Cisco 5921 Embedded Services Router Integration Guide

Current Release: February 12, 2020

Text Part Number: 78-21252-01

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco 5921 Embedded Services Router Integration Guide

© 2021 Cisco Systems, Inc. All rights reserved.

Preface

This preface describes the audience, organization, and documentation conventions for this guide and provides information on how to obtain related documents and technical assistance.

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This preface includes the following major sections:

Audience

This guide is for experienced systems designers and system integrators with competent knowledge of Linux and Cisco IOS who are incorporating the Cisco 5921 Embedded Services Router (ESR) into their designs.

Organization

This guide is organized into the following chapters:

Chapter	Title	Description
1	Product Overview	Introduces the Cisco 5921 Embedded Services Router
2	Installing Cisco IOS	Contains information on required libraries, special cases for the Cisco 5921 Embedded Services Router, and untarring the tar file
3	Understanding the Cisco 5921 ESR	Contains detailed information on the applications and configuration files within the Cisco 5921 Embedded Services Router

Chapter	Title	Description
4	Creating the Cisco 5921 ESR Reference Platform	Describes how to create a reference platform and run your system through the reference platform to obtain troubleshooting information for Cisco Customer Support
5	Troubleshooting	Includes tips, ideas, and Frequently Asked Questions
Appendix A	Command Reference	Provides basic commands with descriptions
Appendix B	Glossary Terms and Acronyms	Defines acronyms and abbreviations used in this guide

Related Documentation

Documentation for the 5921 includes the following documents:

- Software Configuration Guide for the Cisco 5900 Embedded Services Routers
- Release Notes for Cisco IOS Software Release—see these release notes for release-specific information
- Open Source Used in the 5900 Series
- The file *RELEASE_NOTES_C5921.txt*—release notes for the Cisco 5921 Embedded Services Router located in the Cisco 5921 Embedded Services Router tar file.
- *README_C5921.txt* —located in the Cisco 5921 Embedded Services Router tar file.
- Man pages for the Cisco IOS ESR application swr-application.1, swroptions.1, swr_reload and swrvcon.1—located in the Cisco 5921 Embedded Services Router tar file.

Conventions

This document uses the following typographical conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in **boldface**.
italic font	Command arguments for which you supply values are in *italics*.
[ ]	Command elements in square brackets are optional.
{ x I y I z }	Alternative keywords in command lines are grouped in braces and separated by vertical bars.
[x I y I z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A non-quoted set of characters. Do not use quotation marks around the string because the string will include the quotation marks.
`screen` font	System displays are in `screen` font.

Convention	Description
`boldface screen` font	Information you must enter verbatim is in **`boldface screen`** font.
`italic screen` font	Arguments for which you supply values are in *`italic screen`* font.
→	This pointer highlights an important line of text in an example.
^	Represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Non-printing characters such as passwords are in angle brackets.

Notes use the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Cautions use the following conventions:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warnings use the following conventions:

Warning **Safety warnings appear throughout this publication in procedures that, if performed incorrectly, may cause harm to you or the equipment. A warning symbol precedes each warning statement.**

