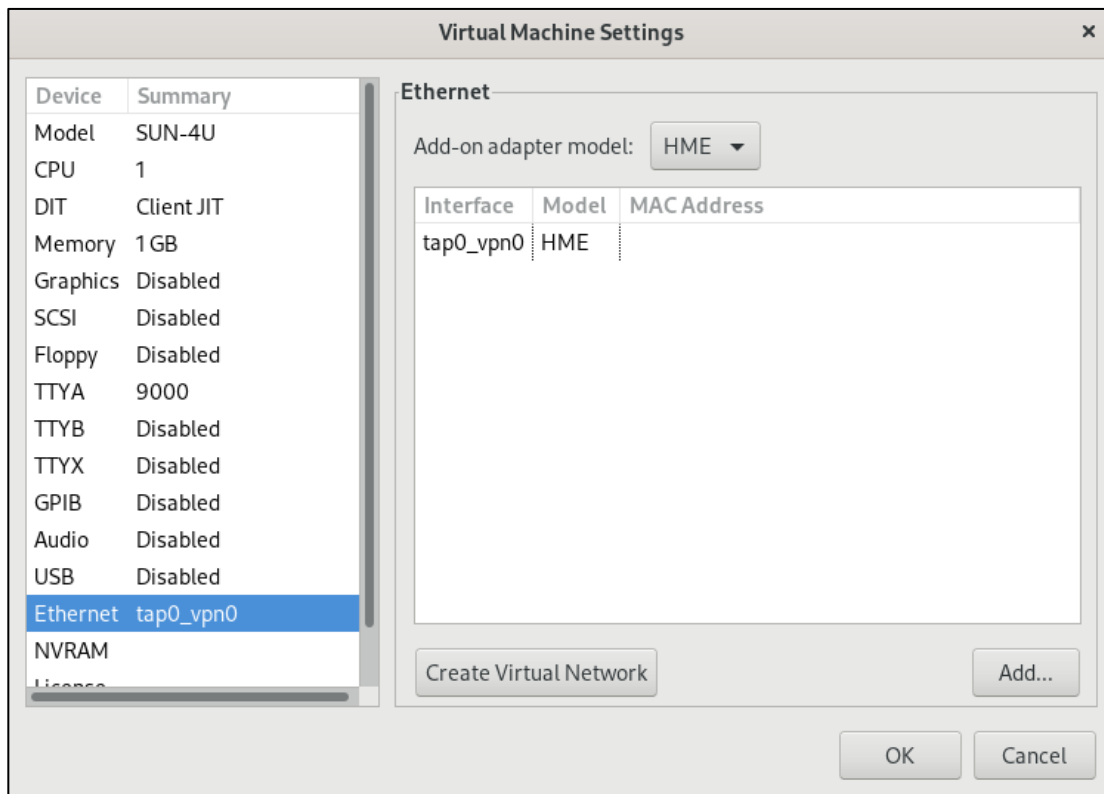


Figure 105: Assign VPN bridge interface to guest

10.2.2 Steps on the Remote Linux System

First configure the Charon-SSP host side. Then, as the user **root**, perform the following steps on the remote Linux system to set up the VPN tunnel according to the overview image above (the **ip** commands are **not persistent**; they should be put into a script once the configuration works):

| Action | Command |
|---|--|
| Create TAP interface. | <code># ip tuntap add dev tap0 mod tap</code> |
| Enable TAP interface. | <code># ip link set tap0 up</code> |
| Create bridge | <code># ip link add name br_vpn0 type bridge</code> |
| Enable bridge interface | <code># ip link set br_vpn0 up</code> |
| Define IP address for bridge | <code># ip addr add 192.168.0.1/24 dev br_vpn0</code> |
| Add TAP interface to bridge | <code># ip link set tap0 master br_vpn0</code> |
| <p>Start the SSH tunnel</p> <p>autossh is a program to start a copy of ssh and monitor it, restarting it as necessary should it die or stop passing traffic.</p> <p>Once started, you can move the program to the background.</p> | <pre># autossh -M 9876 -o ServerAliveInterval=60 -o Tunnel=ethernet \ -w 0:0 -t -i <path-to-private-key> -NCT <user>@<public-charon-host-IP></pre> <p>-M defines the monitoring port autossh uses to monitor the connection -o sets SSH options (bridged tunnel and keepalive) -i denotes the path to the private key matching the public key copied to the host system. -w denotes the number of the local and remote tunnel interfaces for tunnel device forwarding (e.g., the 0 in interface tap0). -N denotes that no remote command should be executed -T disables pseudo-terminal allocation -C requests data compression -f requests that the command go into the background before command execution</p> <p>Value for parameter <i>user</i>: On Baremetal system and cloud-specific images use sshuser, on other systems use the root user or another user for whom you installed the public key.</p> |

Possible additional steps:

If the remote Linux system is to act as a router between the tunnel connection and other systems in the customer network, perform the following steps:

- Enable IP forwarding:
`# /sbin/sysctl -w net.ipv4.ip_forward=1` (to make permanent: add the setting to /etc/sysctl.conf)
- Add static or dynamic routes to distribute the tunnel subnet to other systems in the customer network that need to communicate with the host and the Solaris guest system across the VPN.
- Adapt the firewall on the host system and the remote Linux system as required to allow the VPN traffic to pass.

10.2.3 Steps on the Solaris Guest System

Set the IP address on the Ethernet interface to an address within the VPN subnet. To follow the example above, the address would be 192.168.0.33/24 (# `ifconfig <interface> 192.168.0.33 netmask 255.255.255.0 up`).

To permanently change the IP address on the Solaris system, perform the following steps:

- Solaris 10 and older: change the address of the hostname in `/etc/hosts` that is specified in `/etc/hostname.<interface>`.
- On Solaris 11, use the commands `ipadm create-ip netX` and `ipadm create-addr -T static -a <ip-address>/<netmask> netX/v4`. If an old configuration already exists, you may have to delete it first (e.g., `ipadm delete-ip net0`). Please refer to the Solaris system management documentation for details.

10.2.3.1 Routing to/from Solaris Guest

After following the description above, the Solaris guest system can be reached from the systems that are also connected to the virtual bridge (in the example: remote Linux system and host system). To enable the Solaris guest system to **communicate with other systems** in the customer network (or the Internet) over the VPN connection, perform the following steps:

- Add the VPN address of the remote Linux system as the default gateway for the Solaris guest system.
- Propagate the IP network used for the SSH VPN within the customer network, as required.
- Enable IP forwarding on the remote Linux system and allow forwarded packages through the firewall.

The sample below illustrates the Solaris guest system behavior (after the VPN network has been made known within the customer LAN):

- The interface address shows that the Solaris system is in the 192.168.0.0/24 network using the `ifconfig` command.
- The `netstat -rn` command shows the routing table without a default route.
- The ping to an IP address outside the SSH VPN fails.
- The command `route add default <gwy>` adds the remote Linux host as the default gateway.
- The `netstat -rn` command now shows the default route.
- The ping to an IP address outside the SSH VPN succeeds.

```
bash-3.2# ifconfig hme0
hme0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu 1500 index 2
      inet 192.168.0.33 netmask ffffffff broadcast 192.168.0.255
      ether d4:2:7c:c1:d2:59
bash-3.2#
bash-3.2# netstat -rn

Routing Table: IPv4
  Destination          Gateway                Flags  Ref    Use    Interface
-----
192.168.0.0            192.168.0.33          U        1        1    hme0
224.0.0.0              192.168.0.33          U        1        0    hme0
127.0.0.1              127.0.0.1             UH       4       136   lo0
bash-3.2#
bash-3.2# ping 192.168.2.80
no answer from 192.168.2.80
bash-3.2#
bash-3.2# route add default 192.168.0.1
add net default: gateway 192.168.0.1
bash-3.2#
bash-3.2# netstat -rn

Routing Table: IPv4
  Destination          Gateway                Flags  Ref    Use    Interface
-----
default               192.168.0.1           UG       1        0
192.168.0.0            192.168.0.33          U        1        1    hme0
224.0.0.0              192.168.0.33          U        1        0    hme0
127.0.0.1              127.0.0.1             UH       4       136   lo0
bash-3.2#
bash-3.2#
bash-3.2# ping 192.168.2.80
192.168.2.80 is alive
bash-3.2#
```

Figure 106: Solaris routing via VPN

To make the Solaris routing entry permanent, perform the following steps:

- On Solaris 10: use the `route -p` command (stores routes in `/etc/inet/static_routes`).
- On older Solaris versions: add the address of the default gateway to `/etc/defaultrouter`.

10.3 Stopping the SSH Tunnel

To stop the SSH tunnel, perform the following steps on the remote Linux system:

| Action | Command |
|--|---|
| Terminate the autossh process. | <code># kill -9 <autossh-pid></code> |
| Terminate remaining SSH tunnel connections | <code># kill -9 <tunnel-ssh-pid></code> |
| Delete the bridge | <code># ip link delete br_vpn0</code> |
| Delete the TAP interface | <code># ip link delete tap0</code> |

11 Configuring Charon-SSP Baremetal in Kiosk Mode

With Kiosk mode is enabled, a Charon-SSP Baremetal system appears very much like a traditional SPARC machine, i.e., Solaris is started automatically after power-on and is the only interface visible to the user. To configure Kiosk mode, perform the following steps in the Charon-SSP Manager:

1. Enable graphic emulation to start in full-screen mode. Boot Solaris with the reconfigure option (**boot -r**) to update the device configuration of the Solaris system and configure the graphical interface as required.
2. Configure and test all required functions of the virtual SPARC system.
3. Once the virtual system has been properly configured, enable the option to start the guest system automatically at host system startup ("*Start VM with system*" on the model configuration screen in the Charon-SSP Manager).
4. Shutdown the guest system and stop the emulator.
5. Press **F1** to hide the desktop. The action is persistent across system reboots. Pressing **F1** a second time will show the desktop again. Note that the focus must be on the desktop for this key to work properly. If the focus is on a window (e.g., Charon-Manager), the key will not work.
6. Reboot the system. The graphical SPARC console will appear automatically in full-screen mode.

Important notes:

- It is possible to shut down the host system from Kiosk mode. Before this step, the Solaris guest system must properly shut down to avoid data corruption in the guest system. It is the responsibility of the user to ensure the proper guest system shutdown.
- To achieve the best result, the host system's graphics card should match the Solaris screen resolution and the same resolution should be configured for the Charon-SSP graphics emulation.
- By default, it is possible to switch to a Linux virtual console during Kiosk mode (**CTRL+ALT+Fnum**, where *num* stands for the number of the Linux virtual console). If desired, this key combination can be disabled by the following steps:
 - Open a terminal window from the Baremetal Toolbox.
 - Become the **root** user.
 - Make a backup copy of the file **/etc/X11/xorg.conf.d/xorg.conf**.
 - Open the file **/etc/X11/xorg.conf.d/xorg.conf** with a text editor.
 - Change the option **DontVTSwitch** to **true**.
 - Reboot the Baremetal host system (do not forget to cleanly shut down any running emulator guest systems).

Please note: these steps may have to be repeated after upgrading the Baremetal system with an ISO file.

12 Charon-SSP License Management

This section describes some important aspects of Charon-SSP license management. For a comprehensive description of licensing for Charon products, please refer to the *Charon Licensing Handbook* (for Sentinel HASP licenses) and the *VE License Server User's Guide* (for Charon-SSP VE licenses) in [Licensing Documentation](#).

All Stromasys Charon emulator products are licensed using **one of the following**:

- Sentinel/Gemalto HASP (Hardware Against Software Piracy) USB key
- Software license provided by Sentinel/Gemalto
- A VE (Virtual Environment) license provided by a VE license server (for emulator packages with the string **ve** in the name). See the *VE License Server User's Guide* for more information.
- Automatic licensing for cloud-specific Charon-SSP AL images (either using a public Stromasys-operated license server or a customer-operated AutoVE license server).

A license can be installed locally on the system or—in case of a network license—it can be served to clients on the network by a license server. Unless Automatic Licensing in the cloud is used, the use of the Charon emulator products requires **one of the following**:

- Valid physical license key (HASP dongle)
- Valid Sentinel HASP software license
- Valid VE license

The following Charon-SSP products covered in this guide require a valid license to operate:

- Charon-SSP/4M
- Charon-SSP/4U(+)
- Charon-SSP/4V(+)
- Charon-SSP Baremetal and Barebone distributions

The following products work in conjunction with the licensed products, but do not require a license their own operation:

- Charon-SSP Manager
- Charon-SSP Director
- Charon-SSP Agent

12.1 Licensing Charon-SSP—General Aspects

Apart from Charon-SSP hosts derived from Automatic Licensing cloud marketplace images, creating and updating a license prior and during the operation of Charon-SSP emulators involves the following basic steps:

1. Generate a C2V (customer-to-vendor) file. It contains the data required to generate a license.
2. Submit the C2V information to Stromasys Orders Administration.
3. Receive one or more V2C (vendor-to-customer) license file. It contains the new or updated license data.
4. Apply the V2C files to the local system or the license server.

The following sections describe how to perform these tasks using the different tools available.

The steps **create C2V file** and **apply V2C files** are performed on the Charon-SSP host if a hardware or software license is locally installed or if a license server runs on the same system as the Charon emulator. If a license server in the network is used, these steps are performed on the license server.

For customers licensed with USB HASP keys: these keys contain a built-in battery, which must not be completely discharged. It is recommended that unused keys be connected to spare USB ports from time to time for charging. If a dongle fails, do not discard the dongle. Please contact Stromasys Customer Support immediately.

12.2 Sentinel HASP License Management

Charon-SSP comes with several tools to manage Sentinel HASP licenses. These tools are described below.

12.2.1 Managing Sentinel Licenses with Charon-SSP Manager

The Charon-SSP Manager provides a graphical interface for the management of Sentinel licenses. The following sections describe how to use these tools to perform the following tasks:

- Viewing the License Details
- Gathering Customer-to-Vendor (C2V) Details
- Applying Vendor-to-Customer (V2C) License Update
- Managing License Parameters

12.2.1.1 Viewing the License Details

To view the license details of an attached license, follow the menu path **Tools > HASP Tools > HASP Viewer**. This opens a **HASP Viewer** window, like the following.

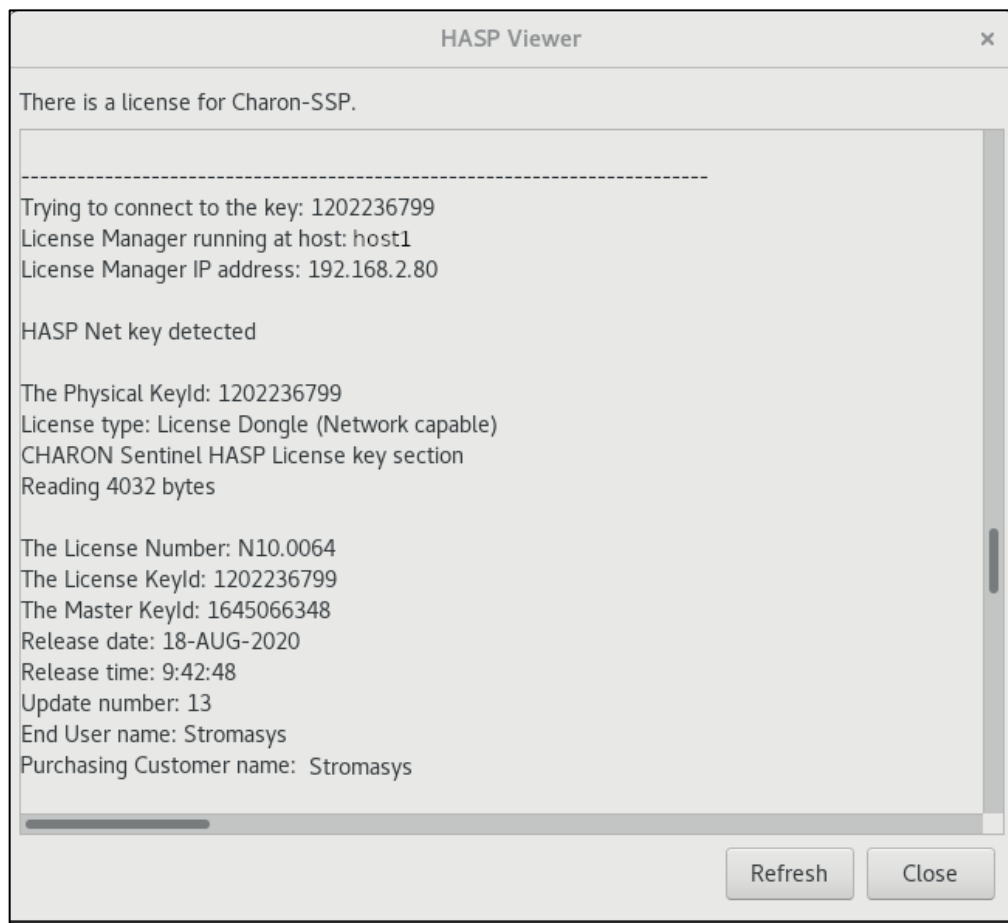


Figure 107: Charon-SSP Manager HASP license viewer showing a network license

If the license details are not displayed and a USB dongle has recently been reconnected or exchanged, it may be necessary to click on **Refresh**. To exit the window, click on **Close**.

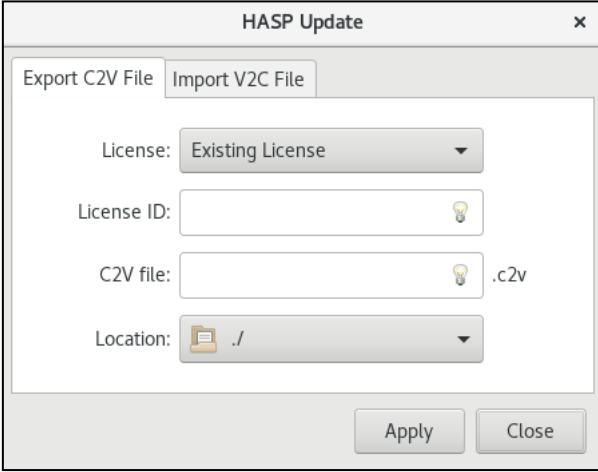
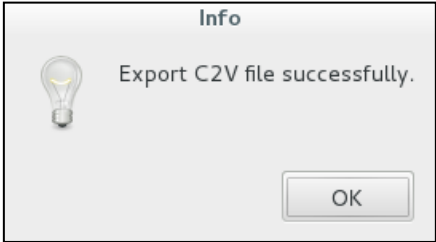
If there is more than one license available on the system, scroll down to see all available licenses.

Starting with version 4.0.x, the license viewer will show any Sentinel HASP network licenses that can be found via broadcast search irrespective of the license manager (or ACC) settings on the client side.

12.2.1.2 Gathering Customer-to-Vendor (C2V) Details

Please note: this tool has changed significantly starting with Charon-SSP version 5.0. For older Charon-SSP versions, please refer to the corresponding user's guide.

The following steps describe the process of **gathering the customer-to-vendor (C2V) data** file. This information is used by Stromasys to generate a license data file.

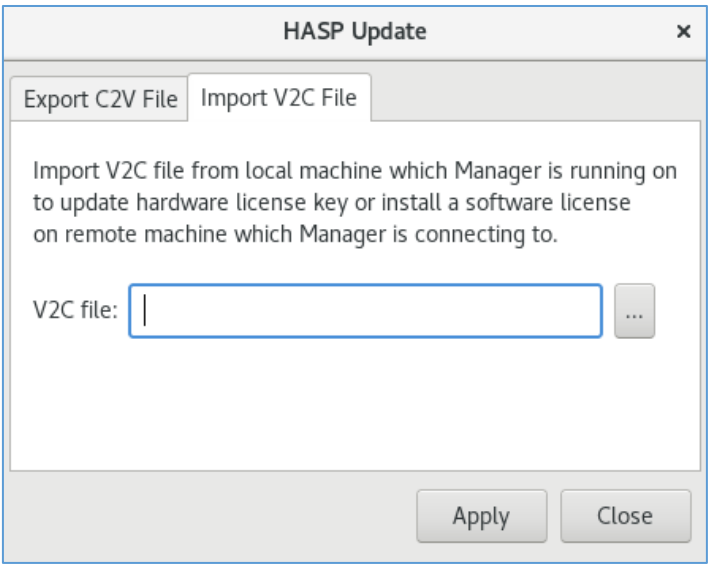
| Step | Description |
|------|--|
| 1 | <p>Open the License Update window by clicking on Tools > HASP Tools > HASP Update in the Charon-SSP Manager.</p> <p>The license update window opens.</p>  |
| 2 | <p>Select the Export C2V File tab.</p> <p>Select the type of operation in License drop-down box. There are two options to choose from:</p> |
| | <p>a Option Existing License: used to create the C2V file required for license updates (for hardware and software license updates). USB dongles are always considered to be existing licenses even at initial installation.</p> <p>Enter the ID of the license in question into the License ID field. The license ID can be found using the license viewer tool.</p> |
| | <p>b Option New Software License: used to create a fingerprint for a new software license.</p> |
| 3 | <ul style="list-style-type: none"> Provide a file name in the C2V file field. Specify where the C2V file should be saved on the local system (where the manager is running) by clicking on the path button adjacent to the Location label. Click on the Apply button. |
| 4 | <p>After a few moments the utility will respond with a confirmation box as shown in the example.</p> <p>Click on OK button to proceed.</p>  |
| 5 | <p>Locate the * .c2v file saved above (on Baremetal use SFTP to copy the file) and send it via email to Stromasys Orders Administration at orders@stromasys.com (for AMS: na.orders@stromasys.com).</p> |

12.2.1.3 Applying Vendor-to-Customer (V2C) License Updates

After sending the C2V file to Stromasys Orders Administration you will receive one or two V2C files. Then, follow the instructions below to apply the V2C files and license the software.

If there are multiple key files (is the case only for hardware dongles), it is important that they be applied in the **correct order**. The format key (*_fmt.v2c) file must always be applied first.

Instructions for **applying Vendor-to-Customer (V2C) data**:

| Step | Description |
|------|--|
| 1 | <p>Save the V2C (vendor-to-customer) files received from Stromasys Orders Administration (on Baremetal use SFTP to copy the files to the host system). Depending on the license type, you may have received one or two V2C files:</p> <ol style="list-style-type: none"> 1. A license file to format the key (for USB hardware dongles only). The file name is of the format *_fmt.v2c. 2. The license key file. The file name is of the format *.v2c. <p>Steps 2 to 4 must be performed for each of the license files, starting with the format key file (if it exists).</p> |
| 2 | <ul style="list-style-type: none"> • Open the License Update window: Click on the menu path Tools > HASP Tools > HASP Update. • Select the tab Import V2C File.  |
| 3 | Locate the saved V2C files by clicking on the button labelled "...". If a format file exists, use this file first. |
| 4 | Apply the license file by clicking on the Apply button. |
| 5 | If a second V2C file was supplied, repeat the process starting with step 2 using the second file. |

12.2.1.4 License Manager

The license manager of the Charon-SSP Manager provides quick access to selected Sentinel/Gemalto license configuration parameters in the file `/etc/hasplm/hasplm.ini`. If you need to edit this file manually (for example, because there is no graphical access to the Charon host, please refer to [this document](#) for more information. A template of the `hasplm.ini` file can be found here: `/opt/charon-agent/ssp-agent/etc/hasplm.ini`.

To access the license manager, use the menu path

Tools > HASP Tools > HASP Manager.

This will open the license manager window:

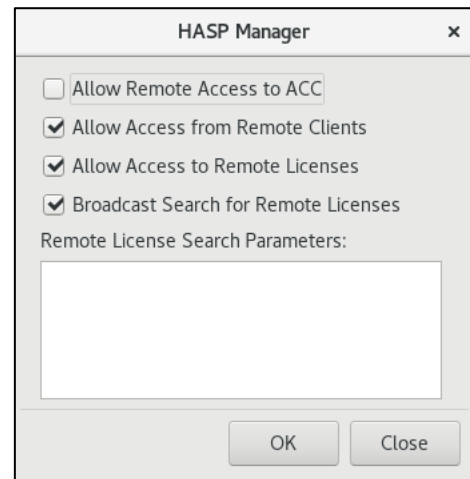


Figure 108: HASP license manager

| Parameter | Description |
|---|--|
| Allow Remote Access to ACC | Other network users can access the Sentinel Admin Control Center (ACC) and perform actions on this system. Unselected by default. The Sentinel ACC provides a web-based user interface for license management. If access is enabled, it can be accessed from a remote system via a browser (URL: <code><target-system>:1947</code>). In this case, ensure that a password is set for ACC in Configuration > Basic Settings . Corresponds to the parameter ACCremote in the [SERVER] section of hasplm.ini . |
| Allow Access from Remote Clients | Enables remote systems to access this Sentinel License Manager. This is needed if the system acts as a license server for clients on the network. Corresponds to the parameter accessfromremote in the [SERVER] section of hasplm.ini . |
| Allow Access to Remote Licenses | The system will search for remote Sentinel License Managers on the local network. This is needed if the system should use a license served by a license manager on the network. Corresponds to the parameter acesstoremote in the [SERVER] section of hasplm.ini . |
| Broadcast Search for Remote Licenses | If enabled, the system will search for remote Sentinel License Managers on the local network via broadcasts. Broadcast search uses a random UDP port from 30000 to 65535. Any firewall between the client and the server must be configured to allow these ports (also on the client side) in addition to the TCP port 1947. If disabled, the addresses of systems to search must be defined in the search parameter field. Starting with version 4.0.x, the license viewer will show any network licenses that can be found via broadcast search irrespective of the license manager (or ACC) settings on the client. Corresponds to the parameter broadcastsearch in the [REMOTE] section of hasplm.ini . |
| Remote License Search Parameters | If broadcast search is disabled: defines the specific systems that will be searched by this system in order to detect remote Sentinel License Managers and licenses served by them. Systems are specified by their IP addresses (for example, 10.1.1.17), by the broadcast addresses of an address range (for example, 10.1.1.255), or by the system name (for example, system1.example.com). When using the IPv6 protocol, use the IPv6 address format. For example, type FF02::1 to access all remote Sentinel License Managers in the local subnet. Corresponds to the parameter serveraddr in the [REMOTE] section of hasplm.ini . |

12.2.2 Managing Sentinel Licenses from the Command-Line

Charon-SSP provides two command-line utilities for the management of Sentinel/Gemalto licenses. The following sections describe how to use these tools for the following tasks:

- Viewing the license details
- Gathering Customer-to-Vendor (C2V) details
- Applying Vendor-to-Customer (V2C) license updates

A complete documentation of the two utilities (**hasp_srm_view** and **hasp_update**) used here can be found in the Appendix [Command-Line Utilities](#).