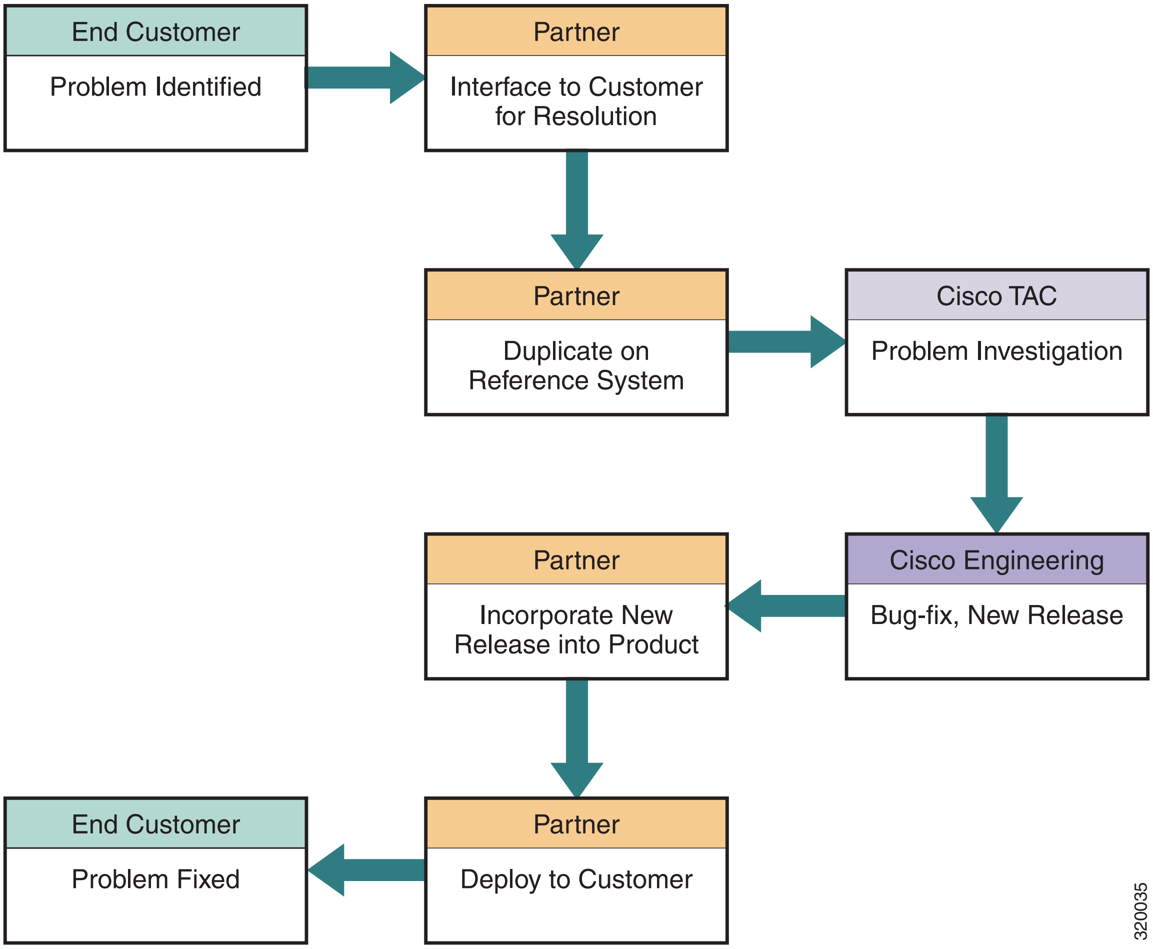
Obtaining Documentation, Support, and Security Guidelines

The System Integrator is responsible for the serviceability of the system and for providing all direct end customer support. Cisco Customer Support is available to the System Integrator provided appropriate service agreements have been obtained. To isolate issues related to the target system or the integration of the Cisco 5921 ESR into this system, issues must be duplicated on the Reference Platform prior to contacting Cisco Customer Support. See chapter 4 "Creating the Cisco 5921 ESR Reference Platform" for detailed information on the Cisco 5921 Embedded Services Reference Platform.

To obtain additional documentation or information, contact your Cisco Sales Representative, or email *ask-c5921-team@cisco.com*.

Figure 1 outlines the generalized support flow for the Cisco 5921 ESR.

Figure 1 Cisco 5921 Embedded Services Router Support Model



4

Product Overview

The Cisco 5921 Embedded Services Router is Cisco IOS compiled as an x86 32-bit Linux application. It is designed to run on a host hardware platform using the Linux kernel 2.6.32 or later.

This chapter provides the following major sections to introduce the Cisco 5921 Embedded Services Router (ESR):

ESR 5921 Evaluation Procedure

Follow these steps if you wish to perform an evaluation of the Cisco ESR 5921:

1. Install the ESR 5921 on your Linux system.
2. Enable Smart Licensing with no Smart Account.
3. Enable the desired throughput level.
4. Instant 90 day evaluation commences.

Note dCloud has 3 levels of training for the 5921. Please do the introductory training as that covers how to install and test the product. See details at the end of this chapter.

Requirements

The Cisco 5921 ESR has the following hardware requirements:

- The host platform must be Intel x86 or compatible family with a minimum 512 Mbytes of RAM.

Note The Linux version may impose additional hardware requirements.