Unless otherwise specified, the path to the license command-line tools is: `/opt/charon-agent/ssp-agent/utils/license/`.

12.2.2.1 Viewing the License Details

To view the current details of the license key, use the **hasp_srm_view** utility. The following shows an example of the output generated by this utility. To display all available licenses, use the `-a11` parameter instead of `-1`.

The **hasp_srm_view** output differs slightly between software versions. The sample below is for illustrative purposes only.

| Example of license data output from hasp_srm_view |
|--|
| <pre> \$ /opt/charon-agent/ssp-agent/utils/license/hasp_srm_view -1 License Manager running at host: localhost.localdomain License Manager IP address: 127.0.0.1 The Physical KeyId: 1538162443 CHARON Sentinel HASP License key section Reading 4032 bytes License Manager running at host: localhost.localdomain License Manager IP address: 127.0.0.1 The License Number: 1002784 The License KeyId: 1538162443 The Master KeyId: 2131943932 Release date: 10-AUG-2018 Release time: 14:50:51 Update number: 12 End User name: User1 Purchasing Customer name: Stromasys Asia Pacific Virtual Hardware: SPARCstation_20, Enterprise_450 Product Name: Charon-SSP/4M, Charon-SSP/4U for Linux x64 Product Code: CHSSP-xxxxx-LI Major Version: 3 Minor Version: 0 Maximum Build: 99999 Minimum Build: 1 Host CPU supported: X64 Host Operating System required: LINUX CPU's allowed: 24 Maximum virtual memory: 32768MB Instances allowed: 3 Released product expiration date: 28-May-2019 <lines removed> </pre> |

The **hasp_srm_view** command only works with a local connection to the system containing the license. Trying to run the command across remote connections leads to the error message:

```
Sentinel HASP key not found or of improper type (1).
```

Workaround:

```
# ssh localhost /opt/charon-agent/ssp-agent/utils/license/hasp_srm_view -1
```

12.2.2.2 Gathering Customer-to-Vendor (C2V) Details

Gathering the customer-to-vendor data file requires different commands depending on the type of license.

With existing software license and for all cases where HASP USB license key is used:

To gather the customer-to-vendor (C2V) data file, use the following command (the **-key** parameter allows the selection of a specific license key):

```
$ hasp_srm_view -c2v /path/to/keydata.c2v [-key <license-id>]
```

If using this command over a **remote connection** (e.g., ssh), use the workaround provided in the section above.

Old versions (before version 2) cannot select a specific key using the **-key** parameter. There, you can use the Sentinel Admin Control Center (ACC) to gather the C2V file for the correct hardware key, or (temporarily) remove all keys but the one for which a C2V file is to be created. See section [Managing Licenses with Sentinel Admin Control Center](#).

To request a new software license:

To gather the customer-to-vendor (C2V) data file that is also called fingerprint in this case, use the following command:

```
$ hasp_srm_view -fgp /path/to/keydata.c2v
```

In both cases, the resulting file should then be sent to Stromasys Orders Administration via email, orders@stromasys.com (the email address for the AMS region is na.orders@stromasys.com).

Please note: in versions before 4.0.x, remote network licenses impact the creation of a fingerprint file. Such licenses must be temporarily disabled

12.2.2.3 Applying Vendor-to-Customer (V2C) License Updates

License keys received from Stromasys Orders Administration can be applied using the **hasp_update** command-line utility. For hardware licenses you will, in most cases, receive two V2C files to apply.

1. A license file to format the key (optional, for hardware licenses only). The file name is of the format ***_fmt.v2c**.
2. The license key file. The filename is of the format ***.v2c**.

If there are multiple key files, it is important that they be applied in the correct order. **The format key file must always be applied first.**

To apply a V2C file, use the following command (as user *root*):

```
# hasp_update u /path/to/key.v2c
```

12.2.3 Managing Licenses with Sentinel Admin Control Center

The license drivers installed to interact with the Sentinel licenses on the system come with a web-based management interface, the Sentinel Admin Control Center (ACC). This tool provides additional management functions and can be run from a local web-browser or from a web-browser on a remote system (URL: <http://<targethost-name-or-ip:1947>>).

12.2.3.1 Viewing Licenses

To display the licenses available on the local system with the Sentinel Admin Control Center (ACC), perform the following steps:

- Open a web-browser
- Go to the URL http://localhost:1947/_int_/devices.html. This option corresponds to the menu item **Sentinel Keys** (if you want to display the licenses available on a remote system, replace localhost with the correct hostname).

A screen like the following opens:

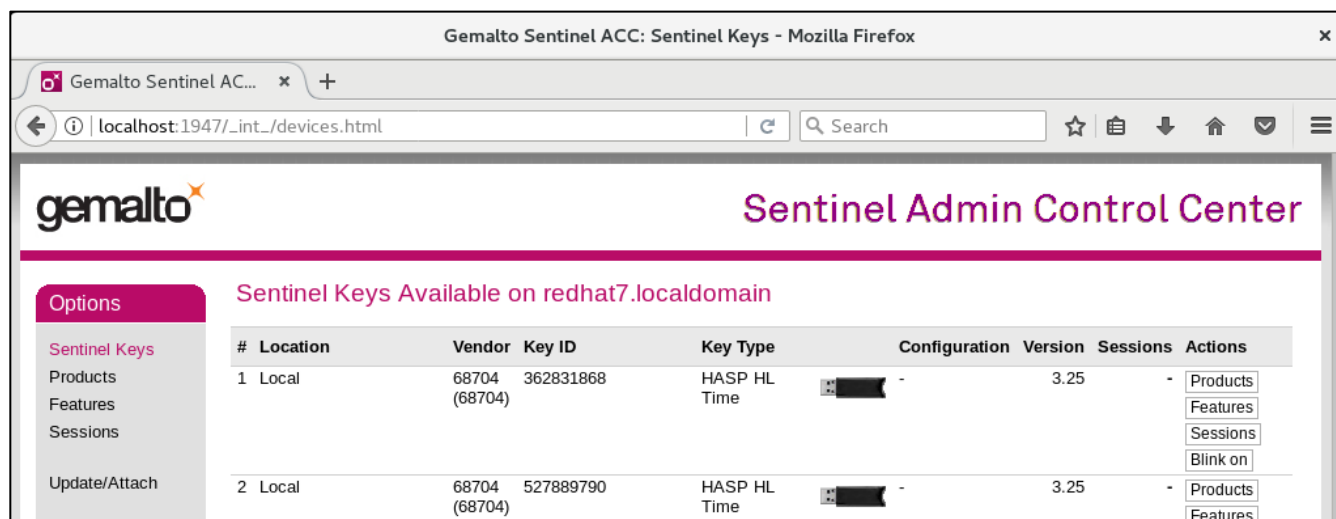


Figure 109: Display of available licenses in Sentinel ACC

The sample shows two time-limited local USB keys (Key Type: HASP HL Time).

The Sentinel ACC does not display the Charon product details. For displaying the detailed license content, please use the Charon-SSP Manager GUI or the command-line tools.

12.2.3.2 Gathering Customer-to-Vendor (C2V) Details

You can create a C2V file using the Sentinel ACC, but not a fingerprint file. Follow the steps below to create the C2V file:

1. Open the web-browser on the **Sentinel Key** page of the Sentinel ACC (URL: http://localhost:1947/_int_/devices.html):

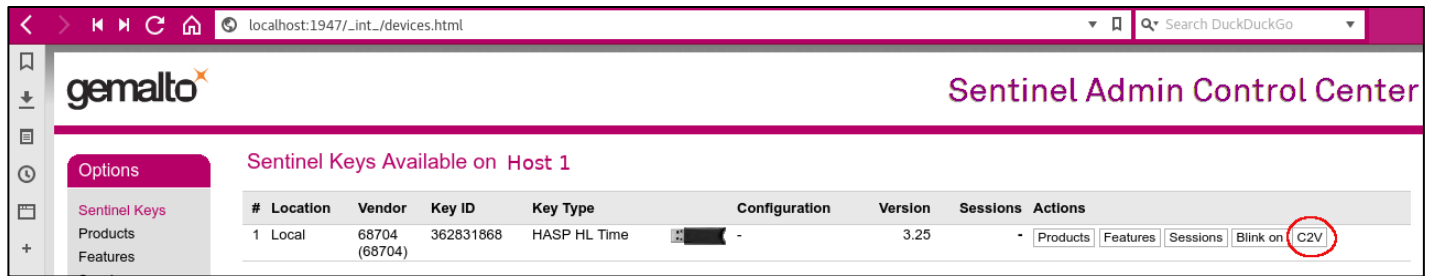


Figure 110: Sentinel Key page with C2V option

If the C2V option is not visible, you must enable it in
Configuration > Basic Settings > Generate C2V file for HASP.

2. Click on the option C2V at the right-hand side of the window for the key you plan to update.
3. A new screen will be displayed to create and download the C2V file:

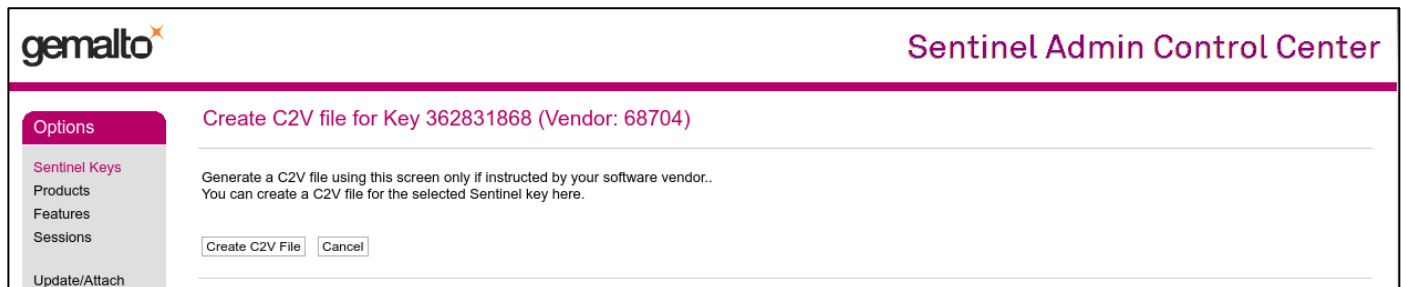


Figure 111: Sentinel ACC C2V download page

4. Download and save the C2V file.

Send the file to Stromasys Orders Administration via email, orders@stromasys.com (the email address for the AMS region is na.orders@stromasys.com).

12.2.3.3 Applying Vendor-to-Customer (V2C) License Updates

To use the Sentinel ACC on the local system to apply V2C files, open the URL http://localhost:1947/_int_/checkin.html (this is the link for the option **Update/Attach**).

The page displayed will be like the one in the following image:

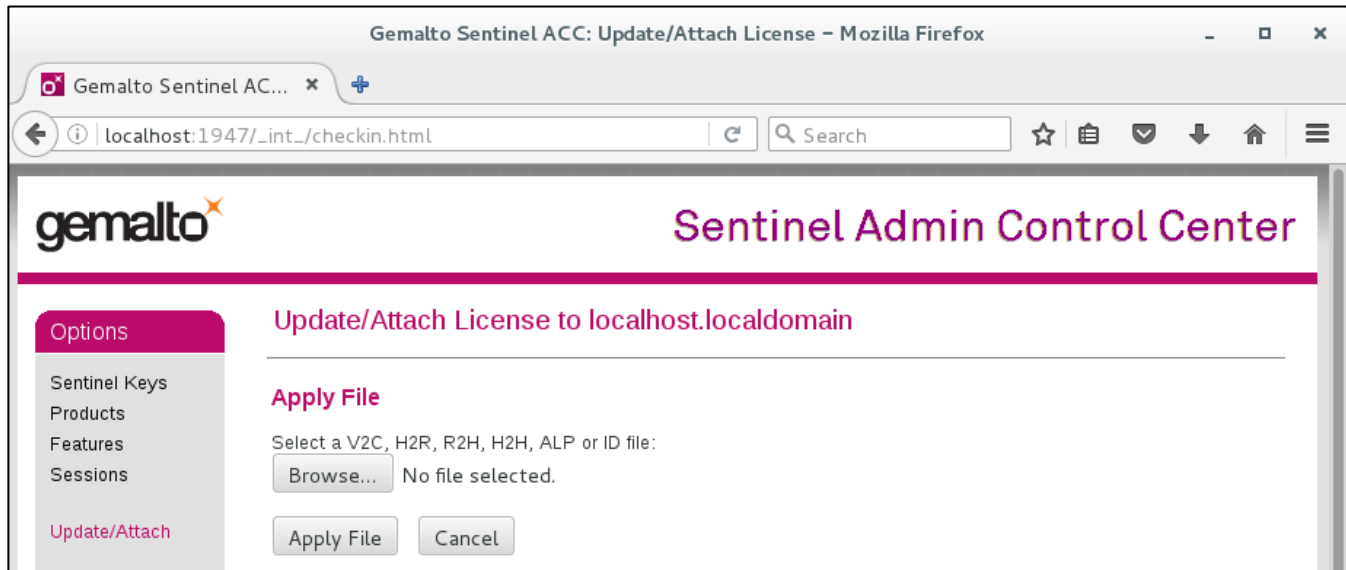


Figure 112: Sentinel ACC V2C installation page

Click on the button **Browse** (called **Choose File** on some products/versions) and select the appropriate file in the file browser window that opens.

Click on **Apply File** to apply the V2C file.

If you received two files, start with the *_fmt.v2c file, and repeat the step for the second V2C file.

12.2.3.4 Allowing Access to and from Network License Servers

The communication between license server and client systems uses the IP protocol. This means that IP connectivity must be established between client and server. If client and server are not on the same network, they need the correct routing entry (or default route) to enable this communication.

Port 1947 is used as the destination port for the communication between the License Managers. The communication uses both UDP and TCP. These protocols are used for different purposes:

- **UDP:** This protocol is used by the client hosts to discover a license server on the network. If the option Broadcast Search for Remote Licenses is enabled in ACC (Access from Remote Clients tab), the Sentinel License Manager on the client host sends UDP broadcasts to port 1947 to discover a license server. Alternatively, search parameters can be specified, in which case the Sentinel License Manager on the client sends UDP "pings" to the addresses listed. UDP is also used to notify connected License Managers in case the local License Manager is stopped.
- **TCP:** This protocol is used to connect to the discovered license servers via port 1947 and to transfer license data from them.

The following ports are used for the communication between license server and client hosts. They must not be blocked by a firewall.

- On the server side (where network license has been installed), port 1947 must be open for incoming TCP and UDP traffic to allow clients access to the license.
- On the client side, traffic is initiated using ports 30000 through 65535 as the source ports and port 1947 as the target port. If broadcast search for remote licenses is to be used, the client also receives UDP traffic from port 1947 of the license server to ports 30000 through 65535.

12.2.3.4.1 Controlling Access to the License Server on the Client Side

A client host on the network can enable or prevent the visibility of network licenses, or change the options used to discover and access network licenses provided by a license server. These tasks are also performed in the Sentinel Admin Control Center:

1. Open a web-browser and go to the URL http://localhost:1947/_int_/config_to.html (option: **Configuration / Access to Remote License Managers**).
2. This opens a configuration page like the following:

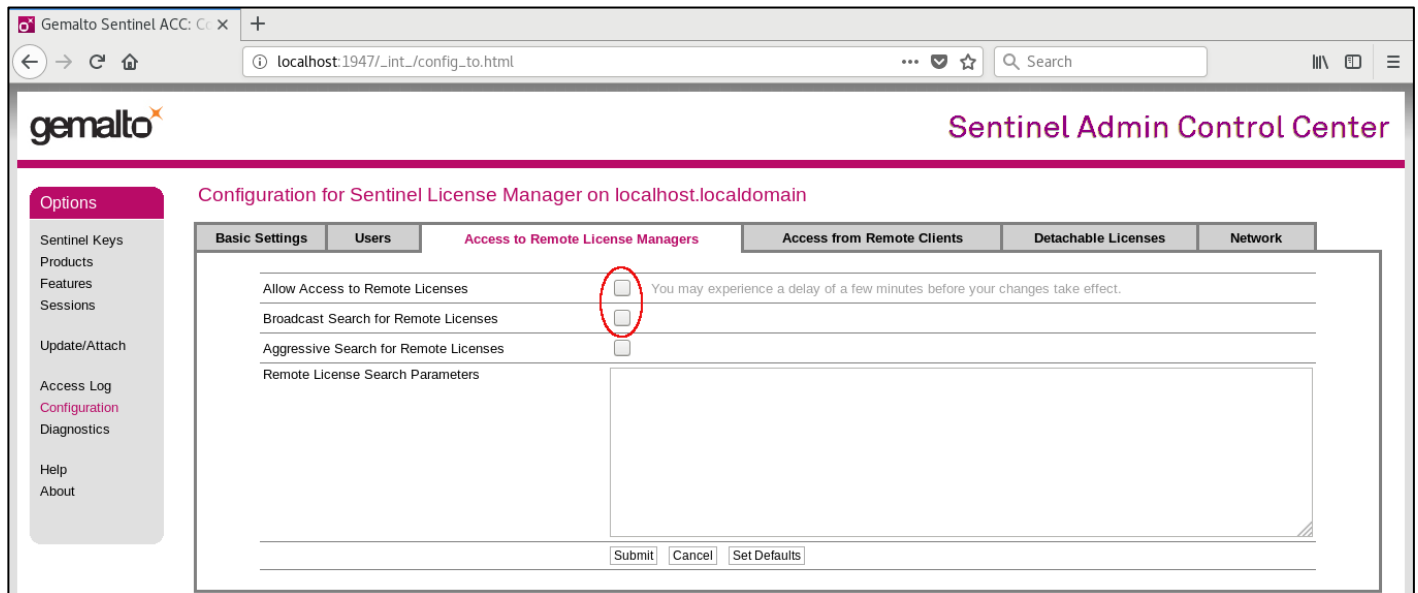


Figure 113: Permitting/Denying access to remote license managers

3. Possible actions:

- **Activate** the check-box next to the field **Allow Access to Remote Licenses** to **enable** access to license servers. Press **Submit** to save the setting.
- **Clear** the check-box next to the field **Allow Access to Remote Licenses** to **disable** access to license servers. Press **Submit** to save the setting.
- If the option **Broadcast Search for Remote Licenses** is activated, it enables a broadcast search for license servers on the local network without having to enter the address of a license server.
- If the option **Broadcast Search for Remote Licenses** is not enabled or cannot be used in the customer specific setting, you can enter specific IP addresses or host names that should be searched for network licenses in the **Remote License Search Parameters** field. Please refer to the Sentinel ACC help function for more information.

12.2.3.4.2 Controlling Access to Network Licenses on the Server Side

The license manager on the license server can be configured to allow or disallow access from remote clients to the network licenses installed on the license server.

To access this configuration option, perform the following steps:

1. Open a web-browser and go to the URL http://localhost:1947/_int_/config_from.html (option: **Configuration / Access from Remote Clients**).
2. This opens a configuration page like the following:

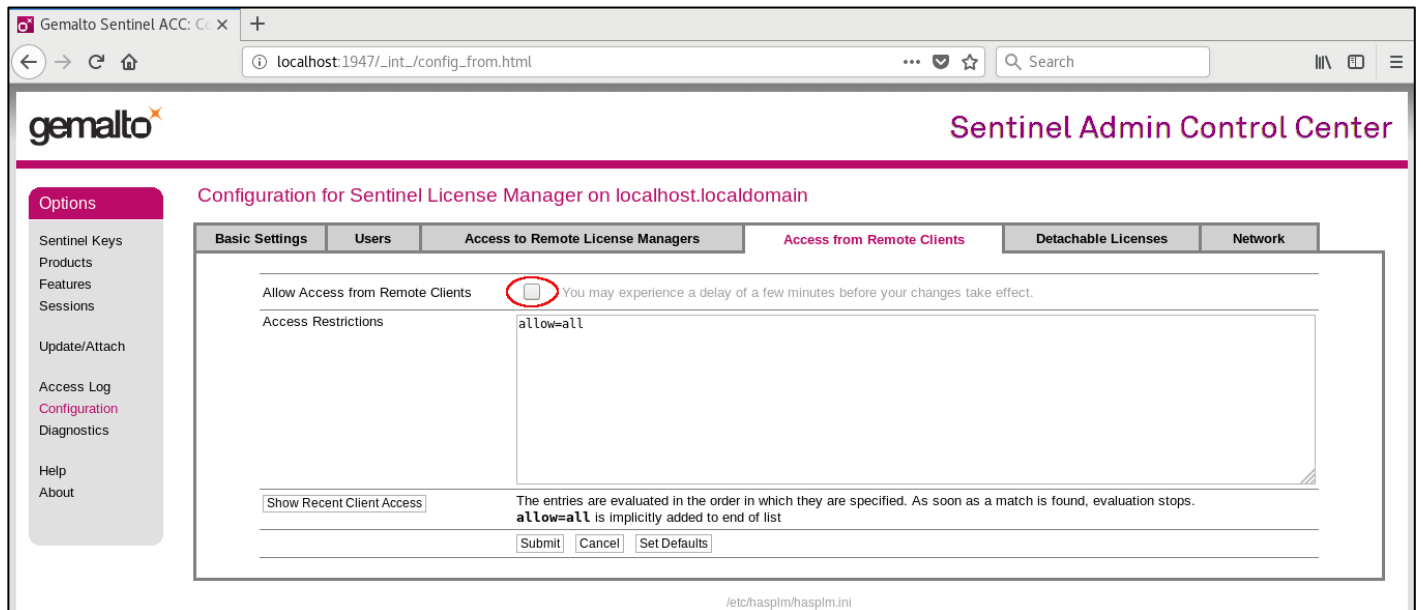


Figure 114: Permitting/Denying access from remote clients

3. Possible actions:

- To **allow access** from remote clients, **activate** the check-box next to the field **Allow Access from Remote Clients** and press **Submit** at the bottom of the page.
- To **refuse access** from remote clients, **clear** the check-box next to the field **Allow Access from Remote Clients** and press **Submit** at the bottom of the page.

The field **Access Restrictions** provides the ability to refine access rules, for example by specifying IP addresses or address ranges. Please refer to the Sentinel ACC help function for details.

12.2.3.5 Removing a Software License

A **hardware license** can simply be unplugged from the system to which it is connected.

An **obsolete software license** must be removed using the steps described below.

To remove an **obsolete software license** or a software license in **cloned state** from the system, perform the following steps:

1. Go to **http://localhost:1947** to access the Sentinel ACC and choose **Sentinel Keys**.
2. In the Sentinel ACC, locate the target *Sentinel SL AdminMode* license to be removed (a cloned license in the example).
3. Press the **Certificates** button to the right of the SL description:

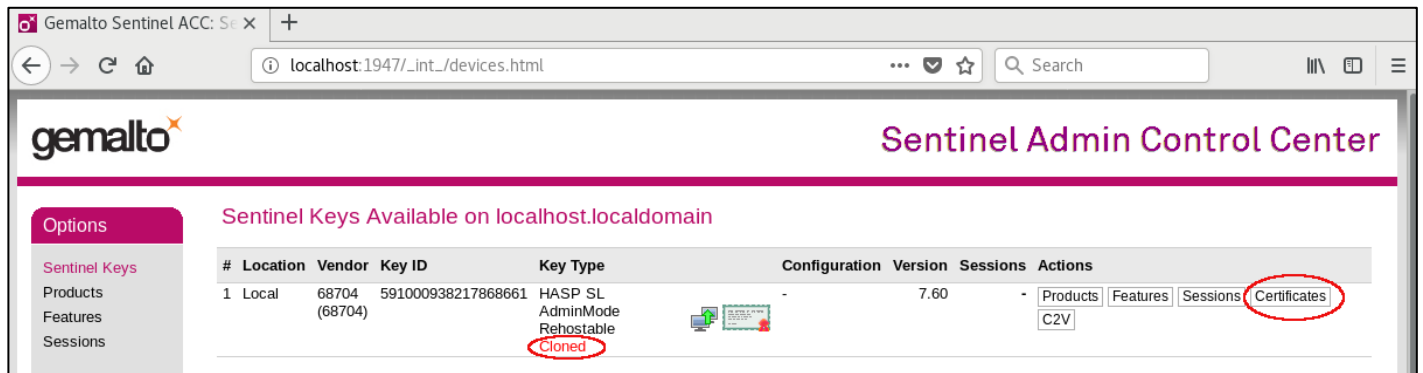


Figure 115: Sentinel Key page with certificate display option

4. In the **Certificates** section, note the name and path of the corresponding certificate:

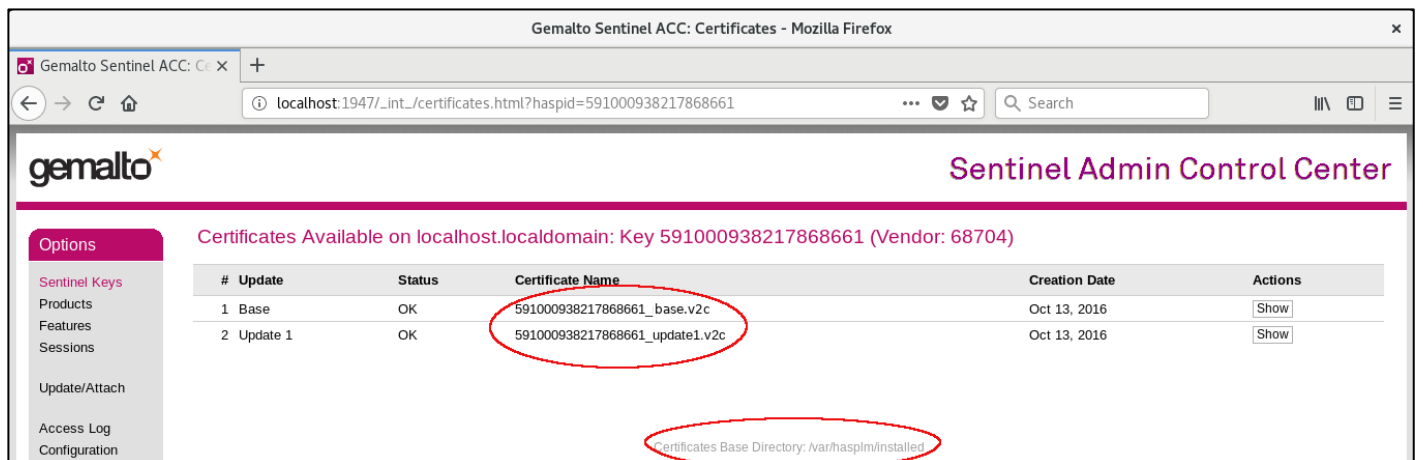


Figure 116: Certificate path information and name needed for deletion

5. As user **root** remove the certificate file(s). In the example above the files to remove are `/var/hasplm/installed/68704/591000938217868661*`
6. Wait a few minutes for the Sentinel runtime software to recognize the change. If this takes too long, reboot the system, or restart the license driver service (`# systemctl restart aksusbd`).
7. Start Sentinel HASP Admin Control Center (ACC) again to verify that the software license has been removed.

12.3 Virtual Environment (VE) License Management

Please note:

- **Charon-SSP itself does not have VE license management tools.** The following actions **must be performed on the VE license server**. However, the VE license server can be on the same system as the Charon emulator.
- If you have not installed the license server yet (and for any more in-depth information), please refer to the VE license server user's guide (see the *VE License Server guide* in [License Documentation](#)).
- The information below shows the command-line tools for license management. Starting with version 1.1.16 of the VE license server, these activities can also be performed using a **web-based management GUI**. Please refer to the appropriate *VE license server user's guide*.
- If you plan to use AutoVE, please refer to the appropriate *VE license server user's guide*.

12.3.1 Creating a VE C2V file

12.3.1.1 Running `esxi_bind` before First C2V Creation on VMware

Only applicable to VMware environments.

The `esxi_bind` command sets up the necessary communication connection between the VE license server and the ESXi host / the vCenter server.

It must be run on the license server (and the backup license server, if applicable)

- once before the first license is requested,
- and again, should the user for the access to the ESXi host / the vCenter server change.

Perform the following steps:

1. Use `ssh` to log in on the license server instance.


```
# ssh <user>@<license-server-ip>
```

 where
 1. `<user>` is the user for interactive login associated with your license server system
 2. `<license-server-ip>` the ip address of your license server system
2. Become the privileged user and run the `esxi_bind` program.
 1. Become the root user: `# sudo -i`
 2. Run the `esxi_bind` program:


```
# /opt/license-server/esxi_bind -a <address> -u <username> -p <password>
```

 where
 1. `<address>` is the IP address of the ESXi host or vCenter server
 2. `<username>` is a user on the ESXi host or vCenter server (**see notes below**)
 3. `<password>` is the password of the user
3. If the command is successful, it will create the file `/opt/license-server/config.ini` containing the connection data (the password is encrypted).

Important notes regarding the user on the ESXi host or the vCenter Server:

1. The **username on the vCenter Server** can take different forms:
 - a) Simple username
esxi_bind parameter example: `-u myusername`
 - b) Username includes a domain name in one of the following two formats:
 - i. `<domain>\<username>`
esxi_bind parameter example: `-u 'mydomain\myusername'` **(quotes are mandatory)**
 - ii. `<username>@<domain>`
esxi_bind parameter example: `-u myusername@mydomain`
2. The user on the ESXi host or the vCenter Server must have at least the following **permissions**:
 - Datastore > Allocate Space
 - VirtualMachine > Config > AddNewDisk
 - VirtualMachine > Config > RemoveDisk

12.3.1.2 Creating the VE C2V Fingerprint File

The fingerprint is collected on the license server using the **c2v** utility.

Perform the following steps to collect the fingerprint on the license server and (if applicable) the backup license server:

1. Use **ssh** to log in on the license server instance.


```
# ssh -i ~/.ssh/<mykey> <user>@<license-server-ip>
```

 where
 1. `<mykey>` is the private key of the key-pair you associated with your cloud instance (for an on-premises VMware installation where login with username/password is allowed, it is not needed)
 2. `<user>` is the user for interactive login associated with your license server instance (e.g., `opc` on OCI, `centos` for a CentOS instance on AWS, or the custom user on your VMware virtual machine or physical server; for an instance installed from a prepackaged Charon-SSP VE marketplace image, use `sshuser`)
 3. `<license-server-ip>` is the ip address of your license server system
2. Become the privileged user and run the **c2v** program.
 1. Become the root user: `# sudo -i`
 2. Run the c2v program: `# /opt/license-server/c2v --filename <my-file>.c2v --platform <my-platform>`
 where
 1. `<my-file>.c2v` is the path and name under which you want to store the fingerprint. The file type is C2V (customer-to-vendor)
 2. `<my-platform>` indicates the platform on which the license server runs (possible values: **physical, aws, oci, gcp, azure, ibm, nutanix, or esxi**)
3. Copy the resulting C2V file to your local system (unless you can send email from the license server system).

12.3.2 Installing a VE V2C File

The license data is installed on the license server using the **v2c** utility.

Perform the following steps to install the license on the license server:

1. Copy the V2C file to the license server (e.g., with SFTP).
2. Use **ssh** to log in on the license server instance.


```
# ssh -i ~/.ssh/<mykey> <user>@<license-server-ip>
```

 where
 1. *<mykey>* is the private key of the key-pair you associated with your license server instance (for an on-premises VMware installation where login with username/password is allowed, it is not needed)
 2. *<user>* is the user for interactive login associated with your license server instance (e.g., *opc* on OCI, *centos* for a CentOS instance on AWS, or the custom user on your VMware virtual machine or your physical server; for an instance installed from a prepackaged Charon-SSP VE marketplace image, use *sshuser*)
 3. *<license-server-ip>* is the ip address of your license server system
3. Become the privileged user and run the **v2c** program.
 1. Become the root user:

```
# sudo -i
```
 2. Run the v2c program:

```
# /opt/license-server/v2c -f <my-file>.v2c
```

 where *<my-file>.v2c* is the path and name under which you want to store the fingerprint. The file type is V2C (vendor-to-customer).

After the installation of the V2C file, the license server will be restarted.

12.4 License Troubleshooting

The most important tool for identifying a license problem are the log files of the Charon emulator, the VE license server, and the Linux system. Always check them first in case of a problem.

12.4.1 Log Files for License Troubleshooting

12.4.1.1 Log File location

Charon-SSP emulator log file location:

The default location of the emulator log files is in the Charon Agent hierarchy.

The default file path is **/opt/charon-agent/ssp-agent/ssp/<architecture>/<vm-name>/<vm-name>.log** where

- *architecture* denotes the emulated hardware model: **sun-4m**, **sun-4u**, or **sun-4v**, and
- *vm-name* denotes the name of the emulated SPARC system.

Please note: the path of the log emulator log files can be configured by the user to a non-default value.

VE license server log file location:

- The path to the VE license server log on the license server is **/opt/license-server/log/license.log**.
- The path to the web GUI error log file is **/opt/license-server/log/webserver.log**.

Linux system log:

The Linux logs can be viewed with the `journalctl` program. Examples:

- Identify VE license server entries (newest first): **journalctl -r -t license_server**
- Identify HASP runtime daemon entries (newest first): **journalctl -r -t aksusbd**

12.4.1.2 Log Example: Loss of License during Operation

Should the license key be removed, become invalid, or expire while the emulator is running, the emulator will detect this problem at the next regular license check (default check interval 1 hour) for HASP licenses. For VE licenses, the loss will be detected if the TCP connection to the server is lost or if the periodic license check fails. The emulator will log a warning and continue for a certain grace period as shown in the examples below:

Emulator log example for **HASP** license:

```
2020-09-08 14:48:27 WARN  HASP      License key : Unable to locate any Feature matching scope!
2020-09-08 14:48:29 ERROR HASP      Can not find license key!
2020-09-08 14:48:31 ERROR HASP      There is no valid license key.
2020-09-08 14:48:33 WARN  HASP      Charon will be terminated within 120 minutes!
2020-09-08 15:48:37 INFO  HASP      License check succeeds, back to normal.
```

Emulator log example for **VE** license:

(License loss detected)

```
2020-07-16 22:25:56 ERROR VE      Failed to connect with the license server!
2020-07-16 22:25:56 WARN  VE      Charon will be terminated within 2 hours!
```

(License server connection restored)

```
2020-07-16 23:26:01 INFO  VE      Connected with license server: 127.0.0.1
2020-07-16 23:26:01 INFO  VE      Found available license ID: 01.00000001.002.044.
2020-07-16 23:26:01 INFO  VE      Customer name: Stromasys/Testing.
2020-07-16 23:26:01 INFO  VE      Virtual hardware model Charon-SSP/4M is licensed.
2020-07-16 23:26:01 INFO  VE      Maximum concurrent instances are limited to 4.
2020-07-16 23:26:01 INFO  VE      Maximum allowed virtual CPU(s) are 4.
2020-07-16 23:26:01 INFO  VE      Maximum allowed virtualized memory is 512 MB.
2020-07-16 23:26:01 INFO  VE      Major allowed version number is 4.
2020-07-16 23:26:01 INFO  VE      Minor allowed version number is 2.
2020-07-16 23:26:01 INFO  VE      Expiration UTC time: 2020-12-31 15:55:00.
2020-07-16 23:26:01 INFO  VE      Local UTC time: 2020-07-16 15:26:01.
2020-07-16 23:26:01 INFO  VE      The license is verified, back to normal operation.
```

The log file can be viewed via in a terminal window or using the **Log** tab of the Charon-SSP Manager.

For HASP licenses only: To get an early warning for a license close to its expiration date, you can install the license expiration checker tool from Stromasys. More information, download and installation instructions can be found here in [How to install the Charon license expiration checker](#).

12.4.2 System Processes for Licensing

12.4.2.1 Sentinel HASP Processes

Correct license operation requires the corresponding system processes: **aksusbd** and **hasplmd**

There are two systemd services to start and stop these processes:

- the **aksusbd** service
- the **hasplmd** service

Checking the status of the services:

```
# systemctl status aksusbd
# systemctl status hasplmd
```

Starting and stopping the services:

The two services have a dependency such that starting one of them will start the other, stopping one of them will stop the other, and restarting one of them will restart the other.

- Starting the services:


```
# systemctl start aksusbd
# systemctl start hasplmd
```
- Stopping the services:


```
# systemctl stop aksusbd
# systemctl stop hasplmd
```

12.4.2.2 VE License Server Process

VE licensing requires that the **licensed** service must run on the license server.

Checking the status of the service:

```
# systemctl status licensed
```

Starting and stopping the service:

- Starting the service:


```
# systemctl start licensed
```
- Stopping the service:


```
# systemctl stop licensed
```

12.4.3 Other Problems

Sentinel HASP licenses:

Should there be a failure when trying to display or load the license key, review the [error codes](#) listed on the Stromasys website and the associated solutions.

VE Licenses:

Refer to the VE license server guide.

If the problem cannot be solved, please contact your VAR or the Stromasys Customer Support Center (maintenance contract required) using the details in the section Obtaining Technical Assistance.

13 Charon-SSP Software Upgrade

13.1 Upgrading via RPM Installation

This section describes the upgrade of the individual Charon-SSP software components that make up the Charon-SSP product.

Cleanly shut down any guest operating systems running in Charon-SSP emulated SPARC systems and power off the Charon-SSP emulated SPARC systems before updating the software.

Important licensing information:

- Upgrading to this Charon-SSP version from an older version **requires a license update**. Please contact your VAR or Stromasys representative to plan the update.
- Additional information for **software licenses**: starting with Charon-SSP 3.0.x, Charon-SSP contains new Sentinel license runtime versions (aksusbd package). The new versions contain important updates. However **older software licenses** (created under runtime version 2.5.1) are not compatible with the new versions. Upgrading to a new version of the runtime software from such an old version in most cases requires the installation of a new software license. Downgrading to an older license runtime version from version 7.63 or later can also cause the invalidation of a software license. Contact your VAR or Stromasys representative to discuss the best way to upgrade.
- For Charon-SSP VE: if a new version of the VE license server is distributed with the new Charon-SSP version to which an upgrade is planned, please refer to the release notes for details on relevant protocol changes. The VE license server may also need to be upgraded to avoid protocol mismatches between license client and server.

13.1.1 Host Operating System Specifics for Upgrade

All supported Linux host operating systems use RPM-packages for upgrading the emulator software (exception: the Charon Manager package for Ubuntu).

Should you need to upgrade the Charon-SSP GUI on Microsoft Windows, please refer to [Charon-SSP GUI for Microsoft Windows](#) in the appendix.

13.1.1.1 Charon-SSP Installation Packages

Stromasys will provide a download location for the software packages. If you do not have the software package(s), please contact either Stromasys or your Value-Added Reseller (VAR).

The following tables show the names of the relevant RPM installation packages. In these tables the placeholders have the following meaning:

- **{version}**: current package version. For example: 5.2.2 for Charon-SSP 5.2.2
- **{architecture}**: the 4m, 4u (+), or 4v (+) emulated SPARC architecture. The 4U+ and 4V+ packages also include 4U and 4V. The 4U+ and 4V+ images in these packages are only supported on real hardware with host operating system kernels supported for Charon-SSP/4U+/4V+.
- **{cloud-id}**: cloud for which the specific Charon-SSP AL image was released (for example, **oci** and **aws**) These RPM packages are only available as part of a cloud-specific Charon-SSP AL image, or they are provided by Stromasys to update such images.

Packages common to version 7.x and 8.x of the supported Linux distributions:

| Component | Charon-SSP software RPM package names |
|--|--|
| Charon-SSP Manager | charon-manager-ssp- <i>{version}</i> .rpm |
| Charon-SSP Director | charon-director-ssp- <i>{version}</i> .rpm |
| Charon-SSP Agent | charon-agent-ssp- <i>{version}</i> -x86_64.rpm |
| Sentinel runtime environment (for Sentinel HASP licenses) | aksusbd- <i>{version}</i> .x86_64.rpm |
| Open-source software license documents | charon-ssp-oss-license- <i>{version}</i> .rpm |

Packages specific to version 7.x of the supported Linux distributions:

| Component | Charon-SSP software RPM package names |
|---|---|
| The emulator software itself: Charon-SSP 4M, 4U, 4U+, 4V, and 4V+ | <u>Emulator packages to be used with Sentinel HASP licenses:</u> charon-ssp- <i>{architecture}</i> - <i>{version}</i> .el7-x86_64.rpm |
| | <u>Emulator packages to be used with a VE license server:</u> charon-ssp- <i>{architecture}</i> - <i>{version}</i> .ve.el7-x86_64.rpm |
| | <u>Emulator packages used to update cloud-specific AL installations:</u> charon-ssp- <i>{architecture}</i> - <i>{version}</i> . <i>{cloud-id}</i> .market-x86_64.rpm |

Packages specific to version 8.x of the supported Linux distributions:

| Component | Charon-SSP software RPM package names |
|---|--|
| The emulator software itself: Charon-SSP 4M, 4U, 4U+, 4V, and 4V+ | <u>Emulator packages to be used with Sentinel HASP licenses:</u> charon-ssp- <i>{architecture}</i> - <i>{version}</i> .el8-x86_64.rpm |
| | <u>Emulator packages to be used with a VE license server:</u> charon-ssp- <i>{architecture}</i> - <i>{version}</i> .ve.el8-x86_64.rpm |

Ubuntu installation packages:

| Product | Charon-SSP software package names (DEB format for Ubuntu) |
|---------------------|---|
| Charon-SSP Manager | charon-manager-ssp- <i>{version}</i> .deb |
| Charon-SSP Director | charon-director-ssp- <i>{version}</i> .deb |

13.1.1.2 Upgrade Commands on Supported Host Systems

The following table provides an overview of the upgrade commands. For details, please refer to the relevant man-pages on Linux.

| | RPM package installation |
|---|--|
| Package manager (uses repositories, takes care of dependencies, etc.) | <pre># yum update <package-name> <path-to-package></pre> <p>Or (version 8.x of the supported Linux distributions):</p> <pre># dnf update <package-name> <path-to-package></pre> <ul style="list-style-type: none"> • If the path to an RPM file is provided, yum and dnf use a local package installation. • If only the package name is provided, yum and dnf will try to use the configured repositories. • The dnf command was introduced in versions 8.x of the supported Linux distributions. Its syntax is very similar to yum. |
| Command to install individual local packages | <pre># rpm -u <path-to-package></pre> |

| | DEB package installation (Ubuntu) |
|--|---|
| Command to install individual local packages | <pre># dpkg -i <package-name></pre> <p>Only for Charon Manager installation.</p> |

All update steps must be performed from a privileged account as denoted by the '#' prompt

13.1.2 Upgrading the Charon-SSP Software Packages

This section describes the package update for RPM-based Linux distributions.

Please note:

- The versions of the Charon-SSP packages installed on one system must match.
- **Exception:** It is possible to have several versions of the Charon-SSP Manager on a system to manage remote systems with versions different from the local system.
- If you want to retain an old version of the Charon-SSP Manager, copy the content of the directory **/opt/charon-manager** to another directory, e.g., **/opt/charon-manager-<version>** before upgrading. Later, you can use this version to manage remote systems with matching versions.

To upgrade the Charon-SSP RPM packages, perform the following the steps.

| Step | Description |
|------|--|
| 1 | Shut down any guest systems running in Charon-SSP instances and power off the Charon-SSP instances before updating the software. |
| 2 | Run the upgrade command for Charon-SSP (assumption: all packages are in current working directory) |
| | <pre># yum update charon*.rpm</pre> Version 7.x of the supported Linux distributions |
| | <pre># dnf update charon*.rpm</pre> Version 8.x of the supported Linux distributions |
| 3 | Run the upgrade command for the Sentinel HASP (if a new version is included with Charon-SSP). |
| | <pre># yum update aksusbd*.rpm</pre> Version 7.x of the supported Linux distributions |
| | <pre># dnf update aksusbd*.rpm</pre> Version 8.x of the supported Linux distributions |
| 4 | Restart the virtual SPARC systems. |

Of course, you can run the update command individually for each package.

The following shows a sample upgrade of Charon-SSP with the packages in the current working directory:

```

Upgrading Charon-SSP RPM packages

# yum update charon*.rpm

<lines removed>

Resolving Dependencies
--> Running transaction check
---> Package charon-agent-ssp.x86_64 0:4.3.4-1 will be updated
---> Package charon-agent-ssp.x86_64 0:5.4.2-1 will be an update
---> Package charon-director-ssp.x86_64 0:4.3.4-1 will be updated
---> Package charon-director-ssp.x86_64 0:5.4.2-1 will be an update
---> Package charon-manager-ssp.x86_64 0:4.3.4-1 will be updated
---> Package charon-manager-ssp.x86_64 0:5.4.2-1 will be an update
---> Package charon-ssp-4m.x86_64 0:4.3.4.el7-1 will be updated
---> Package charon-ssp-4m.x86_64 0:5.4.2.el7-1 will be an update
---> Package charon-ssp-4u.x86_64 0:4.3.4.el7-1 will be updated
---> Package charon-ssp-4u.x86_64 0:5.4.2.el7-1 will be an update
---> Package charon-ssp-4v.x86_64 0:4.3.4.el7-1 will be updated
---> Package charon-ssp-4v.x86_64 0:5.4.2.el7-1 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                Arch    Version      Repository                               Size
=====
Updating:
charon-agent-ssp       x86_64  5.4.2-1      /charon-agent-ssp-5.4.2-x86_64         32 M
charon-director-ssp   x86_64  5.4.2-1      /charon-director-ssp-5.4.2             287 k
charon-manager-ssp    x86_64  5.4.2-1      /charon-manager-ssp-5.4.2              6.0 M
charon-ssp-4m         x86_64  5.4.2.el7-1  /charon-ssp-4m-5.4.2.el7-x86_64       6.1 M
charon-ssp-4u         x86_64  5.4.2.el7-1  /charon-ssp-4u-5.4.2.el7-x86_64       24 M
charon-ssp-4v         x86_64  5.4.2.el7-1  /charon-ssp-4v-5.4.2.el7-x86_64       24 M

Transaction Summary

=====
Upgrade  6 Packages

Total size: 92 M
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction

<lines removed>

Updated:
  charon-agent-ssp.x86_64 0:5.4.2-1      charon-director-ssp.x86_64 0:5.4.2-1
  charon-manager-ssp.x86_64 0:5.4.2-1    charon-ssp-4m.x86_64 0:5.4.2.el7-1
  charon-ssp-4u.x86_64 0:5.4.2.el7-1    charon-ssp-4v.x86_64 0:5.4.2.el7-1

Complete!

```

13.2 Upgrading the Charon-SSP Baremetal Distribution

Stromasys may provide updates to the Charon-SSP Baremetal system. For major updates, this could mean that the system must be updated from a product ISO following the system installation process (the previously created Charon configurations and virtual disks will be preserved). In other cases, Stromasys may provide the customer with Baremetal-specific software packages (*.bm packages) to update the Charon-SSP product packages.

Please note:

- An update via a *.bm package is not possible if updating to Charon-SSP 5.3.11 or higher from an earlier version due to the change of the major version of the underlying operating system. The packages are not compatible between the two major operating system versions.
- Starting with version 4.1.29, a new Baremetal installation creates a **root (“/”) partition with a size of 15GiB**. Baremetal systems that were originally installed from earlier versions have a root partition size of **5GiB**. The partition size is not changed during an upgrade. Hence, the user must carefully evaluate if the available space on the root partition is still sufficient. Starting with version 5.3.11, Charon-SSP Baremetal is based on Rocky Linux 8.5. The change from Linux 7 to Linux 8 caused an increase in the space required on the root partition (at least 4.1GiB) leaving very little free space on a 5GiB root partition. **In such cases, a new installation should be considered if possible (after backing up all relevant Charon and user data).**
- If upgrading from a Charon-SSP version before 5.3.11, the underlying operating system will be upgraded **from Centos 7 to Rocky Linux 8**. This major version change will cause some significant system management changes. For example, the network configuration and user configurations may be reset.