The Cisco 5921 ESR has the following Linux software requirements:

- The Linux kernel must be version 2.6.32 or greater.
- Standard Linux packages installed on the Linux system.

- The Cisco 5921 ESR has no Linux distribution restrictions. You are responsible for building and verifying functionality.

Cisco has tested the Cisco 5921 ESR with CentOS using desktop and embedded configurations.

Components

The Cisco 5921 ESR is a 32-bit application. It will work on a 64-bit Linux system, provided you install and use the 32-bit libraries. The i86 image does not include the 32 bit libraries, whereas the x86 image does include them.

There are two different images for the 5921. The Enterprise Based Image, and the Universal Image.

Note The Universal Image was previously known as the Advanced Enterprise Image.

There are different images for the 5921. For details on the images, see the following table:

Image Name	Description
c5921**i**86-entbasek9-ms.SPA.<*release*>.bin	Supports base set of IOS features
c5921**i**86-entbasek9-tar.SPA.<*release*>.tar	
c5921**i**86-universalk9-ms.SPA.<*release*>.bin	Supports base and advanced set of IOS features
c5921**i**86-universalk9-tar.SPA.<*release*>.tar	
c5921**i**86-universalk9_npe-ms.SPA.<*release*>.bin	Supports base and advanced set of IOS features but no enhanced security features available
c5921**i**86-universalk9_npe-tar.SPA.<*release*>.tar	
c5921**i**86**v**-universalk9-tar.SPA.<*release*>.tar	Supports base and advanced set of IOS features with unlimited SL (Smart License) entitlement and run only on virtual machines
c5921**x**86-entbasek9-ms.SPA.<*release*>.bin	Supports base set of IOS features and runs on a 64 bit machine
c5921**x**86-entbasek9-tar.SPA.<*release*>.tar	
c5921**x**86-universalk9-ms.SPA.<*release*>.bin	Supports base and advanced set of IOS features and runs on a 64 bit machine
c5921**x**86-universalk9-tar.SPA.<*release*>.tar	
c5921**x**86-universalk9_npe-ms.SPA.<*release*>.bin	Supports base and advanced set of IOS features but no enhanced security features available and runs on a 64 bit machine
c5921**x**86-universalk9_npe-tar.SPA.<*release*>.tar	
c5921**x**86**v**-universalk9-tar.SPA.<*release*>.tar	Supports base and advanced set of IOS features with unlimited SL (Smart License) entitlement and runs on a 64 bit virtual machine

Note The tar file contains all of the software components listed below. The binary file is the application and is just one of the components.

Software Components

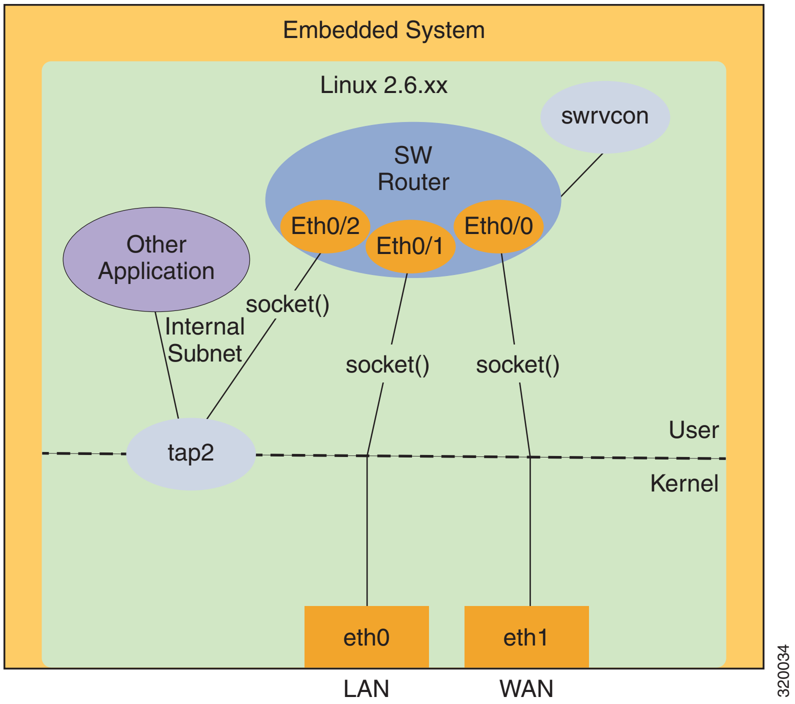
The Cisco 5921 ESR is contained in a compiled tar file with the following components:

- c5921i86-universalk9-ms—application that contains the Cisco IOS code.