Important steps before an upgrade:

- **Cleanly shut down all guest operating systems running in Charon-SSP emulated SPARC systems and power off the emulated SPARC systems.**
- Backup the Charon-SSP VM configuration data.
- In case of a full system upgrade (using an ISO file), also backup your container files and other customer-specific data. You can copy the data to another system via SFTP or to an external device using the Storage Manager and the File Manager

Important licensing information:

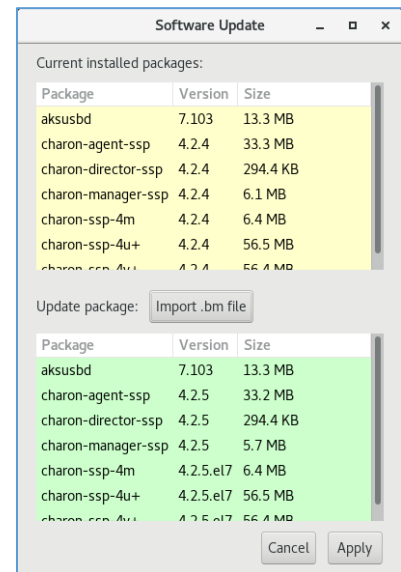
- Upgrading to this Charon-SSP version from an older version requires a license update. Please contact your VAR or Stromasys representative to plan the update.
- Additional information for **HASP software licenses**: starting with Charon-SSP 3.0.x, Charon-SSP contains new Sentinel license runtime versions (aksusbd package). The new versions contain important updates. However **older software licenses** (created under runtime version 2.5.1) are not compatible with the new versions. Upgrading to a new version of the runtime software from such an old version in most cases requires the installation of a new software license. Downgrading to an older license runtime version from version 7.63 or later can also cause the invalidation of a software license. Contact your VAR or Stromasys representative to discuss the best way to upgrade.

13.2.1 Upgrading Charon-SSP Packages Using the Update App

Please note: this upgrade path is only possible within the same Linux major version (7 or 8). The *.bm packages of Charon-SSP 5.3.11 and higher cannot be used to upgrade the Charon-SSP version on a Charon-SSP Baremetal installation prior to 5.3.11. Reason: the underlying operating system changed from Linux 7 to Linux 8 with Baremetal version 5.3.11. Workaround: use the conventional RPM installation kit for Linux 8.

To update the Charon-SSP product packages using *.bm packages, perform the following steps:

| Step | Description |
|------|--|
| 1 | Shut down all guest operating systems running in Charon-SSP emulator instances, power off the instances, and backup the Charon-SSP VM configuration data. |
| 2 | If using SFTP, copy the *.bm package to the charon account of the Charon-SSP Baremetal system (for using a USB device see step 2a). The package will be stored under <code>/charon/storage</code> . |
| 2a | If using a USB device, attach it to the system. Then, you can mount it via the command-line or the Charon-SSP Storage Manager (Tools > Charon Baremetal > Storage Manager). If mounted using the Storage Manager, it will be mounted on <code>/charon/storage/media/<file-system-uuid></code> . If required, use the File Manager (Tools > Charon Baremetal > File Manager) to copy the *.bm file to another location under <code>/charon/storage</code> . When done, close the File Manager , unmount the file system using the Storage Manager , and remove the USB device. If you later want to delete the *.bm file, you can do this via the command-line or the File Manager . |
| 3 | Start the update tool (Toolbox > Update). |
| 4 | Click on Import .bm file in the middle of the update tool window. |
| 5 | A file-browser opens. Navigate to the storage location of the *.bm file and select the *.bm file copied to the system before. Confirm your selection to start the import. |
| 6 | <p>If the .bm file contains appropriate upgrade packages, the package list will be displayed in the lower part of the update application window as shown in the image to the right.</p> <p>To start the update, click on Apply. You will be prompted for your management password. After the update has been completed, the installed package list will show in the upper part of the update application window.</p> <p>Additional information:</p> <ul style="list-style-type: none"> The update operation will be done for all packages included in the .bm package. To reverse this operation, a new .bm package or a manual downgrade via the command-line is needed. The information on the Charon Baremetal home screen tab will continue to show the original base version of the system, i.e., the version that was installed from the ISO file. In addition, it will show the new Charon-SSP product version. |
| 7 | Restart your Charon-SSP instances. |



13.2.2 Upgrading Charon-SSP Baremetal Using an ISO file or Installation Medium

To update the Charon-SSP Baremetal system using an ISO installation medium, follow the steps described in [Charon-SSP Baremetal Installation](#). If the existing system disk is selected as the installation target, the system will be upgraded to the new version using the same procedure as a new installation. The difference is that the data in `/charon/storage` and the previously created Charon-SSP configurations are preserved.

If a **system upgrade instead of an installation is performed**, the host system files and host applications as well as the Charon-SSP Baremetal-specific applications are upgraded while important Charon-SSP configuration data and container files are preserved. However, note the following:

- If performing a system upgrade, the host system will be shut down to boot from the installation CD. **Make sure to cleanly shutdown any guest systems running in Charon-SSP instances**, and to power off the instances before shutting down the host system.
- After shutting down any running guest system and stopping the emulator, **back up all your configuration data, container files, and other customer-specific data before performing an upgrade**. In the case of a downgrade, bear in mind that an older software version may not understand all parameters in a configuration created with the new software version. You can copy the backup data and container files to an external device or across the network to another system.
- If upgrading from a Charon-SSP version before 5.3.11, the underlying operating system will be upgraded from CentOS 7 to Rocky Linux 8. This major version change will cause some significant system management changes. For example, the network configuration and user configurations will be reset.
- Any **customer-specific configurations, applications, and additional Linux packages** added to the original Baremetal installation will be lost during an upgrade using the ISO image (comparable to re-installing a normal Linux system).
- At the time of writing, desktop settings made for the login screen (e.g., keyboard and display settings) via the Settings app are lost during an upgrade using the ISO image. The default settings are restored.
- Starting with version 4.2, the Baremetal system has a **root (“/”) partition size** of 15GiB. Baremetal installations made with earlier versions have a root partition size of 5GiB. The partition size is not changed during an upgrade. Hence, the user must carefully evaluate if/which additional applications should be installed on the system and the available space on the root partition should be verified before any installation.

13.3 Upgrading the Charon-SSP Barebone Distribution

The former Charon-SSP Barebone distribution has been merged with the previous Baremetal distribution into the new Baremetal distribution. Therefore, there will be no Barebone-specific upgrades. However, the Charon-SSP software of pre-existing Barebone installations can be upgraded using RPM packages as described above.

The Charon-SSP/4U+/4V+ and the PCI pass-through drivers use kernel modules that can only be loaded if using a Linux kernel supported by Stromasys. Therefore, do not upgrade the kernel without being advised to do so by Stromasys.

13.4 Cloud Image Upgrade

13.4.1 RPM-based Upgrade

In certain cases, Stromasys will provide RPM packages to upgrade systems based on Charon-SSP AL and VE images. In such cases, please follow the instructions for updating Charon-SSP conventional product.

13.4.2 Upgrading by Creating a New Cloud Instance

One option to upgrade a cloud-specific Charon-SSP image to a newer version is the creation of a new instance based on a new version of the Charon-SSP cloud-specific image.

This may be useful if

- all Charon instance files (e.g., vdisks and ISO images) are on a separate disk storage volume that can easily be moved to a new instance,
- the overall configuration of the Charon-SSP host system is not very complex, i.e., can be recreated without much time and effort,
- a major host operating system upgrade is required.

Steps (only meant for illustration - the details could vary depending on the customer environment):

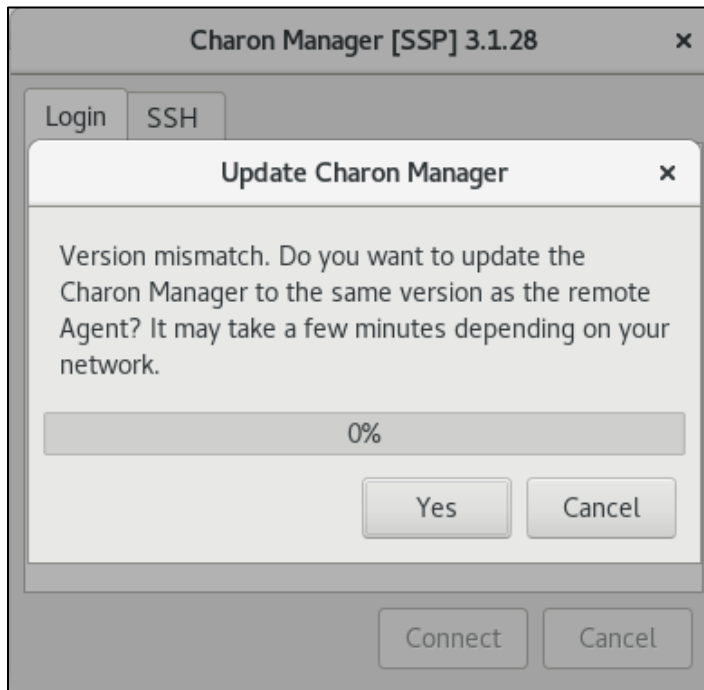
- Create an instance with the new Charon-SSP version in the same subnet as the old instance (same security group and key pair as old instance).
- Shut down guest systems and stop running emulator instances on the old host system.
- Back up emulated SPARC disks and configuration data on the old instance to the separate disk volume.
- If applicable, back up important system configuration files to the separate disk volume.
- Stop the old instance.
- Copy the Charon Manager kit from the new instance and install it on your local Linux system, or connect to the new instance via Charon Manager to initiate the automatic Charon Manager update.
- Move the disk volume(s) and (if applicable) static IP addresses to the new instance.
- Import the virtual SPARC configurations on the new system.
- Adapt the host system configuration as needed.
- Start the guest systems.
- If everything works, terminate the old instance.

Please refer to your cloud provider's documentation for cloud-specific details.

13.5 Charon Manager Automatic Update

If you connect to a Charon host system using an older version of the Charon Manager than the one used on the host system, you will be offered an automatic update to the version of the Charon host system.

The following image shows a sample:



Confirm the upgrade by clicking on **Yes**. This will initiate the download and installation of the new version. If you work as a non-privileged user, you will be prompted for your password.

To decline the automatic update, click on **Cancel**.

This mechanism also works for downgrades (i.e., if the Charon Manager version is newer than the version of the Charon host system it connects to, it can be automatically downgraded to the version of the target system).

14 Charon-SSP Software Deinstallation

14.1 Software Deinstallation on Conventional RPM Installation

This section describes the removal of the Charon-SSP Software on Linux. Should you need to remove the Charon-SSP GUI from Microsoft Windows, please refer to [Charon-SSP GUI for Microsoft Windows](#) in the appendix.

Shut down any running components of Charon-SSP before removing the software.

14.1.1 Removing the Sentinel HASP Software

To remove the Sentinel HASP software, execute the following commands on the Linux host system as shown below.

| Product Name | Deinstallation command for RPM packages |
|------------------------|---|
| Sentinel HASP Software | # rpm -e aksusbd |

Removing the Sentinel HASP license software does not remove any site-specific configuration files in */etc/hasplm*.

14.1.2 Removing the Charon-SSP Packages on Linux

To deinstall the Charon-SSP software for Linux, execute the commands as shown in the table below.

| Product Name | Deinstallation commands for RPM packages |
|---------------------------------|--|
| | <i>Deinstallation of individual packages (rpm example)</i> |
| Charon-SSP/4M | # rpm -e charon-ssp-4m |
| Charon-SSP/4U(+) | # rpm -e charon-ssp-4u # rpm -e charon-ssp-4u+ |
| Charon-SSP/4V(+) | # rpm -e charon-ssp-4v # rpm -e charon-ssp-4v+ |
| Charon-SSP Manager | # rpm -e charon-manager-ssp |
| Charon-SSP Director | # rpm -e charon-director-ssp |
| Charon-SSP Agent | # rpm -e charon-agent-ssp |
| Charon-SSP open-source licenses | # rpm -e charon-ssp-oss-license |
| | <i>Deinstallation of all Charon packages (yum/dnf example)</i> |
| All packages named charon* | # yum erase charon* or (Linux versions 8.x): # dnf erase charon* |

| DEB package removal (Ubuntu) | |
|--------------------------------|---------------------------------|
| Package deinstallation command | # dpkg -r <package-name> |

During the deinstallation process, only the Charon-SSP for Linux software is removed. All user data, including virtual disks, configuration files, and virtual tapes are left untouched.

14.2 Software Deinstallation on Baremetal System

If the hardware is to be re-purposed, the system should be re-installed thereby removing Charon-SSP Baremetal and all its data from the system.

Save any configuration data and Charon-SSP guest-system disk files before re-installing the system.

14.3 Software Deinstallation on Cloud-Specific Images

14.3.1 Deinstalling the Charon Manager on Management System

To deinstall the Charon Manager on your management system, use the commands described in the deinstallation section for systems based on conventional RPM installations.

14.3.2 Terminating the Charon-SSP Cloud Instance

To permanently remove your Charon-SSP cloud instance, select your instance from the list of active virtual machines and select to **terminate** or **delete** the instance. Please refer to your cloud provider documentation for details about this operation.

This will stop the instance and remove it. Unless your data (configuration files, vdisk containers, etc.) was stored on a separate disk volume, it will also be removed (this may be dependent on cloud-specific settings).

Please note: make sure you backup any data you wish to retain before terminating an instance.

A Appendix – Charon-SSP GUI for Microsoft Windows

A.1 Charon-SSP GUI Installation on Windows

This chapter describes the installation of the Microsoft Windows components of the Charon-SSP product and recommended or necessary post-installation tasks.

A.1.1 Prerequisites and General Information

Stromasys provides Charon-SSP Manager and Charon-SSP Director for Microsoft Windows.

Supported versions of Microsoft Windows:

- Microsoft Windows 7, 8, 10

Installation packages for Microsoft Windows:

The following packages are provided by Stromasys:

- Charon-SSP Manager: charon-manager-ssp-*{version}*.zip
- Charon-SSP Director: charon-director-ssp-*{version}*.zip

Other prerequisites:

During the installation, the required .NET run-time version 3.5 will be downloaded and installed if it is not yet available on the system.

A.1.2 Installing the Charon-SSP Manager for Microsoft Windows

The following sections cover the procedures for installing the Charon-SSP Manager for Windows software and the associated post-installation tasks. Please note: the installation process may also install a required version of the .NET software. This happens automatically, if needed, and is not described in this section.

The Charon-SSP Manager for Windows software is shipped as a zipped archive package.

To complete the installation, use the following instructions.

| Step | Description |
|------|---|
| 1 | Right-click the zip archive charon-manager-ssp-{version}.zip and select Extract All... |
| 2 | A window titled Extract Compressed (Zipped) Folders opens. In this window: <ul style="list-style-type: none"> • Click the Show extracted files when complete checkbox. • Click the Extract button. |
| 3 | A new Windows Explorer window opens showing the extracted packages. |
| 4 | Double-click the <code>setup.exe</code> executable to begin the installation. |
| 5 | If you are presented with an Open File - Security Warning window, click the Run button. |
| 6 | You should now see the Charon-SSP Manager Setup Wizard . To proceed with the installation, click the Next button. If the Windows Installer reports that Charon-SSP Manager for Windows is already installed, you must deinstall the currently installed software before you can install a different version. Normally, several versions can coexist. |
| 7 | To accept the default installation options, simply click on Next without modifying any options. Alternatively, the following installation options can be adjusted: <ul style="list-style-type: none"> • Click on Browse to select an alternative installation target. • Click the appropriate radio button, Everyone or Just for Me, to specify system-wide or private installation respectively (the system-wide installation will prompt for the administrator password if you are not using the administrator account). • To determine the approximate disk usage after the installation, click the Disk Cost button. Once all options have been set, click on Next . |
| 8 | Proceed with the installation by clicking on Next . |
| 9 | Once the installation has completed, click on Close to exit the SSP-Manager Setup Wizard. |
| 10 | The installation process creates: <ul style="list-style-type: none"> • A Charon Manager icon on the desktop • A Charon Manager entry in the Start menu (folder Stromasys) |

Starting with Charon-SSP version 2, it is supported to have **several versions of the Charon-SSP Manager** on a system to manage remote systems with versions different from the local system. The Charon-SSP Manager installation preserves older versions. Later, you can use these versions to manage remote systems with matching versions.

Post-installation tasks:

Stromasys recommends that you check **Windows Update** for any critical Microsoft .NET Framework updates, and to install them if any are available.

Upon first login, you will be prompted to set the management password if it has not been set before. See [Starting the Charon-SSP Manager](#) for more information.

A.1.3 Installing the Charon-SSP Director for Microsoft Windows

The following sections cover the procedures for installing the Charon-SSP Director for Windows software. The Charon-SSP Director requires the Charon-SSP Manager to be installed on the same system.

The Charon-SSP Director for Windows software is shipped as a zipped archive package. To complete the installation, use the following instructions.

Please note: the installation process may also install a required version of the .NET software. This happens automatically, if needed, and is not described in this section.

Installing Charon-SSP Director for Microsoft Windows:

| Step | Description |
|------|---|
| 1 | Right-click the zip archive charon-director-ssp-{version}.zip and select Extract All... |
| 2 | A window titled Extract Compressed (Zipped) Folders opens. In this window: <ul style="list-style-type: none"> • Tick the Show extracted files when complete checkbox. • Click the Extract button. |
| 3 | A new Windows Explorer window opens showing the extracted packages. |
| 4 | Double-click the <code>setup.exe</code> executable to begin the installation. |
| 5 | If you are presented with an Open File - Security Warning window, click on Run . |
| 6 | You should now see the Charon-SSP Director Setup Wizard . To proceed with the installation, click on Next . If the Windows Installer reports that Charon-SSP Director for Windows is already installed, you must uninstall the currently installed software before you can install a different version. |
| 7 | To accept the default installation options, simply click on Next without modifying any options. Alternatively, the following installation options can be adjusted: <ul style="list-style-type: none"> • Click on Browse to select an alternative installation target. • Click the appropriate radio button, Everyone or Just for Me, to specify system-wide or private installation respectively (system-wide installation will prompt for the administrator password if you are not using the administrator account). • To determine the approximate disk usage after installation, click the Disk Cost button. Once all options have been set, click on Next . |
| 8 | Proceed with the installation by clicking on Next . |
| 9 | Once the installation has completed, click on Close to exit the SSP-Director Setup Wizard. |
| 10 | The installation process creates: <ul style="list-style-type: none"> • A Charon Director icon on the desktop • A Charon Director entry in the Start menu |

A.1.4 Using the Charon-SSP GUI on Microsoft Windows

Most features of Charon Manager and Charon Director are the same as for the Linux version. Hence, please refer to the sections [Using the Charon-SSP Director](#) and [Using the Charon-SSP Manager](#) in the main part of the UG.

This section will only show some of the differences between the Linux and the Windows versions.

Charon-SSP Manager on Windows – Differences

The Charon-SSP Manager GUI on Windows is almost identical to the Linux version. However, there are some small differences:

- The serial console access on Windows is implemented via **PUTTY**. Hence, the icon bar at the top of the Charon-SSP Manager screen contains an additional terminal symbol for the serial console.
- The **Virtual Machine** menu contains a settings option for **PUTTY** instead of **Console Options**.
- Xserver management is not integrated with the Charon-SSP Manager on Windows. Hence, the menu option **X11 Server** is not available in the **Tools** menu.

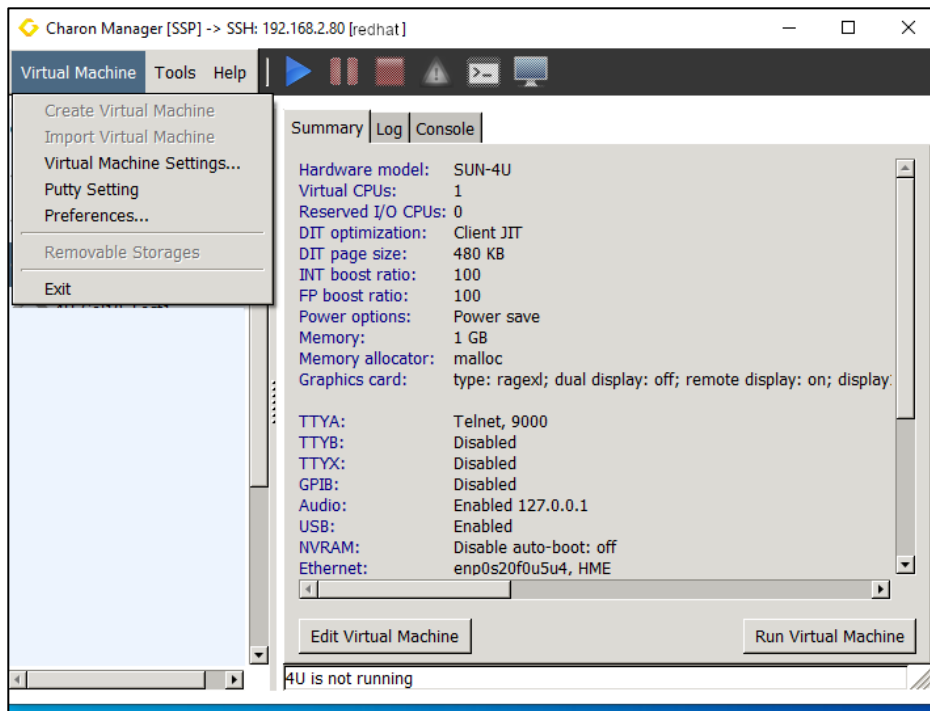


Figure 117: Charon-SSP Manager on Windows – Main window

A.1.4.1 Charon Manager on Windows – Serial Console Access

The Windows version of the Charon-SSP Manager does not have the same built-in console functionality as the Baremetal and Linux versions. Instead, **PuTTY** is used to access the console of a virtual SPARC system on a remote Charon-SSP host. The following image shows the different layout of the Charon-SSP Manager window on a Windows system:

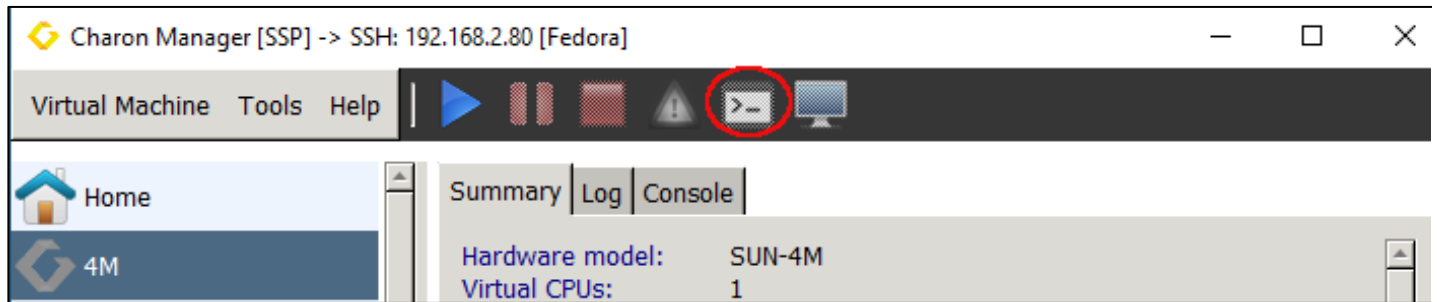


Figure 118: Charon-SSP Manager on Windows

Instead of the access to the built-in console, there is a terminal symbol in the upper bar that represents the **PuTTY** client. The console tab is only used to show cached console output while the instance is not running. Once you have started a virtual SPARC machine, you can click on the **PuTTY** symbol to access the SPARC console. The following image shows an example of console output in a **PuTTY** window.

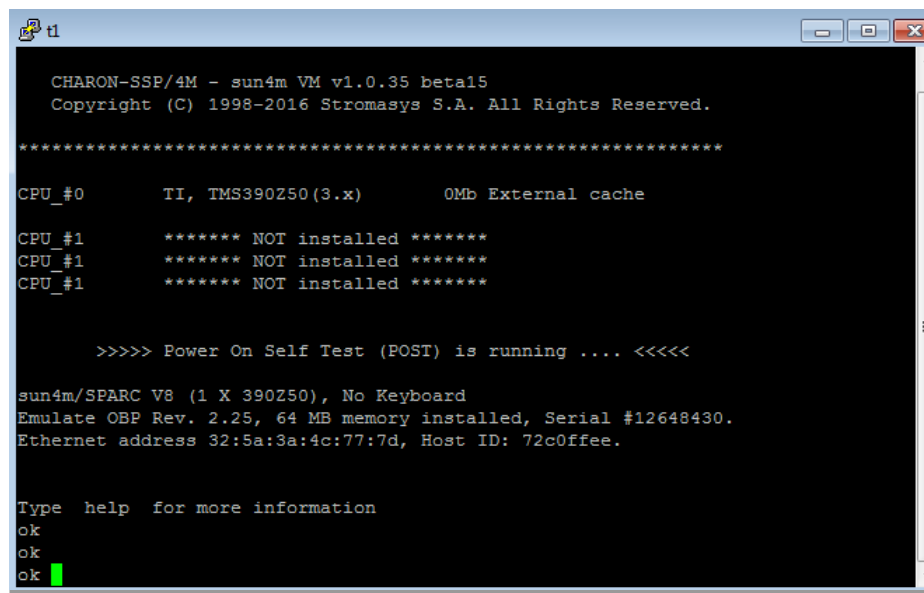


Figure 119: SPARC console access via PuTTY on Windows

The **Tools** menu contains an option to modify the **PuTTY Settings**.