- swrvcon—Linux application that provides a virtual console to access Cisco IOS
- swr_reload—Used to launch and reload the software router in the event that the underlaying Linux OS develops any problems.
- libdyncs.so—Application-specific shared library.
- Man pages for the Cisco 5921 ESR applications/files: swr-application, swr_reload, swroptions, and swrvcon.

Figure 4-1 shows how these component relate to each other.

Figure 4-1 Interactions between Cisco 5921 ESR Components



Using Cisco IOS MIB Tools

This section describes how to access the Cisco IOS MIB Tools web page. The web page provides:

- The MIB Locator to find MIBs in Cisco IOS software releases
- A link to how to load Cisco MIBs
- A link to SNMP MIB technical tips
- The SNMP Object Navigator which translates SNMP object identifiers (OIDs) into SNMP names

To access the Cisco IOS MIB tools site, follow these steps:

Step 1 Go to the Cisco Products and Services page:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

Step 2 Click MIB Locator to launch the application.

Step 3 From the MIB Locator page, click the drop-down menu and select the desired Cisco IOS software release.

Step 4 After you make a selection, follow the links and instructions.

Interface Monitoring

The IOS software is able to acquire status information from raw Linux ports. The status information received includes:

- Link State (up or down)
- Link Speed (10, 100, or 1000 Mbps)
- MTU
- Full or Half Duplex

The Linux interface is polled for a state change whether the current status is up or down. From this information, the mapped IOS interface is updated. The IOS can only monitor and reflect what the Linux interface state is, it can not control the state of the Linux interface.

For the interfaces to be monitored, the following conditions must be met:

- In the interface section of the SWROPTIONS file, `monitor-state=true` must be present for the mapped interface.
- The interface in IOS needs to be configured as `no shut`.

The interface state can be verified with the `show platform software interface-status` command.

dCloud Training

dCloud has free training for the 5921. Training is available to all Cisco employees and Cisco Partners. Cisco field sales account teams and partners can launch a session, make it available for up to 5 days long (dCloud resources permitting) and share the session with a prospective end user.

There are 3 different labs for the ESR 5921.

Cisco 5921 Embedded Services Router Introductory Lab - Installation and Basic Configuration v1

In this demonstration, we are going to feature the Cisco 5921 Embedded Services Router (ESR) installed on an Ubuntu core Linux system in a command line environment. For most embedded applications, their operating environment is usually that of a “black box” with no displays at all. The demonstration section walks the user through the highlights and features of this solution, to improve the end user’s familiarity with the Cisco 5921 ESR. No previous Linux experience is required, since Router 1 and Router 2 have operational preinstalled Cisco 5921 ESR’s. If you have learned the very basics of the IOS command line for the Integrated Services Routers or Catalyst Switches, you will be quite comfortable with this demonstration.

The purpose of the lab sections of this document is to enhance your skills as an integrator by giving a concrete example of how to qualify a Linux embedded core build for use with the Cisco 5921 Embedded Services Router (ESR), as well as, taking you through the install and troubleshooting of the Cisco 5921 ESR installation on that system. Finally, we have included reference materials to provide additional information about this product.

Cisco 5921 Embedded Services Router Intermediate Lab - Exploring IRB and IPSec Features v1

- Scenario 1: Explore Integrated Routing and Bridging
- Scenario 2: Configure Site-to-Site VPN

Cisco 5921 Embedded Services Router Advanced Lab - Exploring DLEP v1

Overview

The Cisco 5921 Embedded Services Router (ESR) is designed to operate on small, low-power, Linux-based platforms. It helps integration partners extend the use of Cisco IOS into extremely mobile and portable communications systems. It also provides highly secure data, voice, and video communications to stationary and mobile network nodes across wired and wireless links.

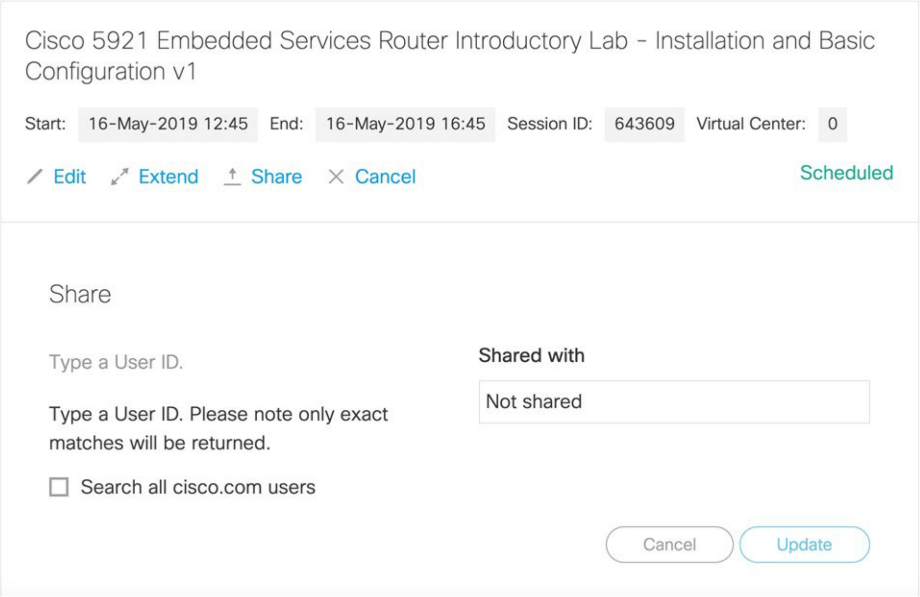
In this lab, we will focus on the Cisco implementation of Dynamic Link Exchange Protocol (DLEP) by using a radio simulator tool and networking environment that makes use of ESR as the platform of choice.

Scenarios

- Scenario 1: Configuring DLEP in a Multipath Environment with a Single Radio
- Scenario 2: Configuring DLEP in a Multipath Environment with Multiple Radios
- Scenario 3: How the Cisco Implementation of DLEP Redistributes Routing Metrics
- Scenario 4: Troubleshoot DLEP

Once the lab is scheduled it can be shared with a prospective end user by clicking the “**Share**” button and entering the person’s cisco.com user id. See Figure 4-2.

Figure 4-2 Sharing the Lab



Cisco 5921 Embedded Services Router Introductory Lab - Installation and Basic Configuration v1

Start: 16-May-2019 12:45 End: 16-May-2019 16:45 Session ID: 643609 Virtual Center: 0

Scheduled

Share

Type a User ID.

Shared with

Not shared

Type a User ID. Please note only exact matches will be returned.

□ Search all cisco.com users

TCP Performance Optimization

TCP performance degradation can occur when using Linux driver optimizations. After disabling Linux interface optimizations, the expected performance returned.

To view current optimizations that are applied, the ethtool command can be used:

```
ethtool -k <interface listed from iftable>
  generic-receive-offload: on  <------- This will toggled on
```

To disable the optimizations, use the following command:

```
ethtool -K <interface listed from iftable> rx off tx off sg off tso off gso off gro off
```

5

Installing Cisco IOS

This chapter discusses how to install Cisco IOS on your hardware platform, retrieve and load the license correctly, and run Cisco IOS.

This document assumes that you have Linux kernel version 2.6.32, or greater, installed on your hardware platform with standard packages also installed.

This document contains the following sections:

Licensing Overview

There are two types of licensing available on the Cisco 5921 ESR. Classic licensing and Smart Licensing. Support for the two types of licensing is based on the following parameters:

Installed within a virtual machine.

- Only Smart Licensing is supported.
- Default state bandwidth is 32 kbps.

Installed directly on a device.

- Both Classic and Smart Licensing would be supported and either Classic or Smart Licensing would be active at any instance of time.
- On transition from Classic Licensing to Smart Licensing mode, installed Classic Licensing licenses would not be transferred automatically in Smart Licensing mode. The user must configure the appropriate speed and ask for Entitlements from the Smart Licensing Back-end using the Command Line Interface. For example:

```
license platform throughput level {c5921-x86-level0|c5921-x86-level1|
c5921-x86-level2|c5921-x86-level3|c5921-x86-level4|c5921-x86-level5|c5921-x86-level6}
```

- After transitioning from Smart Licensing mode to Classic Licensing, a reboot would be required for Classic Licensing licenses to get activated.
- Default evaluation state bandwidth is 32 kbps.