A.1.4.2 Charon Manager on Windows – Graphical Console Access

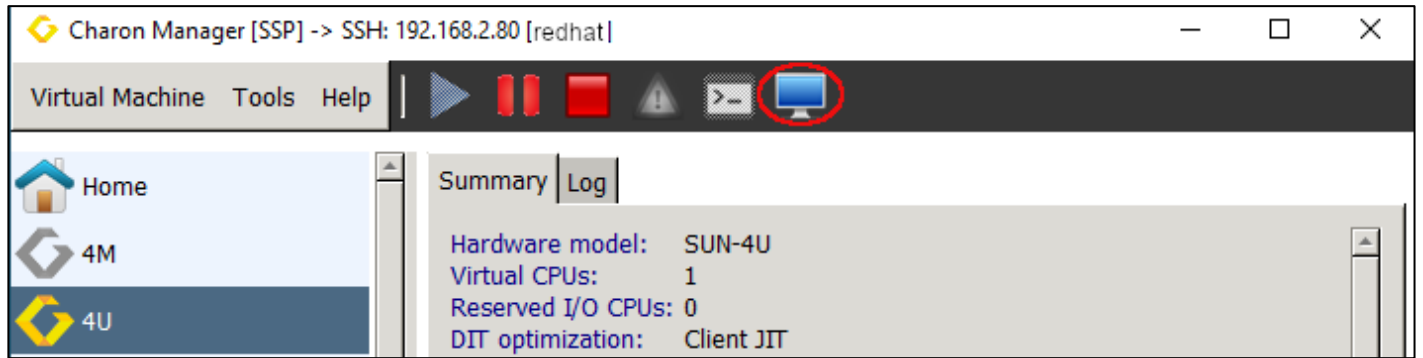


Figure 120: Graphical Console Symbol Charon-SSP Manager on Windows

The Charon-SSP Manager on Windows can also use a remote graphical console if the Charon instance has been configured accordingly. To do this, select a running Charon-SSP instance and click on the monitor symbol at the top of the Charon-SSP Manager: See [Console Access via the Emulated Graphics Device](#) for configuration details.

A.1.5 Upgrading the Charon-SSP GUI on Microsoft Windows

To upgrade the Charon-SSP Manager and the Charon-SSP Director for Windows, follow the installation instructions in the section [Charon-SSP GUI Installation on Windows](#).

Starting with Charon-SSP version 2, it is supported to have several versions of the Charon-SSP Manager on a system to manage remote systems with versions different from the local system. The Charon-SSP Manager installation preserves older versions. Later, you can use these versions to manage remote systems with matching versions.

Should the installer display an error message about an old version already being installed, the existing package must be removed before the new one can be installed. The process for removing the Charon-SSP Manager and the Charon-SSP Director for Windows is described in the section [Removing the Charon-SSP GUI on Microsoft Windows](#). Once removed, use the steps described in the installation sections to complete the upgrade of Charon-SSP Manager and Charon-SSP Director for Windows.

Please note: the automatic Charon Manager update is not supported in the current version of the Charon-SSP Manager for Microsoft Windows.

A.1.6 Removing the Charon-SSP GUI on Microsoft Windows

There are **two ways** to uninstall the Charon Manager and the Director:

1. The installation of the Charon-SSP Manager and the Charon-SSP Director also adds an **Uninstall** option in **Start > Stromasys**. To uninstall the software, open the menu and click on the uninstall option for either the Director or the Manager, as needed. You will be prompted for the Administrator password if you work from a normal user account.
2. Use the control panel:

To remove the **Charon-SSP Manager** software, follow the steps below:

| Step | Task |
|------|--|
| 1 | <ul style="list-style-type: none"> • Press WinKey+R to open the run input window. • Enter control into the window and press OK to open the control panel. |
| 2 | <ul style="list-style-type: none"> • Switch View by to Small Icons or Large Icons. • Click Programs and Features. |
| 3 | <ul style="list-style-type: none"> • Select Charon Manager from the list of installed software and right-click on it. • Click Uninstall. |

To remove the **Charon-SSP Director** software, follow the steps below:

| Step | Task |
|------|--|
| 1 | <ul style="list-style-type: none"> • Press WinKey+R to open the run input window. • Enter control into the window and press OK to open the control panel. |
| 2 | <ul style="list-style-type: none"> • Switch View by to Small Icons or Large Icons. • Click Programs and Features. |
| 3 | <ul style="list-style-type: none"> • Select Charon Director from the list of installed software and right-click on it. • Click Uninstall. |

B Appendix – Configuration File Reference

The Charon-SSP virtual machines can be configured using a text-based configuration file or the Charon-SSP Manager GUI. The configuration settings made via the GUI are stored in the configuration file.

This section describes the format and content of the text-based configuration file. The configuration file is made up of several sections that describe various aspects of the virtual machine environment. This appendix describes the different sections and their syntax.

The initial configuration files used by the Charon-SSP Manager are found in `/opt/charon-manager/ssp-manager/config/ssp/`.

B.1 Syntax

The configuration file format for Charon-SSP follows the “INI file” format originally used by MS-DOS and 16-bit Microsoft Windows. The syntax is described in EBNF (Extended Bakus Naur Format) below.

| Configuration file syntax in EBNF | |
|-----------------------------------|--------------------------------|
| file | := (section)+ |
| section | := '[' NAME ']' '\n' (key)+ |
| key | := NAME '=' VALUE '\n' |

The following briefly describes each syntax element used in the configuration.

B.1.1 Section

The configuration file is divided into sections. These sections are denoted by names enclosed in '[' and ']'. Example section header:

```
[system]
```

There is no “end of section” delimiter. A section ends when either the end of file is reached or a new section is encountered.

B.1.2 Properties

Configuration options (properties) are specified as key/value pairs. Key names are separated from values by the '=' character. All characters following the '=' sign up to the end of the line are considered part of the value. The following example shows a key/value pair.

```
port = /dev/ttyS0
```

Configuration property names are case-sensitive. This means that “port” and “Port” are not equivalent.

B.1.3 Comments

Comments are ignored by the emulator. They are provided for human readers and writers to leave informative notes. A comment starts with the '#' character and continues to the end of the line. The following example shows a comment line:

```
# Charon-SSP configuration file comment
```

B.1.4 Blank Lines

Blank lines are ignored by the emulator. They should be used to break up the configuration file, making it easier for a human reader to analyze.

B.2 Reference

The following sections describe how to use the configuration objects and their members to set up a virtual SPARC machine.

B.2.1 [cpu] Section

This section controls options related to the operation of the virtual machine's CPUs. The following list shows all available options:

- **dit** – Enable/disable dynamic instruction translation
- **dit_page_size** – Specify the dit page size
- **code_cache_size** – DIT code cache (Server JIT only)
- **fp_boost** – Floating point optimization
- **int_boost** – Integer optimization (Server JIT only)
- **number** – Number of virtual CPUs
- **ht** – Indication if the host system uses hyper-threading or runs in a VM
- **idle** – Host CPU behavior when the guest system is idle

B.2.1.1 dit

Enable/disable dynamic instruction translation.

Syntax

```
dit = on | off
```

Description

This option enables or disables the Dynamic Instruction Translation (DIT) mode. This can provide significant performance increase if enabled. Server JIT and Client JIT are implemented in two different images. The chosen image decides which mode is used when DIT has been enabled. **Server JIT is not available on Charon-SSP/4M.**

B.2.1.2 dit_page_size

Defines the page size for Dynamic Instruction Translation (Charon-SSP/4U(+)/4V(+) only).

Syntax

```
dit_page_size = size-in-KB
```

Description

This option defines the DIT page size in KB. It can have a maximum of 2048KB. This parameter should only be changed if the log file indicates that the DIT optimization was disabled because the translation buffer size was too small.

Valid options for *size-in-KB* are values between 480KB and 2048KB in 64KB increments. Exception: the last step to 2048KB only uses a 32KB increment.

B.2.1.3 code_page_size

Defines the size of the Server JIT code cache (Charon-SSP/4U(+)/4V(+) only).

Syntax

```
code_cache_size = size-in-MB
```

Description

This option defines the code page size for Server JIT in MB. It can have a maximum of 8192MB. The default value is 2048MB.

Valid options for *size-in-MB* are values between 1024MB and 8192MB in 1024MB increments.

B.2.1.4 fp_boost

Specifies if floating-point optimization is to be used, and at what level. Applies to client and server JIT. Not available in Charon-SSP/4M.

Syntax

```
fp_boost = boost-ratio
```

Description

Defines the level of floating-point optimization. The *boost-ratio* can be set to a value from 0 to 100. The default is 0 (= no boost). Most floating-point applications will profit from increasing this ratio. However, some applications may not be compatible with the optimization resulting in degraded performance. So testing is required.

B.2.1.5 int_boost

This parameter applies only to Server JIT. It is supported by the **ssp4u-jit/ssp4u-plus-jit** and **ssp4v-jit/ssp4v-plus-jit** images implementing the second level DIT optimization.

Syntax

```
int_boost = boost-ratio
```

Description

Defines the level of integer operation optimization. The *boost-ratio* can be set to a value from 0 to 100. The default is 100 (= maximum boost). The higher the value the more resources are required. Hence high values are likely to provide most benefit if the guest system applications run for a long time.

B.2.1.6 number

Specifies the number of virtual CPUs.

Syntax

```
number = cpu-count
```

Description

This option specifies the total number of CPUs the virtual machine is to provide.

The table below lists the supported virtual machine families and the maximum number of CPUs for each.

| Machine hardware family | Emulated model | Max. CPUs |
|-------------------------|---------------------|-----------|
| SUN-4M | Sun SPARCstation 20 | 4 |
| SUN-4U | Sun Enterprise 450 | 24 |
| SUN-4V | Sun SPARC T2 or T4 | 64 |

B.2.1.7 idle

Defines the host CPU behavior when guest Solaris is in idle state.

Syntax

```
idle = none | pause | sleep
```

Description

When guest Solaris is in idle state, the host CPU behavior is defined by the specific value assigned to idle attribute.

The possible values for the **idle** parameter, and the behavior they cause when the guest CPU is idle, are described below:

| Value | Description |
|-------|--|
| none | Corresponds to “Performance” in the Charon Manager setting. The host CPU keeps running Solaris idle handling instructions. |
| pause | Corresponds to “Balanced” in the Charon Manager setting. The host CPU turns to shallow pause mode. |
| sleep | Corresponds to Power save in the Charon Manager setting. The host CPU turns to deep sleep mode. If hyper-threading is enabled on the host system, the number of real/physical CPU cores is only half the number of vCPUs presented to the Linux operating system. In such environments, the hyper-threading option must be enabled. For best performance, the host system should have enough physical CPU cores to fully satisfy the emulator requirements. If this can be ensured , the power option can be set to Performance . In all other cases Power save must be used to allow optimal load distribution across the limited system resources. Since every customer environment is different, the above can only provide a general guideline and testing is always required to assess the overall performance of the system. |

B.2.1.8 ht

Indicates that the host system uses hyper-threading or runs on a VM, for example in VMware. Turn on if hyper-threading cannot be disabled on the host system.

Syntax

```
ht = on | off
```

Description

The Charon instance can adapt to a host environment with hyper-threading or a host running in a VM.

Values for the **ht** parameter:

| Value | Description |
|-------|--|
| off | Host system does not use hyper-threading and does not run in a VM. |
| on | <p>With this mode enabled, Charon-SSP does not use CPU core affinity on the host system, but relies on the scheduler of the host operating system. In combination with the power option Power save, this will allow rescheduling idle guest system CPU threads, thus making best use of the available, possibly limited, CPU capacity. For best performance, however, there should be enough CPU cores to avoid two active emulator threads sharing one CPU core.</p> <p>Recommended setting:</p> <ul style="list-style-type: none"> • The Charon-SSP hyper-threading option must be enabled if hyper-threading is enabled on the Charon host system. • If the Charon host system is running as a VM under a Hypervisor (e.g., VMware) or in a cloud instance, where hyper-threading is often active by default, and you are not sure whether the underlying server has hyper-threading enabled or not, there will be no detrimental effect if you enable the Charon-SSP hyper-threading option as a precaution. |

B.2.2 [ethernet] Section

This section describes the virtual Ethernet adapters attached to the virtual machine. The following list describes the available options:

- **interface** – Attach virtual Ethernet interface to host attached adapter.
- **mac** – Specifies virtual Ethernet adapter MAC address.
- **model** – Specifies which type of Ethernet adapter is presented to the guest system.

For virtual models that support the feature, it is possible to configure multiple Ethernet controllers. For an example configuration and details on section naming, see **[ethernet_n]** section.

B.2.2.1 interface

Attaches a virtual Ethernet interface to host attached adapter.

Syntax

```
interface = host-device
```

Description

The **interface** option is used to attach the virtual Ethernet adapter to a physical host adapter.

- **The interface must allow promiscuous mode unless the MAC address of the emulated interface is configured to be the same as the MAC address of the host NIC as described below (parameter mac).**
- It is permitted to assign the localhost interface (lo) to an emulated device (if the device will not be used by the guest).
- It is also permitted to add the same physical device to multiple emulated Ethernet devices of the **same instance**. However, this is not recommended for performance reasons.
- Sharing a NIC between emulator and host (not recommended for performance reasons) is possible but requires promiscuous mode and the MAC addresses of host and emulated system to be different (normally this is automatically taken care of by Charon-SSP toggling the locally-administered bit of the MAC address for the interface assigned to the guest system). It also requires mutual host routes via a default gateway for the Charon host and guest system if they should be able to communicate with one another on a shared NIC. Not an option in cloud environments.
- VMware and Solaris MAC addresses: VMware has several parameters to protect the environment from forged MAC addresses (e.g., the forged transmits and the address change parameters). If a MAC address is to be used

for a Charon instance that is different from the host NIC MAC address, these parameters must allow such a configuration.

- Assigning the same physical interface to more than one Charon-SSP instance is possible but **not supported for production operation**. It requires promiscuous mode and manual setting of unique MAC addresses for the Charon instances. I/O performance will be significantly degraded. For testing purposes only.

B.2.2.2 mac

Specify virtual Ethernet adapter MAC address.

Syntax

```
mac = aa:bb:cc:dd:ee:ff
```

Description

This optional parameter can be used to force the physical address of the network adapter to a certain address. This option can be useful in cases where licensing is tied to a network adapter MAC address. It can also be used to avoid having to set a VMware virtual network adapter to promiscuous mode, or in a cloud environment where promiscuous mode is generally not supported. **If this configuration is used, the emulator needs a dedicated NIC on the host system.**

B.2.2.3 model

Defines the Ethernet adapter model will be presented to the guest system. Available only on Charon-SSP/4U(+) and Charon-SSP/4V(+). Charon-SSP/4M always uses controller type **le**.

Syntax

```
model = controller-type
```

Description

Possible values for *controller-type*:

- SUN-4U: **hme**, **qfe**, and **bge** (BGE support was added in versions 5.2.5 and 5.3.4)
- SUN-4V: **bge** and **qfe**

Charon-SSP can emulate the QFE 4-port Ethernet adapter. This is indicated by setting the model to **qfe** in the configuration.

The selection of the controller type applies to all configured emulated Ethernet cards of the emulated system. Exception: on Charon-SSP/4U, the **first configured Ethernet interface** represents the SPARC on-board device and must be of type HME.

B.2.3 [ethernet_n] Section

This section is used for virtual machine configurations where there are multiple Ethernet controllers. The properties are the same as those described in the **[ethernet]** section. However, the section naming is slightly different in that the names are suffixed with an underscore and the controller number.

Valid controller numbers are

- 1 for Charon-SSP/4M
- 1 to 18 for Charon-SSP/4U(+)
- 1 to 7 for Charon-SSP/4V(+) (before version 5.3.4 only the values 1 to 3 are allowed)

The configuration example below demonstrates the configuration of two virtual Ethernet adapters.

| Sample Ethernet adapter configuration |
|---|
| <pre>[ethernet_1] interface = enp0s3 model = hme [ethernet_2] interface = enp0s8 mac = 08:00:2b:aa:bb:cc model = hme</pre> |

B.2.4 [log] Section

This section describes the configuration of the Charon-SSP for Linux virtual machine logging facility. The following list shows the properties supported by this section:

- **destination** – Logging facility output destination.
- **path** – Logging output path.
- **severity** – Logging severity level.
- **rotation** – Number of old versions of the log file to be kept

It is important to note that all properties in this section must be configured correctly to ensure that the virtual machine will start. The message **Failed to set up the log!** indicates that something in the logging section is not configured correctly or there are no write permissions to the log file.

Faults in the logging facility configuration could be:

- incorrect path or insufficient privilege to create the file,
- property or properties not configured, or
- misconfigured property value.

B.2.4.1 destination

Logging facility output destination.

Syntax

```
destination = log-destination
```

Description

This property controls the destination of the logging facility output.

The table below lists all possible values for *log-destination*.

| Destination | Description |
|-------------|--|
| all | Write output to all possible destinations. |
| console | Write output to the <code>/dev/console</code> device only. |
| file | Write output to the file specified by path only. |

B.2.4.2 path

Logging output file path.

Syntax

```
path = log-path
```

Description.

Specify a file path in *log-path*. The virtual machine log messages will be written to the specified file.

This configuration property must be present, even if **destination** is set to a value other than **all** or **file**.

B.2.4.3 severity

Logging severity level.

Syntax

```
severity = level
```

Description

Virtual machine logging messages are arranged into levels to make messages more relevant and reduce log file size. The severity property controls the level of the messages that are included in the logging output. The list below lists these levels from most to least verbose. Setting logging to a specific level also includes all levels below it. For example, setting severity to warning ensures that error and fatal are also included in the output and other levels are not.

Logging severity levels:

| Level | Description |
|----------------|--|
| debug | Debug and all lower-level messages are logged. |
| info | Informational and all lower-level messages are logged. |
| warning | Warning and all lower-level messages are logged. |
| error | Error and all lower-level messages are logged. |
| fatal | Only fatal error messages are logged. |

B.2.4.4 rotation

Number of old versions to be kept.

Syntax

```
rotation = number-of-versions
```

Description.

Specify the number of versions in the *number-of-versions*. The number can be between 1 and 20. The Charon-SSP logs are rotated upon startup of the emulator instance and, during operation, based on the number of lines written to the log. Once the number of log lines reaches 800.000, the log is rotated.

B.2.5 [nvram] Section

This section is used to configure the location of the NVRAM backing file store as well as some other NVRAM options. The list below describes the options that can be set:

- **disable_autoboot** – Autoboot of the guest system can be disabled.
- **hostid** – Set Sun Host ID.
- **path** – Specify location and name of console NVRAM file.

B.2.5.1 disable_autoboot

Disable autoboot of guest system even if this has been configured in the OpenBoot Console.

Syntax

```
disable_autoboot = 0 | 1
```

Description

If the value of this parameter is set to **1**, the guest system will not boot automatically when the Charon-SSP instance is started with autoboot enabled in the OpenBoot Console. If the value is set to **0**, the OpenBoot Console setting will be honored.

B.2.5.2 hostid

Set Sun Host ID.

Syntax

```
hostid = hex-hostid
```

Description

This optional property can be used to configure the Sun Host ID of the virtual machine. This may be useful in cases, where the software licenses are tied to the host ID of the physical SPARC system. The value of *hex-hostid* is of the format *0xxxxxxxx*.

B.2.5.3 path

Specify location and name of console NVRAM file.

Syntax

```
path = nvram-path
```

Description

This property specifies the location and name of the console NVRAM image. It contains information such as environment variable settings, boot flags, etc. Set *nvram-path* to filename (including the path) on the host system where this information can be stored.

B.2.6 [ram] Section

This section controls the virtual machine memory environment. The following list describes these options:

- **allocator** – Virtual machine memory allocator.
- **size** – Virtual machine memory size.

B.2.6.1 allocator

Virtual machine memory allocator.

Syntax

```
allocator = memory-allocator
```

Description

The **allocator** option is used to indicate to the virtual machine which memory allocation method the host system uses to allocate the virtual memory environment.

The *memory-allocator* value can be set according to the table below:

| Allocator | Description |
|---------------|---|
| malloc | All virtual machine RAM is allocated from system heap. This is the default and is appropriate for most cases. Please contact Stromasys if your environment has special memory requirements. |
| mmap | All virtual machine RAM is allocated from file-backed virtual memory via mmap. |

B.2.6.2 size

Virtual machine memory size.

Syntax

```
size = memory-size
```

Description

This option specifies the amount of host memory that the virtual machine should make available to the guest. It is specified in megabytes (MB).

The table below describes the allocation rules and maximum sizes for each virtual machine model.

| Virtual machine family | Memory allocation rules |
|---|--|
| SUN-4M (represented by the Sun SPARCstation 20) | 64MB, 128MB, 256MB, and 512MB |
| SUN-4U (represented by the Sun Enterprise 450) | 1 to 128GB in 1GB increments (expressed as MB: 1GB = 1024MB) |
| SUN-4V (represented by the Sun SPARC T2 and T4) | 1 to 1024GB in 1GB increments (expressed as MB: 1GB = 1024MB) Actual limits are different depending on guest OS: Solaris 10: 1TB, Solaris 11: 512 GB. |

If the memory size for a SUN-4U/4V is not correctly entered as a MB value corresponding to a multiple of 1024, the virtual machine does not start and prints the following error message in the log:

```
ERROR   VM      Memory size should be integral multiple of 1024MB
```

Please note that Solaris 11 requires a minimum amount of memory to boot (the real memory requirements usually are significantly higher):

- Solaris 11.1: 2GB
- Solaris 11.2: 2GB
- Solaris 11.3: 2GB
- Solaris 11.4: 4GB

B.2.7 [scsi_n] Section

This section describes the virtual SCSI storage device configuration. Each SCSI target ID used requires a separate section heading, where *n* is replaced with a SCSI ID number between 0 and 15 (0 to 6 for SUN-4M). The example below shows the configuration entries for a physical tape device attached at SCSI ID 5. All devices in this section are attached to the internal bus of the SPARC virtual machine.

| Physical tape device attached at SCSI ID 5 |
|---|
| <pre>[scsi_5] lun_0 = /dev/tape type = tape</pre> |

The following list describes the available options for [scsi_n] sections:

- **lun_X** – Virtual SCSI storage device path.
- **type** – Specify type of virtual SCSI storage device.
- **pass_through** – Turn SCSI pass-through on or off.
- **removable** – Defines if emulator can start with this device missing.

The parameters **path** and **serial_number** in earlier versions of the software have been superseded by the **lun_X** parameter.

For virtual models that support the feature, it is possible to configure additional virtual SCSI devices attached on an external SCSI controller. For an example configuration and details on section naming, see section [scsix_n].

Important additional information:

- Charon-SSP does not place any restrictions on the SCSI bus and target ID configured for emulated SCSI devices, e.g., a virtual CD-ROM. However,
 - Charon-SSP/4M normally expects the boot CD-ROM device to have SCSI ID 6 / LUN 0,
 - Charon-SSP/4U normally expects the boot CD-ROM device to be on the external bus and SCSI ID 6 / LUN 0, and
 - Charon-SSP/4V normally expects the boot CD-ROM device on the internal (primary) bus and SCSI ID 6 / LUN 0.

If you encounter the problem that the boot CD-ROM is not found when trying to boot from it, verify its expected location in the OBP environment (using the `devalias` command).

- The SCSI ID 7 is reserved for the virtual SCSI host bus adapter; consequently, it is not possible to configure a section titled [scsi_7].
- Charon-SSP/4M supports a maximum of 7 SCSI target IDs. If you add a SCSI device section outside this range, e.g., [scsi_8], they will be ignored and the disks will not be available to the virtual machine.
- Currently, the maximum size of a disk presented to a Charon-SSP emulator is 2TB.

B.2.7.1 lun_X

Physical or virtual device identification and LUN definition for virtual SCSI storage devices.

This parameter **supersedes** the following parameters of earlier versions:

- **serial_number** – if a physical disk is added by specifying its serial number, this number is stored in the lun_X parameter.
- **path** – the path to the physical device or storage container file is now stored in the lun_X parameter.

The old parameters are understood by the new version. The Charon-SSP Manager will convert the parameters when the configuration is first saved via the Charon-SSP Manager. After this, manual changes will be necessary to return to the old version should this be required.

Syntax

```
lun_X = [virtual device container file path |
        physical host-device |
        physical disk serial number |
        path to generic scsi device]
```

Description

The **lun_X** option is used to define the LUN of the virtual SCSI device and to attach the virtual SCSI device to

- a virtual storage container file on the host (*virtual device container file path*),
- a physical host storage device (*path to physical host device* or *physical disk serial number*), or
- a generic SCSI device on the host (*path to generic scsi device*).

One [scsi_n] section can have several lun_X definitions. X can have a value from 0 through 7 and must be unique in one scsi_n section. The values specified for X must be contiguous and start from 0. The exact number of possible LUNs depends on the emulated hardware, the guest operating system and driver versions, and the SCSI devices used.

Persistent device naming for physical disks:

Linux device names in the form of /dev/sdX are not guaranteed to be persistent across Linux reboots.

Hence, for physical disks, it is strongly recommended to use a **persistent device name** from

- /dev/disk/by-id, or
- /dev/disk/by-uuid

instead of a non-persistent /dev/sdX device name.

Alternatively, the physical disk serial number can be used (device type must be **disk**). It can be determined using the following command:

```
# udevadm info -q property -n /dev/diskname | grep SERIAL
```

Either ID_SERIAL_SHORT or ID SCSI_SERIAL can be used.

All lun_X entries in one [scsi_n] section must be for virtual SCSI devices of the same type. Support for multiple LUNs also depends on the capabilities of the emulated system and the Solaris guest system.

B.2.7.2 type

Specify type of virtual SCSI storage device.

Syntax

```
type = device-type
```

Description

This property describes the type of the virtual SCSI storage device. It is possible to attach a range of devices in various formats, including container files and physical devices.

The table below describes each possible value for *device-type*.

| Device type | Description | Example path |
|---|--|--|
| iso | Virtual CD-ROM ISO container file. | /usr/local/share/iso/suns-4.1.4.iso |
| vdisk | Virtual disk container file. | /usr/local/vm/bender/disk0.vdisk |
| vtape | Virtual tape container file. | /usr/local/vm/leela/tape0.vtape |
| disk | Physical disk device or physical disk partition. | /dev/sda /dev/disk/by-uuid/586c8447-3e83-4f93-8581-b03c2110762e |
| The following devices are only supported on conventional and Baremetal on-premises installations: | | |
| cdrom | Physical CD-ROM | /dev/cdrom |
| tape | Physical tape device. | /dev/tape |
| generic | Generic SCSI device | /dev/sg0 |

B.2.7.3 pass_through

Enable or disable SCSI pass-through for physical disks.

Syntax

```
pass_through = on | off
```

Description

This option allows you to turn SCSI pass-through on or off. Turning it **on** allows direct access to SCSI devices to the virtual system. This feature is useful, for example, for using shared disks in cluster environments (fencing / persistent reservations). Also, with SCSI pass-through, the data bypasses the Linux I/O cache layer. Read/write operations go directly to the storage controller.

B.2.7.4 removable

Defines if emulator can start with this device missing.

Syntax

```
removable = on | off
```

Description

If the parameter is set to **on**, the emulator can start even when the device is not present on the host system. If the parameter is set to **off**, Charon-SSP will log an error and stop if the device is missing from the host.

B.2.8 [scsix_n] Section

These sections are used for virtual machine models that support the external SCSI bus (hardware family SUN-4U and SUN-4V). Each SCSI target ID used requires a separate section heading, where *n* is replaced with a SCSI ID number between 0 and 15.

The example below shows a configuration of three devices (one CD-ROM backed by an ISO container, one generic SCSI device, and two physical disks on one SCSI target ID) attached to the external SCSI bus.

| Example SCSI device configuration for the emulated external SCSI bus |
|---|
| <pre>[scsix_3] type = generic lun_3 = /dev/sg0 removable = on [scsix_5] type = disk lun_0 = S21JNXAGA00047N lun_1 = S51JNXAGA00047N removable = off [scsix_6] type = iso lun_0 = /usr/local/share/iso/solaris.1.1.2.iso removable = off</pre> |

The section properties described in the section **[scsi_n]** are also applicable here. Please refer to the **[scsi_n]** section for more information.

Important additional information:

- Charon-SSP does not place any restrictions on the SCSI bus and target ID configured for emulated SCSI devices, e.g., a virtual CD-ROM. However,
 - Charon-SSP/4M normally expects the boot CD-ROM device to have SCSI ID 6 / LUN 0,
 - Charon-SSP/4U normally expects the boot CD-ROM device to be on the external bus and SCSI ID 6 / LUN 0, and
 - Charon-SSP/4V normally expects the boot CD-ROM device on the internal (primary) bus and SCSI ID 6 / LUN 0.

If you encounter the problem that the boot CD-ROM is not found when trying to boot from it, verify its expected location in the OBP environment (using the **devalias** command).

- The SCSI ID 7 is reserved for the virtual SCSI host bus adapter; consequently, it is not possible to configure a section titled `[scsix_7]`.

B.2.9 [vds_*] sections

This feature is available on Charon-SSP/4V only. The vds sections are used to define VDS storage devices.

There is a separate section for each storage device added (*N* denotes the configured unique VDS ID):

- **[vds_disk*N*]** – one for each physical or virtual disk added
- **[vds_cdrom*N*]** – one for each physical or virtual CD-ROM added (physical CD-ROM is not available in cloud environment)

Each **vds** section has one or more parameters depending on the device type used:

- **path** – path to a virtual disk container, iso file, physical disk or physical CD-ROM device, or the serial number of a physical disk
- **pass_through** – turn SCSI pass-through on or off
- **write_through** – turn write through mode on or off

B.2.9.1 path

Physical or virtual device identification of a VDS storage device.

Syntax

```
path = [virtual device container file path |
        physical host-device |
        physical disk serial number]
```

Description

The **path** option is used to attach the VDS device to

- a virtual storage container or ISO file on the host (*virtual device container file path*), or
- a physical host storage device (*path to physical host device* or *physical disk serial number*)

Physical CD-ROM devices are not supported in cloud environments.

Persistent device naming for physical disks:

Linux device names in the form of `/dev/sdX` are not guaranteed to be persistent across Linux reboots.

Hence, for physical disks, it is strongly recommended to use a **persistent device name** from

- `/dev/disk/by-id`, or
- `/dev/disk/by-uuid`

instead of a non-persistent `/dev/sdX` device name.

Alternatively, the physical disk serial number can be used. It can be determined using the following command:

```
# udevadm info -q property -n /dev/diskname | grep SERIAL
```

Either `ID_SERIAL_SHORT` or `ID SCSI_SERIAL` can be used.

B.2.9.2 pass_through

Enable or disable SCSI pass-through for physical disks.

Syntax

```
pass_through = on | off
```

Description

This option allows you to turn SCSI pass-through on or off. Turning it **on** allows direct access to VDS physical devices for the guest system. This feature is useful, for example, for using shared disks in cluster environments (fencing / persistent

reservations). Also, with SCSI pass-through, the data bypasses the Linux I/O cache layer. Read/write operations go directly to the storage controller.

B.2.9.3 write_through

Enable or disable VDS write-through mode.

Syntax

```
write_through = on | off
```

Description

This option allows you to turn VDS **write-through** mode on or off for virtual and physical disks. If write-through mode is disabled (value = off), the disk operates in write-back mode. This means that it will confirm a successful write operation once the data has been written to cache. If write-through mode is enabled (value = on), a successful write operation will only be confirmed once the data has been written to the Linux storage.

B.2.10 [chassis] Section

This feature is available on Charon-SSP/4V only. It can be used to adapt the chassis identification parameters to the ones of the original system if the guest system requires it. The following options can be set in this section:

- **part_number**
- **serial**
- **manufacturer**
- **product**
- **firmware**

Syntax

```
parameter = value
```

Where *parameter* is one of the parameters listed above and the *value* must be obtained from the original system.

B.2.11 [floppy] Section

The section is used to configure a virtual floppy device. This feature is available on Charon-SSP/4M and Charon-SSP/4U only (for Charon-SSP/4U starting with 5.3.1). The following options can be set in this section:

- **type** – Specify whether a container file or a physical floppy is to be used.
- **path** - Path to container file or physical device.

B.2.11.1 type

Specify whether a container file or a physical floppy is to be used.

Syntax

```
type = virtual | physical
```

Description

Set parameter to **virtual** if a container file is to be used. Set it to **physical** if a real device on the host system is to be used.

B.2.11.2 path

Path to container file or physical device.

Syntax

```
path = path-to-container | physical-device
```

Description

Set the parameter to the path of the container file to be used for the virtual floppy (e.g., `/my/virtual/devices/flptest.flp`). Set the parameter to the physical device if a real device on the host system is to be used (e.g., `/dev/fd0`).

B.2.12 [system] Section

The system section is used to configure "system-wide" properties of the virtual machine. The list below describes the options that can be set:

- **cpu_affinity** – Assigns virtual machine CPU processing to a specific host CPU.
- **io_affinity** – Assigns virtual machine I/O processing to a specific host CPU.
- **io_cpus** – Number of host CPUs reserved for virtual machine I/O processing.
- **machine** – Specifies the SPARC system model of the virtual machine.
- **model** – Specifies the model of a SUN-4V machine.

B.2.12.1 cpu_affinity

Assign virtual CPU processing to a specific host CPU.

Syntax

```
cpu_affinity = cpu-affinity [, cpu-affinity [, ... ]]
```

Description

This option is a comma-separated list of host CPU cores that the virtual machine assigns to virtual CPU threads. The CPU ID index starts from 0. If this option is used, there must be exactly one host CPU in the list for each virtual SPARC CPU. The virtual machine will assign affinity automatically (starting from the highest CPU ID) if this option is not set (recommended). **Cannot be used with hyper-threading mode enabled.** CPU cores assigned to emulated CPUs are never shared between instances.

B.2.12.2 io_affinity

Assigns virtual machine I/O processing to a specific host CPU.

Syntax

```
io_affinity = io-affinity [, io-affinity [, ... ]]
```

Description

This option accepts a comma-separated list of specific host CPUs (depending on the hyper-threading configuration, the CPUs visible to the host OS can be CPU cores or logical CPUs) which the virtual machine assigns to I/O processing. If this option is not set, the virtual machine will assign affinity automatically (starting from the lowest CPU ID). Automatic assignment is recommended. CPUs reserved for I/O here cannot be shared with other instances.

B.2.12.3 io_cpus

Number of host CPUs reserved for virtual machine I/O processing.

Syntax

```
io_cpus = cpu-count
```

Description

Use this option to reserve a specific number of host CPUs for virtual machine I/O processing. If neither **io_affinity** nor **io_cpus** are set, the virtual machine will automatically reserve one third of the CPUs visible to the Charon host for I/O processing (rounded down, minimum one). Allocation starts from the lowest CPU ID. Depending on the hyper-threading configuration, the CPUs visible to the host OS can be CPU cores or logical CPUs. If this definition overlaps with reservations of CPUs for I/O processing on other instances (automatic or via this parameter), the overlapping CPUs will be shared between instances.

B.2.12.4 machine

Specify the SPARC hardware-family of the virtual machine.

Syntax

```
machine = machine-name
```

Description

The machine keyword is used to indicate the specific hardware family of the SPARC-based systems to be emulated.

The following table lists the possible values of *machine-name* and the systems they represent.

| Machine name (configured hardware family) | Hardware represented |
|---|--------------------------|
| SUN-4U | Sun Enterprise 450 |
| SUN-4M | Sun SPARCstation 20 |
| SUN-4V | Sun SPARC T2 or SPARC T4 |
| The <i>machine-name</i> value must be specified exactly as specified above (including capitalized letters). The specified model must be covered by the Charon-SSP license for the emulator instance to run successfully. | |

B.2.12.5 model

Specify the SPARC hardware model of the virtual machine. **Currently available for SUN-4V only.**

Syntax

```
model = T2 | T4
```

Description

The model keyword is used to specify the SUN-4V model emulated by Charon-SSP. If the guest system is Solaris 11.4, the model must be SPARC T4. Default: T2.

B.2.13 [vconsole] Section (Charon-SSP/4V only)

The ttya section is used to configure the first serial port (often used as the console) on the SPARC virtual machine. Using the associated options, it is possible to attach the virtual serial port to a network socket or a physical serial port attached to the host. The list below describes the options that can be configured:

- **port** – Specify to which device the virtual serial port is connected.
- **start_console** – Start PuTTY at virtual machine boot.
- **restrict_access** – Allow access from remote hosts or only from the local host
- **type** – Virtual serial port type.
- **log_path** – Path of the console log.

B.2.13.1 port

Specification of the device for the virtual serial port.

Syntax

```
port = port-spec
```

Description

This option is dependent on the setting of the **type** option.

The table below describes the valid settings for *port-spec* in relation to the port type listed in the first column.

| If port type is... | Values for <i>port-spec</i> parameter | Description |
|--------------------|---|---|
| physical | Path to physical device (e.g., /dev/ttyS0) | The virtual serial port is attached to a physical serial port attached to the host. |
| socket | TCP/IP socket number (e.g., 9000) | The virtual serial port is attached to a network port to which terminal applications can connect. Raw serial connection. The port number must be unique on the host system. |
| telnet | TCP/IP socket number (e.g., 9000) | The virtual serial port is attached to a network port to which terminal applications can connect. The port supports the telnet protocol. The port number must be unique on the host system. |
| terminal | Local Linux host terminal is used (port number is optional) | The virtual port uses the current Linux terminal for virtual SPARC input & output. If a unique TCP port is specified for this type, the console is also reachable via the network. This should be reserved for exceptional, well-defined cases. |

B.2.13.2 start_console

Start PuTTY at virtual machine boot depending on the serial console configuration in Charon Manager. The option can only be changed manually by editing the configuration file

Syntax

```
start_console = off | on
```

Description

This option is **enabled** automatically if the console is defined to be **external** in the Charon-SSP Manager. It is **disabled** automatically if the console is defined as **built-in**. It enables (or disables) the automatic starting of PuTTY when the virtual machine starts.

If you change the option manually and start the Charon Manager, it will adapt the console definition automatically to the value set in the configuration file (on = external console, off = built-in console). To enable the combination of external console and PuTTY start disabled, the emulator must be run from the command line.

B.2.13.3 restrict_access

Allow access from remote hosts or only from the local host.

Syntax

```
restrict_access = on | off
```

Description

If set to **on**, access to the console on the Vconsole is only possible from the local system. Otherwise, when the Vconsole port is defined as a network port (TCP socket), access from remote systems is possible.

B.2.13.4 type

Port type of the virtual serial device.

Syntax

```
type = port-type
```

Description

This option configures how the serial console port is connected.

The table below lists the possible values for *port-type* and their purpose.

| Port type | Description |
|-----------|---|
| physical | The virtual serial port is connected to a physical, host-attached serial port. |
| socket | The virtual serial port is to be connected to a network socket. Raw serial connection. |
| telnet | The virtual serial port is to be connected to a network socket and supports telnet. |
| terminal | The virtual port uses the current Linux terminal for virtual SPARC input & output. If a unique TCP port is specified for this type, the console is also reachable via the network. This should be reserved for exceptional, well-defined cases. |

For details of the port specifications available for the selected port type, see the **port** section.

B.2.13.5 log_path

Enable console log and define path of log file.

Syntax

```
log_path = file-path
```

Description

This option enables the collection of console log information in the file designated by *file-path*.

B.2.14 [ttya] Section

The ttya section is used to configure the first serial port on the SPARC virtual machine (often used as the console on Charon-SSP/4M/4U emulated systems). On Charon-SSP/4V, this port can only be used as a normal serial line. Using the associated options, it is possible to attach the virtual serial port to a network socket or a physical serial port attached to the host. The list below describes the options that can be configured:

- **port** – Specify to which device the virtual serial port is connected.
- **restrict_access** – Allow access from remote hosts or only from the local host.
- **start_console** – Start PuTTY at virtual machine boot.
- **type** – Virtual serial port type.
- **log_path** – Path of the console log.

B.2.14.1 port

Specification of the device for the virtual serial port.

Syntax

```
port = port-spec
```

Description

This option is dependent on the setting of the **type** option.

The table below describes the valid settings for *port-spec* in relation to the port type listed in the first column.

| If port type is... | Values for <i>port-spec</i> parameter | Description |
|--------------------|---|---|
| physical | Path to physical device, e.g., /dev/ttyS0 | The virtual serial port is attached to a physical serial port attached to the host. |
| socket | TCP/IP socket number, e.g., 9000 | The virtual serial port is attached to a network port to which terminal applications can connect. Raw serial connection. The port number must be unique on the host system. |
| telnet | TCP/IP socket number, e.g., 9000 | The virtual serial port is attached to a network port to which terminal applications can connect. The port supports the telnet protocol. The port number must be unique on the host system. |
| terminal | Local Linux host terminal is used (port number is optional) | The virtual port uses the current Linux terminal for virtual SPARC input & output. If a unique TCP port is specified for this type, the console is also reachable via the network. This should be reserved for exceptional, well-defined cases. |

B.2.14.2 restrict_access

Allow access from remote hosts or only from the local host. **Not available on Charon-SSP/4V.**

Syntax

```
restrict_access = on | off
```

Description

If set to **on**, access to the console on TTYA is only possible from the local system. Otherwise, when TTYA is defined as a network port (TCP socket), access from remote systems is possible.

B.2.14.3 start_console

Start PuTTY at virtual machine boot depending on the serial console configuration in Charon Manager. The option can only be changed manually by editing the configuration file

Syntax

```
start_console = off | on
```

Description

This option is **enabled** automatically if the console is defined to be **external** in the Charon-SSP Manager. It is **disabled** automatically if the console is defined as **built-in**. It enables (or disables) the automatic starting of PuTTY when the virtual machine starts.

If you change the option manually and start the Charon Manager, it will adapt the console definition automatically to the value set in the configuration file (on = external console, off = built-in console). To enable the combination of external console and PuTTY start disabled, the emulator must be run from the command line.

B.2.14.4 type

Port type of the virtual serial device.

Syntax

```
type = port-type
```

Description

This option configures how the serial console port is connected.

The table below lists the possible values for *port-type* and their purpose.

| Port type | Description |
|-----------|---|
| physical | The virtual serial port is connected to a physical, host-attached serial port. |
| socket | The virtual serial port is to be connected to a network socket. Raw serial connection. |
| telnet | The virtual serial port is to be connected to a network socket and supports telnet. |
| terminal | The virtual port uses the current Linux terminal for virtual SPARC input & output. If a unique TCP port is specified for this type, the console is also reachable via the network. This should be reserved for exceptional, well-defined cases. |

For details of the port specifications available for the selected port type, see the **port** section.

B.2.14.5 log_path

Enable console log and define path of log file. **Not applicable to Charon-SSP/4V.**

Syntax

```
log_path = file-path
```

Description

This option enables the collection of console log information in the file designated by *file-path*.

B.2.14.6 Examples

The following configuration extract demonstrates how to attach the virtual serial port to the host device `/dev/ttyS0`.

| Configuring ttya for physical console access |
|--|
| <pre># Virtual serial console attached to host device /dev/ttyS0. [ttya] type = physical port = /dev/ttyS0</pre> |

The configuration extract below shows the parameters for **ttya** to accept incoming telnet connections on port 9000/tcp on the host. Note that the option to restrict access to the local host is disabled.

| Configuring ttya for network console access |
|---|
| <pre># Serial console redirected to network port 9000/tcp using the telnet protocol. [ttya] type = telnet port = 9000 restrict_access = off</pre> |

B.2.15 [ttyb] Section

The ttyb section is used to configure the second serial port on the SPARC virtual machine. Using the available options, it is possible to attach the virtual serial port to a network socket or a physical serial port attached to the host. The configuration options are the same as for **ttya** described in preceding section.

B.2.16 [ttyx] Section

The ttyx section is used to configure additional serial ports on the SPARC virtual machine.

For **Charon-SSP/4U(+)**, this is accomplished either as additional emulated ports (max. 14 ports named ttyc, ttyd, etc.) or as an emulated DIGI board (max. 32 ports named tty_1, tty_2, etc.). The latter requires a special driver on the Solaris guest system.

Charon-SSP/4V(+) supports additional emulated ports (max. 14 ports named ttyc, ttyd, etc.), and for **Charon-SSP/4M**, 8 serial ports named tty_1 through tty_8 can be configured.

The new serial devices on Solaris are located under **/dev/term** and **/dev/cua** for emulated on-board devices and the 4M STC devices. They are located under **/dev/dty** for the emulated DIGI board devices.

In Charon-SSP/4V, TTYA and TTYB must be enabled if additional ports are configured in TTYX. Otherwise, the additional devices will not be created in /dev/term.

It is possible to attach the virtual serial port to a network socket or a physical serial port of the host system using the available configuration options. Please refer to the description of the **port** and **type** parameters in the **ttya** section for further information (port type terminal, port type telnet, restrict_access, and terminal log **are not supported in the ttyx section**).

B.2.17 [digi_ppt_n] Section

This section defines a serial DIGI board in PCI pass-through mode. Available on Charon-SSP/4U(+) only. The number **n** defines the sequence number of the pass-through device definitions. The section has one configurable parameter:

- **path** – Specifies the special file representing the PCI device.

B.2.17.1 path

Specifies the special file representing the PCI device.

Syntax

```
path = special-file
```

Description

The PPT (PCI pass-through) driver delivered as part of the Charon-SSP kit provides **/dev/kdigi*** devices. Such devices can be defined in the path parameter to be handed to the guest operating system as a PCI pass-through device. Currently, Digi AccelePort 920 and C/X cards are supported. The appropriate driver is required on Solaris.

B.2.18 [gpib_n] Section

This section defines a GPIB board in PCI pass-through mode. Available on Charon-SSP/4U(+) only. The number **n** defines the sequence number of the pass-through device definitions. The section has one configurable parameter:

- **path** – Specifies the special file representing the PCI device.

B.2.18.1 path

Specifies the special file representing the PCI device.

Syntax

```
path = special-file
```

Description

The PPT (PCI pass-through) driver delivered as part of the Charon-SSP kit provides **/dev/kni*** devices. Such devices can be defined in the path parameter to be handed to the guest operating system as a PCI pass-through device. The appropriate driver is required on Solaris.

B.2.19 [parallel] Section

This section allows the definition of one parallel port for Charon-SSP/4M instances only. It has one configurable option:

- **printer**

B.2.19.1 Printer

Defines the name of the parallel port or file on the host system that should be mapped to the parallel port of the Charon-SSP instance.

Syntax

```
printer = parallel-device-name
```

Description

The parameter *parallel-device-name* contains the device or file name on the host system that is mapped to the parallel port of the guest system.

B.2.20 [license] Section

The license section offers parameters to define which licenses or license servers the emulator is to use. The list below describes the options that can be configured:

- **regular_key_id** – License ID of the primary HASP key
- **backup_key_id** – License ID of the backup HASP key
- **server** – IP address of the VE license server
- **backup_server** – IP address of the VE backup license server
- **server_key** – passphrase for authentication with primary VE license server
- **backup_server_key** – passphrase for authentication with backup VE license server

B.2.20.1 regular_key_id

Specifies the license ID of the primary Sentinel HASP key.

Syntax

```
regular_key_id = hasp-license-id
```

Description

The numerical license ID of the primary HASP key. Use the **hasp_srm_view** command to identify the ID.

B.2.20.2 backup_key_id

Specifies the license ID of the backup Sentinel HASP key.

Syntax

```
backup_key_id = hasp-license-id
```

Description

The numerical license ID of the backup HASP key. Use the **hasp_srm_view** command to identify the ID.

B.2.20.3 server

Specifies the IP address and (if non-default) the TCP port of a VE license server.

Syntax

```
server = ve-license-server-ip-address[:port]
```

Description

The IP address and optionally the TCP port number of a VE license server that is configured for use by the system. The default TCP port number is 8083.

B.2.20.4 backup_server

Specifies the IP address and (if non-default) the TCP port of a VE backup license server.

Syntax

```
backup_server = ve-backup-license-server-ip-address[:port]
```

Description

The IP address and optionally the TCP port number of a VE backup license server that is configured for use by the system. The default TCP port number is 8083.

B.2.20.5 server_key and backup_server_key

The parameters **server_key** and **backup_server_key** specify the passphrase(s) to be used for authentication with the primary and/or backup VE license server. The passphrase for a license is defined when the license is created by Stromasys. Use the command **/opt/license-server/license_viewer** on the VE primary and backup license server to identify the correct passphrase. If the license contains more than one product section, use the correct passphrase for the product section the emulator will use. **The use of a passphrase requires emulator versions 4.3.x and higher, and license server versions 1.1.x and higher.**

Syntax

```
[server_key | backup_server_key] = passphrase
```

Description

The parameter passphrase must be set to the passphrase defined on the license served by the VE license server to the VE license client.

B.2.21 [graphics] Section

The graphics section defines the parameters for the emulated graphics device. **Not applicable to Charon-SSP/4V.** The following parameters can be configured:

- **type** – type of graphics device
- **dual_display** – whether one or two screens should be used
- **remote_display** – whether the display should be local or remote
- **display1** – DISPLAY variable for screen 1
- **display2** – DISPLAY variable for screen 2
- **remote_port1** – port on which the display is served to remote clients (screen 1)
- **remote_port2** – port on which the display is served to remote clients (screen 2)
- **console** – whether the graphical device should be the console of the guest system
- **mouse_port** – port to transmit mouse events
- **keyboard_port** – port to transmit keyboard events
- **keyboard_layout** – country specific keyboard setting
- **resolution** – resolution of the emulated graphical display
- **full_screen** – **enable** full-screen mode for the emulated device
- **refresh_rate** – refresh rate of the emulated graphical display

B.2.21.1 type

Type of graphics display to be emulated.

Syntax

```
type = graphics-type
```

Description

This option enables the emulation of a graphical device or disables it.

Possible settings of *graphics-type*:

| Graphics type | Description |
|---------------|---|
| disabled | Graphical device emulation is disabled. |
| cgsix | Enables the emulation of graphics card CGSIX. CGSIX emulation is not supported for SunOS 4.x guest systems. |
| cgthree | Enables the emulation of graphics card CGTHREE (Sun-4m only). |
| ragexl | Enables the emulation of graphics card Rage XL (Sun-4u only). |

The configuration creates `/dev/fb*` device links pointing to the actual devices. Driver names: *cgsix*, *cgthree*, and *m64*.

B.2.21.2 dual_display

Optionally enables the dual-screen configuration.

Syntax

```
dual_display = on | off
```

Description

This option determines if one or two screens will be used. If enabled, it will also activate the settings for **display2** and **remote_port2** (if **remote_display** is enabled).