Smart Licensing

Cisco Smart Licensing is a flexible licensing model that provides you with an easier, faster, and more consistent way to purchase, activate, manage, renew, deploy, and upgrade software products across the Cisco portfolio and across your organization. Smart Licensing streamlines the way you activate and manage licenses with the following:

- **Easy Activation**. Smart Licensing establishes a pool of software licenses that can be used across the entire organization
- **Unified Managemen**t. With My Cisco Entitlements (MCE), you get a single pane of glass view into all of your Cisco products and services in an easy-to-use portal, so you always know what you have and what you are using.
- **License Flexibility**. Your software is not node-locked to your hardware, so you can easily use and transfer licenses as needed.

In order to retrieve your Smart Licenses, you will first need to set up a **Smart Account** on Cisco Software Central (software.cisco.com).

For a more detailed overview on Cisco Licensing, go to **cisco.com/go/licensingguide**.

Smart License PIDs

License PID	Description	Enforced Rate in Mbps
LS-FL-5921-XL0-K9	Smart License for Cisco 5921 ESR - Adv Ent - 5Mbps	5
LS-FL-5921-XL1-K9	Smart License for Cisco 5921 ESR - X86 Level 1 - 10Mbps	10
LS-FL-5921-XL2-K9	Smart License for Cisco 5921 ESR - X86 Level 2 - 25Mbps	25
LS-FL-5921-XL3-K9	Smart License for Cisco 5921 ESR - X86 Level 3 - 50Mbps	50
LS-FL-5921-XL4-K9	Smart License for Cisco 5921 ESR - Adv Ent - 100Mbps	100
LS-FL-5921-XL5-K9	Cisco 5921 ESR Adv Ent SW - 200 Mbps	200
LS-FL-5921-XL6-K9	Cisco 5921 ESR Adv Ent SW - 500 Mbps	500
LS-FL-5921-BL0-K9	Smart License for Cisco 5921 ESR - Base SW - 5Mbps	5
LS-FL-5921-BL1-K9	Smart License for Cisco 5921 ESR - Base SW - 10Mbps	10

License PID	Description	Enforced Rate in Mbps
LS-FL-5921-BL2-K9	Smart License for Cisco 5921 ESR - Base SW - 25Mbps	25
LS-FL-5921-BL3-K9	Smart License for Cisco 5921 ESR - Base SW - 50Mbps	50
LS-FL-5921-BL4-K9	Smart License for Cisco 5921 ESR - Base SW - 100Mbps	100
LS-FL-5921-BL5-K9	Cisco 5921 ESR Base SW - 200 Mbps	200

Enabling Smart Licensing

Smart Licensing can be enabled in configuration mode with the following command.

```
Router(config)#[no] license smart enable
```

After enabling Smart Licensing on the device Classic Licensing is disabled and all licensing calls will now go through the Smart Agent.

Registering to the Cisco Back-End

The device can be registered to the Cisco Back-end with the following command executing in exec mode.

```
Router#[no] license smart register idtoken <idtoken>
```

When the device supplies the tokenID via the Command Line Interface to the Cisco Back-end, the back-end responds back with a Device Certificate that is valid for a 365-day period. Now the device enters into Authorized mode.

A tokenID is a token generated by the Administrator for a Virtual Account from which the license to be used for the device.