B.2.21.3 remote_display

Optionally enables the remote graphics display (starting with version SSP 5.3.9 for Charon Manager on Linux, and with SSP 5.3.11 for Charon Manager on Windows). This option requires that the Charon Manager be installed on the remote system.

Syntax

```
remote_display = on | off
```

Description

This option determines if the graphical display is local to the emulator host or if it can also be displayed on a remote host. If enabled, it will also activate the settings for **remote_port1** and **remote_port2** (if **dual_display** is enabled). For cloud-based Charon installations, remote display is the only option offered by the Charon Manager GUI.

B.2.21.4 display1 and display2

The X display of the host that is to be used for the emulated graphical device.

Syntax

```
display1 = display-variable1
display2 = display-variable2
```

Description

These options define the display(s) of the Linux host that should be used. Format: *:display.screen*, for example *:1.0* to select display 1, screen 0.

The *display-variable* value(s) must be set to match the display configuration of the system on which the graphics output is to be displayed. Use the command **echo \$DISPLAY** on the target Linux system to identify the correct value.

The option **display2** is only active when **dual_display** is enabled.

B.2.21.5 remote_port1 and remote_port2

The port on the host that is to be used for making the graphical display available to remote systems.

Syntax

```
remote_port1 = port-number1
remote_port2 = port_number2
```

Description

These options define the port number(s) on the Linux host that should be used to make the display available to remote systems. The port number settings are only relevant if **remote_display** is enabled.

The option **remote_port2** is only active when **dual_display** is also enabled. Default ports: 11100 and 11101. The ports must be unique on the host system.

B.2.21.6 console

Enables or disables the Solaris console on the emulated graphics display.

Syntax

```
console = on | off
```

Description

This option determines if the graphical display is also the console of the emulated system. If enabled, the console tab in the Charon-SSP Manager will be unavailable.

B.2.21.7 mouse_port

The port on the host that is to be used for transmitting mouse events.

Syntax

```
mouse_port = port-number
```

Description

This option defines the port number on the Linux host that should be used to transmit mouse events. Default port: 11001. The port must be unique on the host system.

B.2.21.8 keyboard_port

The port on the host that is to be used for transmitting keyboard events.

Syntax

```
keyboard_port = port-number
```

Description

This option defines the port number on the Linux host that should be used to transmit keyboard events. Default port: 11000. The port must be unique on the host system.

B.2.21.9 keyboard_layout

Defines the keyboard layout to be used for the emulated graphics device.

Syntax

```
keyboard_layout = layout-number
```

Description

This option sets the keyboard layout to be used.

The following table lists the values for *layout-number*.

| Value | Layout | Value | Layout |
|-------|----------------|-------|----------------|
| 0 | US | 11 | SWEDEN_FINLAND |
| 2 | FRENCH_BELGIUM | 12 | SWISS_FRENCH |
| 3 | CANADA_FRENCH | 13 | SWISS_GERMAN |
| 4 | DENMARK | 14 | UK |
| 5 | GERMANY | 32 | JAPAN |
| 6 | ITALY | | |
| 7 | NETHERLANDS | | |
| 8 | NORWAY | | |
| 9 | PORTUGAL | | |
| 10 | SPAIN_LATIN | | |

B.2.21.10 resolution

Defines the resolution of the emulated graphics device.

Syntax

```
resolution = screen-resolution
```

Description

This option defines the resolution to be used.

| Value | Comment |
|-----------|------------------------|
| 800X600 | GCTHREE only |
| 1024X768 | All |
| 1152X900 | All |
| 1280X1024 | CGSIX and Rage XL only |
| 1600X1280 | CGSIX and Rage XL only |

B.2.21.11 full_screen

Allows starting the emulated graphics device in full-screen mode.

Syntax

```
full_screen = on | off
```

Description

This option enables (**on**) or disables (**off**) the start in full-screen mode for the emulated device. Best results are achieved if the host display resolution matches the resolution of the emulated device. The mode can be toggled during operation using the key combination **CTRL+SHIFT+F** while focus is in the graphics window.

B.2.21.12 refresh_rate

The refresh rate of the emulated graphics device. This parameter is only available for Sun-4U(+).

Syntax

```
refresh_rate = rate
```

Description

This option defines the refresh rate of the emulated graphics device. The rate can be set to a value between 20 and 100.

B.2.22 [audio] Section

This section is not applicable to Charon-SSP/4V. The audio section has two parameters:

- enable
- server

B.2.22.1 enable

Turns the audio function on or off.

Syntax

```
enable = on | off
```

Description

Set the parameter to **off** to disable the option and to **on** to enable the option.

B.2.22.2 server

Defines the audio server

Syntax

```
server = ip-address
```

Description

Set the IP-address of the audio server. The default is the local host system.

B.2.23 [usb] Section

This option enables or disables the emulation of a USB port on the guest system. **Not available on Charon-SSP/4M.**

The usb section has only one parameter:

- **enabled**

B.2.23.1 enable

Turns the USB function on or off.

Syntax

```
enabled = on | off
```

Description

Set the parameter to **off** to disable the option and to **on** to enable the option. Currently, this option can be used to connect USB storage to an emulated system.

B.2.24 [obp] Section

This section allows the configuration of an OpenBoot PROM image if the emulator itself does not include OBP support.

The option is only usable in special testing/debugging cases following the instructions of Stromasys support. It is not used under normal circumstances.

The section has only one parameter:

- **path**

B.2.24.1 path

Specifies the path to an OpenBoot PROM image.

Syntax

```
path = path-to-obp-file
```

Description

The parameter *path-to-obp-file* defines the location of the image file.

C Appendix – OpenBoot Console

C.1 OpenBoot Console Overview

The Charon-SSP SPARC virtual machines use a subset of the Sun OpenBoot console found on native Sun workstations and servers. The figure below shows the initial console screen at boot on a virtual SPARCstation 20.

```

SPARCstation 20 OpenBoot console

SMCC SPARCstation 20 Emulator by Stromasys

CPU_#0      TI, TMS390Z50 (3.x)      0Mb External cache
CPU_#1      ***** NOT installed *****
CPU_#1      ***** NOT installed *****
CPU_#1      ***** NOT installed *****

>>>> Power On Self Test (POST) is running .... <<<<<

SPARCstation 20 (1 X 390Z50), No Keyboard
Emulate OBP Rev. 2.25, 64 MB memory installed, Serial #12648430.
Ethernet address 2:c:29:4a:d3:29, Host ID: 72c0ffee.

Type help for more information

Can not load boot block!
ok

```

C.2 OpenBoot Console Command Reference

The following sections describe the currently supported console commands.

C.2.1 banner

Display power-on banner.

Syntax

```
banner
```

Description

Use this command to display the power-on banner.

Example

The following example demonstrates the output of the banner command on Charon-SSP configured as a SPARCstation 20.

```

Example banner command output

ok banner
SPARCstation 20 (1 X 390Z50), No Keyboard
Emulate OBP Rev. 2.25, 64 MB memory installed, Serial #12648430.
Ethernet address 2:c:29:4a:d3:29, Host ID: 72c0ffee.

```

C.2.2 boot

Load operating system.

Syntax

```
boot [device-alias] [boot-args]
```

Description

This command boots the specified *device-alias* passing any optional *boot-args* to the kernel. The *boot-args* must be recognized as valid by the Solaris kernel used. Booting from a ZFS disk is supported starting with Charon-SSP version 1.4.1 if the Solaris version supports this feature.

Starting from Charon-SSP version 2.0.5 there is a special boot argument when booting from CD-ROM:

```
boot cdrom -slow=<sec>
```

This parameter should only be used if there are problems when booting from ISO files resulting in a BAD TRAP error.

For more information about device aliases, see the **devalias** command.

Example

The following example demonstrates the output of the boot command on Charon-SSP configured as a SPARCstation 20 and booting SunOS 4.1.4 from CD-ROM.

Example boot command output

```
ok boot cdrom
Boot device: /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@6,0:d   File and args: -v
Boot Release 4.1.4 (sun4m) #2: Fri Oct 14 11:07:52 PDT 1994
Copyright (c) 1983-1990, Sun Microsystems, Inc.
Boot: Romvec version 3.
root on /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@6,0:d fstype 4.2
Boot: vmunix
.Size: 868352.....
.....
.....+2319136+75288 bytes
Statistics:
SuperSPARC: PAC ENABLED
SunOS Release 4.1.4 (MUNIX) #2: Fri Oct 14 11:09:07 PDT 1994
Copyright (c) 1983-1993, Sun Microsystems, Inc.
```

C.2.3 devalias

Display device aliases.

Syntax

```
devalias
```

Description

These commands display the current device aliases. This shows the link between the aliases, such as `cdrom` and the devices shown in the device tree, listed by **show-devs**.

Example

The following example demonstrates the output of the **devalias** command.

Example devalias command Output

```
ok devalias
ttyb          /obio/zs@0,100000:b
ttya          /obio/zs@0,100000:a
keyboard!    /obio/zs@0,0:forcemode
keyboard     /obio/zs@0,0
floppy       /obio/SUNW,fdtwo
scsi         /iommu/sbus/espdma@f,400000/esp@f,800000
net-auri     /iommu/sbus/ledma@f,400010:auri/le@f,c00000
net-tpe      /iommu/sbus/ledma@f,400010:tpe/le@f,c00000
net          /iommu/sbus/ledma@f,400010/le@f,c00000
disk         /iommu/sbus/espdma@f,400000/esp@f,800000/sd@3,0
cdrom        /iommu/sbus/espdma@f,400000/esp@f,800000/sd@6,0:d
tape         /iommu/sbus/espdma@f,400000/esp@f,800000/st@4,0
tape1        /iommu/sbus/espdma@f,400000/esp@f,800000/st@5,0
tape0        /iommu/sbus/espdma@f,400000/esp@f,800000/st@4,0
disk3        /iommu/sbus/espdma@f,400000/esp@f,800000/sd@3,0
disk2        /iommu/sbus/espdma@f,400000/esp@f,800000/sd@2,0
disk1        /iommu/sbus/espdma@f,400000/esp@f,800000/sd@1,0
disk0        /iommu/sbus/espdma@f,400000/esp@f,800000/sd@0,0
```

C.2.4 help

Display OpenBoot console help.

Syntax

```
help [command]
```

Description

Use this command to display the list of commands supported by the OpenBoot console. For brief help on individual commands specify the **command** parameter.

Example

Example help command output

```
ok help
Following commands are supported by this version:
boot      devalias   nvalias   nvunalias
printenv  setenv     probe-scsi show-devs
reset     banner    history   help

Enter 'help command-name' for more help
Examples: help setenv
```

C.2.5 history

Display console command history.

Syntax

```
history
```

Description

This command displays a list of all commands previously entered at the OpenBoot Console.

Example

The following example demonstrates the output of the history command.

Example history command output

```
ok history
 1 printenv
 2 help
 3 help devalias
 4 help history
 5 help probe-scsi
 6 probe-scsi
 7 show-devs
 8 banner
```

C.2.6 nvalias

Stores devalias values in NVRAMRC.

Syntax

```
nvalias <alias> <device-path>
```

Description

Stores the device aliases in NVRAMRC. The alias persists until the **nvunalias** or **set-defaults** command is executed.

Example

The following example demonstrates the use of the `nvalias` command to create and store a device alias named `disk3` that represents a SCSI disk with a target ID of 3 on a SPARCstation 10 system.

Example nvalias command

```
ok nvalias disk3 /pci@1f,0/pci@1,1/ide@3/disk@3,0
```

C.2.7 nvunalias

Removes a device alias from NVRAMRC.

Syntax

```
nvunalias <alias>
```

Description

Deletes the corresponding alias from NVRAMRC.

C.2.8 printenv

Display environment variables.

Syntax

```
printenv
```

Description

Use this command to print the current and default values of OpenBoot console variables.

Example

The following examples illustrate the output of the **printenv** command.

| Sample printenv command output on Charon-SSP/4M | | |
|---|--------------|---------------|
| ok printenv | | |
| Parameter Name | Value | Default Value |
| auto-boot? | false | true |
| local-mac-address? | false | true |
| boot-file | -v | |
| boot-device | disk:a disk1 | disk net |
| ttya-mode | 9600,8,n,1,- | 9600,8,n,1,- |
| ttyb-mode | 9600,8,n,1,- | 9600,8,n,1,- |

| Sample printenv command output on Charon-SSP/4U | | |
|---|--------------------------------|---------------|
| ok printenv | | |
| Variable Name | Value | Default Value |
| auto-boot? | false | true |
| local-mac-address? | true | true |
| output-device | ttya | screen |
| input-device | ttya | keyboard |
| boot-file | -v | |
| boot-device | /pci@1f,4000/scsi@3/disk@1,0:a | disk net |
| ttya-mode | 9600,8,n,1,- | 9600,8,n,1,- |
| ttyb-mode | 9600,8,n,1,- | 9600,8,n,1,- |
| diag-file | -v | |
| diag-device | net | disk net |
| diag-switch? | true | |

C.2.9 probe-scsi

Scan SCSI bus for attached devices.

Syntax

```
probe-scsi
```

Description

This command scans the SCSI bus to locate attached devices.

Example

The following example demonstrates the output of the **probe-scsi** command on system with a single virtual CD-ROM.

| Example probe-scsi command output | | | |
|-----------------------------------|------|------------------------------|--|
| ok probe-scsi | | | |
| Target 0 | | | |
| Unit 0 | Disk | virtual Scsicdrom (c)SRI0200 | |

C.2.10 quit

Shutdown virtual machine.

Syntax

```
quit
```

Description

Use this command to shut down the virtual machine.

Example

The following example demonstrates the output of the **quit** command on Charon-SSP configured as a SPARCstation 20.

Example quit command output

```
ok quit
The system will be shutdown soon...
```

C.2.11 reset

Restart the system.

Syntax

```
reset
```

Description

This command restarts the SPARC virtual machine.

Example

The following example demonstrates the output of the **reset** command on Charon-SSP configured as a SPARCstation 20.

Example reset command output

```
ok reset

          SMCC SPARCstation 20 Emulator by Stromasys

CPU_#0      TI, TMS390Z50 (3.x)          0Mb External cache

CPU_#1      ***** NOT installed *****
CPU_#1      ***** NOT installed *****
CPU_#1      ***** NOT installed *****

          >>>> Power On Self Test (POST) is running .... <<<<

SPARCstation 20 (1 X 390Z50), No Keyboard
Emulate OBP Rev. 2.25, 64 MB memory installed, Serial #12648430.
Ethernet address 2:c:29:4a:d3:29, Host ID: 72c0ffee.

Type help for more information
ok
```

C.2.12 setenv

Set console environment variables.

Syntax

```
setenv variable value
setenv variable --
```

Description

This command sets a console configuration variable to a specific value. The current and default values of the variables are shown by the **printenv** command. To restore a variable to its default value, specify '--' in place of the value. For a complete list of possible variable names and their descriptions, see the table below.

| Variable | Description |
|--------------------|---|
| auto-boot? | If true, boots automatically after power on or reset. |
| local-mac-address? | If true, the MAC address of the network card is used instead of the system MAC address. |
| output-device | Output device used at power-on. |
| input-device | Input device used at power-on. |
| boot-device | Space-separated list of devices to define boot attempt sequence. |
| boot-file | A string of arguments to be passed to the boot loader (e.g., -a or -v). |
| ttya-mode | Serial line configuration for ttya |
| ttyb-mode | Serial line configuration for ttyb |
| diag-file | Diagnostic mode boot arguments. |
| diag-device | Diagnostic startup source device. |
| diag-switch? | Indicates if system should run in diagnostics mode. |

Changes to environment variables are stored in NVRAM and are permanent. However, they only take effect after executing the **reset** command.

Example

The following example illustrates the use of the **setenv** command.

Example of the setenv command

```
ok setenv auto-boot? true
auto-boot? = true
```

C.2.13 show-devs

Display device tree.

Syntax

```
show-devs
```

Description

This command displays the tree of devices visible from the console.

Example

The following example demonstrates the output of the **show-devs** command.

| Example show-devs command output |
|--|
| <pre>ok show-devs /TI,TMS390Z50@f,f8fffffc /SUNW,sx@f,80000000 /eccmemctl@f,0 /virtual-memory@0,0 /memory@0,0 /obio /iommu@f,e0000000 /openprom /aliases /options /packages /obio/power@0,a01000 /obio/auxio@0,800000 /obio/SUNW,fdtwo@0,700000 /obio/interrupt@0,400000 /obio/counter@0,300000 /obio/eeprom@0,200000 /obio/zs@0,0 /obio/zs@0,100000 /iommu@f,e0000000/sbus@f,e0001000 /iommu@f,e0000000/sbus@f,e0001000/SUNW,bpp@f,4800000 /iommu@f,e0000000/sbus@f,e0001000/ledma@f,400010 /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000 /iommu@f,e0000000/sbus@f,e0001000/ledma@f,400010/le@f,c00000 /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000 /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/st /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd /packages/obp-tftp /packages/deblocker /packages/disk-label</pre> |

C.2.14 show-disks

Show all disks found by the system (SCSI and VDS devices). Available only on Charon-SSP/4V.

Syntax

| |
|------------|
| show-disks |
|------------|

Description

This command lists all the disks (SCSI and VDS devices) found by the system. Example of an VDS device:

| Example show-disks command output |
|--|
| <pre>ok show-disks DISK: /virtual-devices@100/channel-devices@200/disk@0</pre> |

D Appendix – Command-Line Utilities Reference

In many cases, it may be preferable to be able to perform maintenance and management tasks for the command-line of the Linux host system. This section describes how to set up the PATH environment variable to use these utilities as well as a comprehensive reference.

D.1 Prerequisites

The utilities described in this reference section are installed as a part of the Charon-SSP Agent for Linux software. Depending on your environment, you may not need or want support for the GUI environment. If this is the case, you can disable the Charon-SSP Agent service.

D.2 Disabling the Charon-SSP Agent Service

Use the following commands to **disable the Charon-SSP Agent service** and avoid automatic start up at system boot.

```
# systemctl stop ssp-agentd
# systemctl disable ssp-agentd
```

Important information: a problem exists in versions 4.0.x before version 4.0.4 that will cause the Agent to stop all active emulator instances that were started by the Charon Manager or configured via the Charon Manager for automatic startup at host system boot when the Agent itself is stopped. Please review the release notes for more details and the description of a workaround.

D.3 Configure the Shell Path

C Shell PATH login profile:

To add the command-line utilities to a C Shell environment, add the following to the end of **.login** (or create a file for **systemwide** settings, e.g., **/etc/profile.d/charon-ssp.csh**):

```
setenv PATH $PATH:/opt/charon-agent/ssp-agent/utils/license
setenv PATH $PATH:/opt/charon-agent/ssp-agent/utils/mkdisk
setenv PATH $PATH:/opt/charon-agent/ssp-agent/utils/mktape
```

Bourne Shell login profile:

To add the command-line utilities to a Bourne Shell environment (e.g., bash or sh), add the following to the end of **.profile**, **.bash_profile**, or **bashrc**, or create a file for **systemwide** settings, e.g., **/etc/profile.d/charon-ssp.sh**):

```
PATH=$PATH:/opt/charon-agent/ssp-agent/utils/license
PATH=$PATH:/opt/charon-agent/ssp-agent/utils/mkdisk
PATH=$PATH:/opt/charon-agent/ssp-agent/utils/mktape
export PATH
```

D.4 Tools Reference

The following utilities can be used from the command-line to support and manipulate the Charon-SSP host environment.

D.4.1 Sentinel HASP License Utilities

D.4.1.1 hasp_srm_view

Name

hasp_srm_view – Charon Sentinel HASP Utility

Synopsis

```
/opt/charon-agent/ssp-agent/utills/license/hasp_srm_view [OPTION]
```

Description

The **hasp_srm_view** utility provides a simple command-line utility for gathering Sentinel license information. If no options are specified, **-l** is specified by default.

| Parameter | Description |
|-----------------------|---|
| -?, -h, -help | Display the utility usage message. |
| -c2v FILENAME | Collect the Sentinel HASP key status information and write it to FILENAME . |
| -fgp FILENAME | Collect the host fingerprint information for generating a new Sentinel software license to FILENAME . |
| -l | Show the product license details for default key. |
| -all | Show the product license details for all available keys. |
| -key KEYNUMBER | Show the product license details for specific key. If used with -c2v , extract the C2V file for a specific key (available starting from Charon-SSP version 2). |

Exit Status

The **hasp_srm_view** utility exits with 0 on success and with a non-zero value if an error occurs.

Retrieve C2V data:

- 0 – success
- 1 – failure

Create a host fingerprint:

- 0 – success
- 1 – failure

Command-line argument processing:

- 0 – success
- 1 – missing argument
- 2 – unknown argument

The command also returns return code zero if the command itself ran successfully, but a Sentinel restriction was encountered (e.g., trying to display a license over a remote connection). In such cases, an error message is displayed.

Examples

The following example shows the output of the **-l** qualifier for an attached Sentinel USB HASP key.

```
License Manager running at host: localhost.localdomain
License Manager IP address: 127.0.0.1
The Physical KeyId: 663427931
CHARON Sentinel HASP License key section
Reading 4032 bytes
License Manager running at host: localhost.localdomain
License Manager IP address: 127.0.0.1

The License Number: 1002783
The License KeyId: 663427931
The Master KeyId: 2131943298
Release date: 24-MAR-2015
Release time: 11:47:56
Update number: 3
End User name: Stromasys Asia Pacific
Purchasing Customer name: Stromasys Asia Pacific

Virtual Hardware: SPARCstation_20
Product Name: Charon-SSP/4M for Linux x64
Product Code: CHSSP-xxxxx-LI
Major Version: 1
Minor Version: 0
Maximum Build: 99999
Minimum Build: 1
Host CPU supported: X64
Host Operating System required: LINUX
CPU's allowed: 1
Maximum virtual memory: 512MB
Instances allowed: 4
Released product expiration date: 01-Oct-2015
Field Test product expiration date: 01-Oct-2015
```

The following example shows how to create a C2V (customer-to-vendor) file for requesting a license update from Stromasys.

```
$ /opt/charon-agent/ssp-agent/utils/license/hasp_srm_view -c2v /tmp/hasp.c2v
```

D.4.1.2 hasp_update

Name

hasp_update – Sentinel HASP Update and Transfer Utility

Synopsis

```
/opt/charon-agent/ssp-agent/utils/license/hasp_update u filename
```

Description

The **hasp_update** utility provides a simple command-line interface for manipulating the HASP License Key.

u Apply the HASP key update found in **filename**.

For hardware licenses, you will receive two update files in most cases. In such cases, the format key (*_fmt.v2c) file must always be applied first.

Exit Status

The **hasp_update** program exits with **0** on success and with **255** if an error occurs.

Examples

The following example demonstrates how a V2C (vendor-to-customer) license key file is applied.

```
# /opt/charon-agent/ssp-agent/utils/license/hasp_update u /tmp/0002_1002784.v2c
```

D.4.2 Storage Utilities

D.4.2.1 mkdiskcmd

Name

mkdiskcmd – Charon virtual disk container creation utility.

Synopsis

```
/opt/charon-agent/ssp-agent/utils/mkdisk/mkdskcmd [PARAMETERS] [OPTIONS] ...
```

Description

Create virtual disk container files for use with the Charon family of virtual machines. By default, this utility displays a usage message.

Mandatory arguments to long options are mandatory for short options too.

| Parameter | Description |
|---------------------------------|--|
| -a, --avtable PATHNAME | Use this option to specify an alternate location, PATHNAME , of the known disk device table. |
| -c, --blcount BLOCKCOUNT | Specify the number of blocks, BLOCKCOUNT , in the virtual disk container file. Use this option with -z or --blsize to set the block size. |
| -d, --disk NAME | Specify the NAME of a known disk type. Use -l (--list) to see a list of disk types supported by the utility. |
| -h, --help | Display the utility usage message. |
| -l, --list | Display a list of the known disk types. |
| -o, --output FILENAME | Specify the pathname of the virtual disk container file |
| -s, --silent | Do not write any output to the terminal. |
| -z, --blsize BLOCKSIZE | Specify the BLOCKSIZE in bytes when creating a custom virtual disk container file. This option must be used with -c (--blcount) . |

Exit Status

The **mkdiskcmd** utility exits with 0 on success and with a non-zero value if an error occurs.

Examples

The following example creates the virtual disk container file `disk0.vdisk` using the geometry of a Seagate ST446452W 46GB disk drive.

```
# /opt/charon-agent/ssp-agent/utils/mkdisk/mkdskcmd -o disk0.vdisk -d ST446452W
```

This example creates a virtual disk container file (`disk0.vdisk`) using a block size of 4,096 bytes and 16,384 blocks.

```
# /opt/charon-agent/ssp-agent/utils/mkdisk/mkdskcmd -o disk0.vdisk -z 4096 -c 16384
```

D.4.2.2 mtd

Name

mtd – Charon virtual tape creation utility.