Enable Throughput level

License Throughput level can be enabled in configuration mode with the following command:

```
Router(config)# [no] license platform throughput level
[c5921-x86-level0|c5921-x86-level1|c5921-x86-level2|c5921-x86-level3|c5921-x86-level4|c592
1-x86-level5|c5921-x86-level6]
```

Verifying the License State

Use the following commands to view and verify the current license state.

```
Router#show license tech support
Smart Licensing Tech Support info

Smart Licensing Status

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Smart Account: c5921
  Virtual Account: Default Virtual Account
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Oct 20 05:48:01 2015 IST
  Last Renewal Attempt: None
  Next Renewal Attempt: Apr 17 05:48:01 2016 IST
  Registration Expires: Oct 19 05:45:17 2016 IST

License Authorization:
  Status: AUTHORIZED on Oct 20 05:49:55 2015 IST
  Last Communication Attempt: SUCCEEDED on Oct 20 05:49:55 2015 IST
  Next Communication Attempt: Nov 19 05:49:55 2015 IST
  Communication Deadline: Jan 18 05:47:12 2016 IST
Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 90 days, 0 hours, 0 minutes, 0 seconds

License Usage
=============
Handle: 1
    License:
regid.2015-05.com.cisco.c5921-x86-level4,1.0_437326de-91d3-42e3-b20f-bffe6f8bfd20
    Entitlement Tag:
regid.2015-05.com.cisco.c5921-x86-level4,1.0_437326de-91d3-42e3-b20f-bffe6f8bfd20
    Description: This entitlement tag was created via Alpha Extension
                 application
    Count: 1
    Version: 1.0
    Status: AUTHORIZED(3)
    Status time: Tue Oct 20 05:49:55.000
    Request Time: Oct 20 05:49:18 2015 IST

Product Information
==================
UDI: PID:CISCO5921-K9,SN:9NI4877PW72
Agent Version
=============
Smart Agent for Licensing: 1.4.0_rel/20
Component Versions: SA:(1_4_rel)1.0.19, SI:(rel21)1.0.0, CH:(rel4)1.0.21, PK:(dev18)1.0.13

Upcoming Scheduled Jobs
=======================
Current time: Oct 20 05:57:22 2015 IST
IdCert Expiration Warning: Aug 20 05:45:17 2016 IST (304 days, 23 hours, 47 minutes, 55
seconds remaining)
Daily: Oct 21 05:47:29 2015 IST (23 hours, 50 minutes, 7 seconds remaining)
Certificate Renewal: Apr 17 05:48:01 2016 IST (179 days, 23 hours, 50 minutes, 39 seconds
remaining)
Certificate Expiration Check: Oct 19 05:45:17 2016 IST (364 days, 23 hours, 47 minutes, 55
seconds remaining)
Authorization Renewal: Nov 19 05:49:54 2015 IST (29 days, 23 hours, 52 minutes, 32 seconds
remaining)
Authorization Expiration Check: Jan 18 05:47:11 2016 IST (89 days, 23 hours, 49 minutes,
49 seconds remaining)
Init Flag Check: Not Available
Register Period Expiration Check: Not Available
Ack Expiration Check: Not Available

License Certificates
===================
Production Cert: False
PIID: 6971b1b3-16c9-4eb4-98f9-2f98028f10c9
Licensing Certificated:
    Id certificate Info:
        Start Date: Oct 20 05:45:18 2015 IST
        Expiry Date: Oct 19 05:45:18 2016 IST
        Version Number: 3
        Serial Number: 224911
        Common Name: 7686AFFC5BD74A5E2BFB975BA45592D0A0DAB392::1,2

    Signing certificate Info:
        Start Date: Jun 14 14:48:52 2013 IST
        Expiry Date: Apr 24 16:25:42 2033 IST
        Version Number: 3
        Serial Number: 3
        Common Name: MMI Signer

    Sub CA Info:
        Start Date: Apr 24 16:49:15 2013 IST
        Expiry Date: Apr 24 16:25:42 2033 IST
        Version Number: 3
        Serial Number: 2
        Common Name: Smart Licensing CA - DEV


HA Info
==========
RP Role: Active
Chassis Role: Active
Behavior Role: Active
RMF: True
CF: True
CF State: Stateless

Other Info
==========
Software ID: regid.2014-08.com.cisco.C5921,1.0_56a2b9b1-1a84-474d-92b4-f0db266657dd
Agent State: authorized
TS enable: False
Transport: Callhome
Locale: en_US.UTF-8
Debug flags: 0x0
Privacy Send Hostname: True
Privacy Send IP: True
Build type:: Production
sizeof(char)  : 1
sizeof(int)   : 4
sizeof(long)  : 4
sizeof(char *): 4
sizeof(time_t): 4
sizeof(size_t): 4
Endian: Big
Write Erase Occurred: False
enableOnInit: False
routingReadyByEvent: True
systemInitByEvent: True
platformOverrideEvent: UnknownPlatformEvent
WaitForHaRole: False
standbyIsHot: False
chkPtType: 2
delayCommInit: False
roleByEvent: True
maxTraceLength: 150
traceAlwaysOn: False
debugFlags: 0

Router#
```

```
Router#show license feature
WARNING: the API function SmartAgentShowFeatures has been deprecated. Use the new _v2
function

Entitlement:
    Tag:
regid.2015-05.com.cisco.c5921-x86-level4,1.0_437326de-91d3-42e3-b20f-bffe6f8bfd20,
Version: 1.0, Enforce Mode: Eval period
    Requested Time: Oct 22 05:24:22 2015 UTC,  Requested Count: 1
    Vendor String:
```