Synopsis

```
/opt/charon-agent/ssp-agent/utls/mktape/mtd [PARAMETERS] [OPTIONS] ...
```

Description

This utility can convert a physical tape or a tar archive file to a vtape container file usable by the emulator. It can also convert a vtape container file to a physical tape or a tar archive. Starting with Charon-SSP 5.x, conversion between random Unix files and vtape format is also supported. By default, this utility displays the usage message.

| Commands and options | Description |
|---|--|
| mtd <tape-device> <vtape-file> [options] | Convert a physical tape to a vtape container file. <ul style="list-style-type: none"> <i>tape-device</i>: name of the physical tape device <i>vtape-file</i>: name of the virtual tape file |
| mtd <vtape-file> <tape-device> | Convert a vtape container to a physical tape. |
| mtd -u2v <unix-file> <vtape-file> | Convert a Unix file to a vtape container. <ul style="list-style-type: none"> <i>unix-file</i>: name of the Unix file <i>vtape-file</i>: name of the virtual tape file <p>Please note: the filetype of the Unix file must be a type understood by the Solaris guest for which the vtape is prepared.</p> |
| mtd -v2u <vtape-file> <unix-file> | Convert a vtape container to a Unix file. <p>Please note: the filetype of the file written to the vtape by the Solaris guest must be a type understood by the receiving Linux system.</p> |
| Options: | |
| /log=<file name> | Log file name |
| /reads=<number> | Number of attempts to read a damaged data block |
| /ignore | Try to ignore the bad blocks and continue execution |
| /buffer=<buffer size> | The unit of buffer size in kilobytes. The default size is 64k. If the tape block size is larger than that, you should indicate it to prevent the backup from failing. |

D.4.3 Other Utilities

D.4.3.1 The create-config Utility

The **create-config** utility (introduced in Charon-SSP version 5.2.6, renamed to use “-“ instead of “_” in version 5.2.7) enables the user to **create** and **modify** an emulator configuration file from the command-line without using a text editor.

The **create-config** utility has the following syntax and parameters:

Usage: `/opt/charon-agent/ssp-agent/bin/create-config [options]`

The options are described in the table below:

| Option | Description |
|-------------------------|--|
| -h, --help | Print the usage information. Default if no parameter is selected. |
| -f, --file[=] | Name of the configuration file. If the file does not exist, it is created. If it already exists, it is modified by additional commands – not overwritten. |
| -s, --section[=] | Configuration file section. Only one section can be specified for each execution of the create-config command. |
| -k, --key[=] | Key within a configuration section. Only one key/value pair can be specified for each execution of the create-config command. |
| -v, --value[=] | Value of the specified key |

The table below shows a list of available section names and the keys within them. For the possible values that can be assigned to each key, please refer to the [Configuration File Reference](#).

| Section name | Available keys |
|---------------------|--|
| system | machine, cpu_affinity, io_affinity, io_cpus, model |
| log | path, severity, destination, rotation |
| ram | size, allocator |
| license | regular_key_id, backup_key_id, server, server_key, backup_server, backup_server_key |
| graphics | type, dual_display, display1, display2, remote_display, remote_port1, remote_port2, console, mouse_port, keyboard_port, keyboard_layout, resolution, full_screen, refresh_rate |
| cpu | ht, number, idle, dit, fp_boost, int_boost, dit_page_size, code_cache_size |
| nvrAM | path, hostid, disable_autoboot |
| chassis | part_number, serial, manufacturer, product, firmware |
| vconsole | type, port, restrict_access, log_path, start_console |
| ttya | type, port, restrict_access, log_path, start_console |
| ttyb | type, port, log_path |
| tty_[1-32] | type, port |
| tty[c-q] | type, port |
| digi_ppt_[0-3] | path |
| gpib_[0-3] | path |
| vds_disk[0-119] | path, pass_through, write_through |
| vds_cdrom[0-119] | path, pass_through, write_through |
| scsi_[0-6 or 8-15] | type, lun_[0-7], pass_through, removable |
| scsix_[0-6 or 8-15] | type, lun_[0-7], pass_through, removable |

| Section name | Available keys |
|-----------------|-----------------------|
| parallel | printer |
| floppy | type, path |
| audio | enable, server |
| usb | enabled |
| ethernet | interface, model, mac |
| ethernet_[1-18] | interface, model, mac |

Example to set the memory size of an emulated system to 1GB (in configuration file `conf1.cfg`):

```
$ /opt/charon-agent/ssp-agent/bin/create-config -f conf1.cfg -s ram -k size -v 1024
```

Please note:

- To create a configuration file, you must run the **create-config** command for each line of the configuration file.
- Any section that is not configured will be disabled for the emulated instance.
- If the built-in console of the Charon Manger should be enabled after importing the configuration with the **import-config** utility, you must set the parameter **start_console** (`vconsole` or `ttya`) to **off** (in addition to configuring the serial line itself).

D.4.3.2 The verify-config Utility

The **verify-config** utility (introduced in Charon-SSP version 5.2.6, renamed to use “-“ instead of “_” in version 5.2.7) checks a configuration file for syntactical correctness.

The **verify-config** utility has the following syntax and parameters:

Usage: `/opt/charon-agent/ssp-agent/bin/verify-config [options]`

The options are described in the table below:

| Option | Description |
|--|--|
| -f, --config <i>config-file</i> | Emulator configuration file to be imported (the filename must have the extension cfg , e.g., <code>myconfig.cfg</code>). |
| -h, --help | Print the usage information. Default if no parameter is selected. |

If an error is found, the command will exit with the corresponding message and exit status 1. If no error was found, the command will exit quietly with exit status 0.

The command

```
echo $?
```

can be used to check the exit status in the bash shell.

D.4.3.3 The import-config Utility

The Charon Manager offers an option to import an emulator configuration not created via this Charon Manager into the management environment of the local Manager and Agent.

Starting with version 5.2.6 (renamed to use “-“ instead of “_” in version 5.2.7), there is a command-line utility: **`/opt/charon-agent/ssp-agent/bin/import-config`**

The **import-config** utility must be run as the **root** user. It has the following syntax and parameters.

Usage: **`/opt/charon-agent/ssp-agent/bin/import-config`** *[options]*

The options are described in the table below:

| Option | Description |
|--|---|
| -n, --name <i>vm-name</i> | Name of the emulated instance |
| -f, --config <i>config-file</i> | Emulator configuration file to be imported (*.cfg) |
| -j, --jit | If set, server JIT is used. Please note that if JIT is disabled in the configuration (dit = off), this parameter will have no effect. |
| -p, --plus | If set, 4U+ or 4V+ is being used (as applicable). The import command does not check if the host system supports the necessary CPU capabilities. |
| -a, --auto-boot <i>delay</i> | Enables emulator start with host system start and configures a delay in seconds for the start (0 to 300) |
| -h, --help | Print the usage information. Default if no parameter is selected. |

Please note: the *ssp-agentd.service* will be restarted when the command is run; a Charon Manager connected to this Agent will have to reconnect.

E Appendix – Additional Information for Cloud

E.1 Dedicated NIC for Guest System

Providing a dedicated NIC for guest operating systems is the standard method in non-cloud environments. However, this configuration poses some challenges in cloud environments where MAC address / IP address combinations are fixed parameters set by the cloud provider. This section will provide some information about how to configure such a setup in a cloud environment.

E.1.1 Basic Concept

The following images illustrates the basic concept when working with a dedicated network interface for the guest operating system. There are, of course, many variations depending on the specific environment.

Scenario: host and guest system have a dedicated NIC. The NIC used by the Charon host has a private and a public IP address, the NIC used by the guest system a private IP address and optionally a public IP address. The Internet and VPN gateways are only used for illustration and are not part of this example.

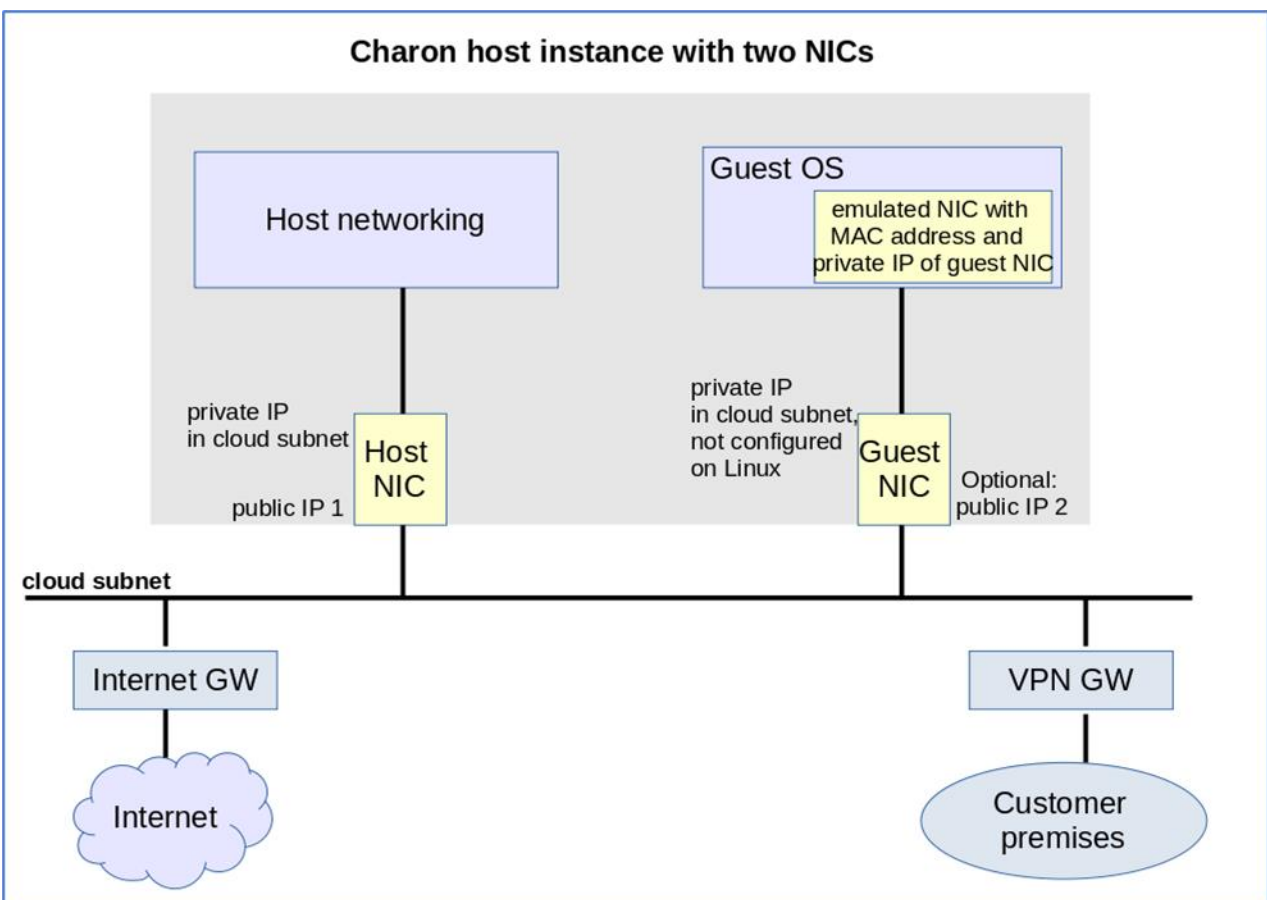


Figure 121: Cloud Dedicated Guest NIC

Please note: If the NIC dedicated to the guest OS does not have a public IP address, the guest system may still be able to access the Internet via the customer network reachable across a VPN gateway. This will depend on the customer specific network configuration. This type of connection is the recommended way to provided external network access to the guest system as the VPN ensures that traffic across a public network is encrypted.

The basic steps to implement the above configuration are as follows:

- Create a cloud instance in which the Charon host system runs.
- Add two NICs to the Charon host system. One for the Charon host and one for the guest system.
- Configure the appropriate access rules for instance and NICs.
- One NIC is dedicated to the Charon host, one to the guest system. Configure a private and public IP address for the NIC used by the Charon host. Configure a private IP address for the NIC used by the guest system (and optionally a public IP address - not recommended).
- On the Charon host, remove the private IP address from the NIC dedicated to the guest system if it was automatically configured and ensure that the interface will be enabled when the system starts.
- Assign the appropriate NIC to the guest system.
- Configure the guest system MAC address to be the same as the MAC of the host NIC selected for the guest.
- After booting the guest system, configure the private IP originally assigned to the guest NIC by the cloud provider as the IP address of the guest Ethernet interface.
- Set the default route of the guest system to the default gateway or VPN gateway of the LAN.

Depending on firewall rules and cloud-specific security settings, the guest system should then be able

- to communicate with the host system,
- other systems in cloud-internal network (e.g., other guest and host systems),
- the customer internal network via a previously configured VPN gateway,
- directly with the Internet if a public IP address was configured for the interface (not recommended).

The additional sections in this chapter show the basic configuration steps for the above example.

Please note:

- In this scenario any traffic between host and guest system (if configured with a public IP address) and external systems reachable via the Internet gateway is not encrypted by default. If this traffic runs across a public network, it is exposed to being monitored and even modified by third parties. The user is responsible for ensuring data protection conforming to the user's company security rules. It is strongly recommended to use encrypted VPN connections for any sensitive traffic.
- Guest operating systems are often old and no longer maintained by the original vendor. This means they are more easily compromised by attacks from the Internet. Therefore, direct Internet access for the guest system is not recommended.
- The actual configuration steps vary depending on the cloud environment used. The sample configuration below will have to be adapted to the specific environment.

E.1.2 Configuration Examples

Important information:

- The example assumes that the host operating system is a RHEL version 7 or 8 compatible Linux system. If you use a different host operating system version, you must adapt the example accordingly. If you use a different host operating system version, you must adapt the example accordingly.
- **Especially for AWS:** note that any automatically assigned public IP addresses will be removed by the cloud provider once the instance is restarted with a second NIC. Hence, on AWS Elastic IP addresses must be used.
- **Especially for Google cloud:**
 - The default is that all interfaces are configured with IP addresses automatically by GCP services on the Linux host. Please refer to the Network Management section in the respective Getting Started guide for information on how to disable this automatic configuration.
 - Some base images used to create a Charon host instance may be configured to use /32 netmasks for additional interfaces, and only ARP requests for the default gateway are answered by Google. This can cause communication problems between Solaris and other instances on the same subnet (ARP requests are not answered). The workaround is to use static ARP entries on Solaris. Please refer to the Getting Started guide for more information. The latest images provided by Stromasys use /24 netmasks, so this point does not apply to them.
- The interface names used in this example (eth0 and eth1) may be different on your system. Please verify the names on your system and refer your cloud provider's documentation for more detail.
Make sure you use the correct names!
- The example uses only a private address for the dedicated interface. If a public address is required, the basic steps for making the interface available to the guest system are the same.

E.1.2.1 Step 1: Prepare Second NIC on Host for Use by the Solaris Guest

The host system interface configuration must ensure that the private IP address allocated to the new interface by the cloud provider is not configured on the Linux Ethernet interface. This address will be used by the guest system.

The configuration depends on whether the network configuration on the Linux host is file-based (typically Linux 7.x) or NetworkManager-based (typically Linux 8.x). Examples for both are shown below. Typically, the NetworkManager is disabled on Charon-SSP marketplace images based on Linux 7.x, and enabled on Charon-SSP marketplace images based on Linux 8.x.

Expected result of the examples in this section:

- The system should still be reachable via eth0.
- Interface eth1 should be up without having an IP address configured.

E.1.2.1.1 File-based Configuration Example

This configuration applies to systems with a file-based network configuration where the NetworkManager is either not active, or where network interfaces should be excluded from NetworkManager control (e.g., to be managed by the Charon Manager). The NetworkManager is disabled by default in older Charon-SSP marketplace images that are based on Centos 7.

Please note:

- The sample configuration assumes a CentOS 7 system and that the interface is configured outside the control of the NetworkManager.
- Should the NetworkManager be active, the plugins **ifcfg-rh** and **ibft** must be enabled in section main of the NetworkManager configuration file `/etc/NetworkManager/NetworkManager.conf`.
- After the initial creation of the ifcfg-file, the interface can be managed by the Charon-SSP Manager.

To make the second interface usable for the Charon guest system, perform the following steps:

1. Add a second interface to your instance as described in the cloud-specific Getting Started guide and your cloud provider's documentation.
2. Log into the instance and become the root user (use: `sudo -i`)
3. Identify the names of the two Ethernet interfaces:
`# ip link show`
4. Create an interface configuration file for the second interface (the file for the first one should exist).
Example (use correct interface name for your configuration):
`# cp /etc/sysconfig/network-scripts/ifcfg-eth0 \
/etc/sysconfig/network-scripts/ifcfg-eth1`
5. Edit this file to match the characteristics of **eth1** (use correct interface name for your configuration). The private IP address used for this interface will be assigned to the Solaris guest. Therefore, configure the Linux Interface without IP address, like the example below.

Please note:

On some cloud platforms, the automatic cloud-specific configuration prevents the entries in the ifcfg-file to take effect (for example on GCP). Please refer to your cloud-provider's documentation and the Network Management section in the Getting Started Guide of your version for additional information.

```
BOOTPROTO=none
DEVICE=eth1
NAME=eth1
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
NM_CONTROLLED=no
```

6. Restart the network:
`# systemctl restart network`

Please note: Should there be an error when executing this command, kill the DHCP client process and retry the command.

E.1.2.1.2 NetworkManager-based Configuration Examples

The following sections show two examples:

- Configuration using nmcli commands
- Configuration using the Charon-SSP Manager

Using nmcli Commands

To configure the interface dedicated to the emulator such that it receives no IP address but is activated at start, you could use command similar to the following:

1. Identify the NetworkManager connection to configure. In the example, it is "Wired connection 1" on device eth1.

```
# nmcli conn show
NAME                UUID                                TYPE      DEVICE
System eth0        5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03  ethernet  eth0
Wired connection 1  027a1c2b-3397-37fb-a6e2-f2e02eb59992  ethernet  eth1
```

2. Configure an appropriate name for the connection if required:

```
# nmcli conn mod "Wired connection 1" con-name eth1
```

3. Set the IP configuration such that no IP address is assigned:

```
# nmcli conn mod eth1 ipv4.method manual ipv4.address 0.0.0.0
```

4. Configure automatic interface activation at boot:

```
# nmcli conn mod eth1 connection.autoconnect yes
```

5. Restart the interface to activate the configuration changes (note that syntax used below ensures that there is no accidental interruption of connectivity that could happen if the two commands were entered separately):

```
# nmcli con down eth1 && nmcli con up eth1
```

Using the Charon Manager

The Charon-SSP Manager provides basic network configuration options.

- To access them, start the Charon Manager and open the menu option:
Tools > Network Settings
- To configure a host system for use by the emulator perform the following steps
 - **Select** the correct interface.
 - In the **IP setting** field select **None**.
 - Click on **Apply**.

E.1.2.2 Step 2: Add the Dedicated NIC to the Emulator Configuration

- Start the Charon Manager and open the configuration window for the emulated system.
- Configure the emulated system with the dedicated Ethernet interface as its interface.
- Set the MAC address to the same value as used by the host interface (the value assigned by your cloud provider).
- Save your configuration.

E.1.2.3 Step 3: Configure the Solaris Guest to Use the Private Cloud IP Address

1. Using the steps below, the Solaris guest system is configured to use the second NIC configured on the host system (please refer to your Solaris documentation for configuration details).

Boot Solaris and configure the IP address assigned to the dedicated guest NIC for the Solaris Ethernet interface as shown in the examples below:

```
# ifconfig <interface-name> <private-guest-nic-ip>/<netmask> up (Solaris 10 example)
or
# ifconfig <interface-name> <private-guest-nic-ip> netmask<mask> up (Solaris 2.6 example)
or
# ipadm create-ip netX and ipadm create-addr -T static -a <private-guest-nic-
ip>/<netmask> netX/v4 (Solaris 11 example)
```

For Solaris versions before version 11, make permanent by editing **/etc/hosts** and set the new address for the systems hostname. Then edit **/etc/netmask** and add the netmask for the subnet-network.

2. Add default route on Solaris:

```
# route add default <default-gateway-of-cloud-lan> <metric>
```

Make permanent by editing **/etc/defaultrouter** and add the address of the gateway (use **route -p** for newer Solaris versions).

3. Add DNS server to Solaris (if needed):
 1. Edit **/etc/resolv.conf** and add a nameserver line for the DNS server.
 2. Make sure, DNS is used for hostname resolution: ensure that **/etc/nsswitch.conf** is configured to allow **dns** (in addition to **files**) for the hostname resolution.

For Solaris 11, please refer to [the Oracle Solaris documentation](#).

Expected result (depending on security rules and firewalls):

1. The guest system should be able to communicate with the host system across the cloud LAN using the private IP addresses.
2. The guest system should be able to communicate directly with the Internet if the dedicated NIC has a public IP address (not recommended).

Please note: Do not forget that traffic transmitted across the Internet by the guest system is not encrypted by default. Take appropriate measures to protect your data. It is strongly recommended to protect the Solaris guest system by an appropriate firewall and security group configuration. If possible, any communication across the Internet should be encrypted (e.g., by using a VPN).

F Index

B

Barebone

- Barebone ISO, 15
- Enroll key for secure boot, 64

Baremetal

- Root partition size, 58, 247, 249
- User accounts (Baremetal), 66
- User interface (GUI), 67

C

Charon-SSP CLI, 175

Charon-SSP command-line utilities, 299

Charon-SSP configuration file, 261

Charon-SSP deinstallation, 252

Charon-SSP GUI for Microsoft Windows, 254

Charon-SSP installation

- Baremetal installation, 58
- Bootable USB device, 59
- Dependencies, 42
- Installation commands, 45
- Installation directory content, 74
- Installation steps (RPM), 46
- License installation, 36
- License requirements, 42
- PATH variable configuration, 53
- RPM installation, 38
- RPM package overview, 39
- UEFI secure boot, 63

Charon-SSP packaging, 16

Charon-SSP products

- Baremetal appliance**, 15
- Product variants, 25

Charon-SSP upgrade, 242

Cloud

- Data transfer options, 207
- Dedicated NIC for guest, 308
- Networking behavior, 152
- User accounts (cloud images), 65

Cloud-specific images, 15

E

Emulated (virtual) hardware

- Emulated hardware families, 21
- Supported virtual hardware, 23

Emulator config

- Serial lines (Vconsole), 118

Emulator configuration

- 4V memory limitations, 98
- Audio configuration, 133

Batch configuration creation, 305

Charon-SSP Director overview, 76

Charon-SSP Manager Getting Started, 81

Charon-SSP Manager overview, 85

Charon-SSP software components, 75

Chassis configuration, 116

Container file creation, 105

CPU configuration parameters, 93

Default I/O CPU calculation, 94

Emulated graphics devices, 99

Emulated USB port, 135

Ethernet configuration, 137

Ethernet configuration parameters, 139

External serial console, 119, 122, 184

GPIO configuration, 131

Graphics configuration parameters, 101

License configuration, 141

Log configuration parameters, 143

Memory configuration parameters, 98

NVRAM configuration, 140

Parallel port, 132

Performance optimization, 95

Physical disks parameters, 111

Power option for hyper-threading, 94

SCSI configuration parameters, 107, 114

SCSI device drivers, 103

Serial lines (TTYA), 120

Serial lines (TTYX), 123

Solaris 11.4 SCSI drivers, 103

VDS Storage Devices, 112

Virtual Machine creation, 88

Virtual SCSI devices, 107, 113

Emulator operation

Charon Agent operation, 180

Charon Agent preferences, 162

Charon management password, 170

Command-line options, 172

Jumpstart, 206

Public SSH key, 215

SSH VPN Tunnel, 214

Start, Stop, Suspend, 145

Virtual machine context menu, 149

Virtual machine import, 161

G

Graphical guest console device, 187

Guest graphical X-Display, 188

H

Host system management

- Baremetal and Cloud additional tools, 165

- Baremetal GUI, 67
- Charon-SSP Baremetal, 66
- Network configuration, 151
- Network interface parameters, 154
- SSP user accounts (Baremetal), 66
- SSP user accounts (cloud), 65
- Virtual bridge configuration, 156
- VLAN interface configuration, 159
- X11-Forwarding for Charon Manager, 84

I

- iSCSI tool, 199

L

- Licensing
 - AutoVE mode, 20
 - Charon-SSP Manager licensing tools, 225
 - HASP command-line utilities, 229
 - HASP licensing, 224
 - Licensing options, 19
 - Sentinel ACC configuration file, 52
 - Sentinel Admin Control Center (ACC), 231

N

- NetworkManager and Linux versions, 31

- NFS tools**, 203

O

- OpenBoot (OBP) console, 291

P

- PuTTY key format, 216

R

- Requirements

- Firewall, 34
- Hardware requirements, 28
- Linux Operating System, 30

S

- Serial guest console (Manager), 182
- Serial guest console (network), 183
- Serial guest console (physical), 181

T

- Troubleshooting

- Charon Agent log file, 180
- Core dump files, 164
- CPU allocation error, 94
- Default I/O CPU calculation, 94
- Disk with non-zero LUN ID not recognized, 110
- Insufficient hardware for 4U+/4V+, 97
- Log files, 144
- Manager to Agent connection, 83
- Not enough CPUs, 94
- Password reset, 171
- SCSI ID is in use message, 108
- Unable to create I/O thread, 145
- Virtual CD-ROM SCSI bus/target, 107, 271, 274
- VT-x or AMD-v not supported, 96

V

- VNC server, 205

X

- XDMCP configuration, 189