The following output shows the pre-15.6(1)M licensing:

```
Router#show platform software license

Packet forwarding: Enabled

Current enforcement forwarding rate: 25 Mbps

Unique Device Identifier: CISCO5921-K9:9D21806DBOR

License features supported:
       Feature              Rate     Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps    -
  c5921-x86-evaluation      50 Mbps    -
  c5921-x86-level1          10 Mbps    -
  c5921-x86-level2          25 Mbps  In Use
  c5921-x86-level3          50 Mbps    -

Router#
```

The following output shows the15.6(1)M and above licensing:

```
Router#show platform software license
Packet forwarding: Enabled

Current enforcement forwarding rate: 200 Mbps

Unique Device Identifier: CISCO5921-K9:92K4304QKB3

License features supported:
       Feature              Rate     Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps    -
  c5921-x86-evaluation      50 Mbps    -
  c5921-x86-level0           5 Mbps    -
  c5921-x86-level1          10 Mbps    -
  c5921-x86-level2          25 Mbps    -
  c5921-x86-level3          50 Mbps    -
  c5921-x86-level4         100 Mbps    -
  c5921-x86-level5         200 Mbps  In Use
  c5921-x86-level6         500 Mbps    -

Router#
```

Note If the "write erase" and "reload" commands are done after doing "wr mem" at least once, the License throughput level has to be re-enabled after a reload.

Note The Smart License configuration must be saved before a reload of the 5921 to save the authentication keys, otherwise a keys mismatch happens. It is recommended to do a 'write mem' of Smart License configuration before reload.

Note From releases 15.6(3)M6,15.7(3)M4 and 15.8(3)M, license enforcement to default 8kbps throughput level, once evaluation/license expired.

Smart Licensing States

The Smart Licensing Agent keeps track of the following states:

State Name	Description and Command	Licensing State
Un-configured	Smart Licensing has not been enabled	"Not Enabled"
Un-registered (un-identified)	Smart Licensing is enabled but product instance has not registered with Cisco license cloud service. **`license smart enable`**	"Success" unless Evaluation period has expired in which case it will be "Evaluation Expired"
Registered	Short live state indicating that the instance has successfully registered with Cisco but has not received an authorization for its MMI request. **`license smart register idtoken <token>`**	"Success"
Authorized	The product instance is fully authorized for all entitlement requests (in compliance) **`license platform throughput level <MB>`**	"Success"
Out of Compliance	In this state the customer is out of compliance with their contract. They have either used too many licenses /entitlements or they are not authorized to use a particular entitlement tag.	"Out of Compliance"
Expired	The agent will enter into this state if the grace period (90 days) also is expired after authorization renewal has not happened for 90 days.	"Success" if there is still a 90 grace period after the authorization has expired or "Unavailable" if the grace period has also expired.

Classic Licensing

The Cisco 5921 ESR uses licenses issued by Cisco Software Licensing (CSL). After you identify that you want to include the Cisco 5921 ESR in your system, you need to determine the license level based on bandwidth requirements so that you can then purchase the license from Cisco. Note that the Cisco 5921 ESR is made available to authorized customers only. Contact your Cisco account representative if you are not authorized and want to find out if and how you can be authorized. To acquire a Unique Device Identifier (UDI), first request and receive the Cisco 5921 ESR tar file from your Cisco account representative, and load it on your target platform. You untar the tar file, load the components onto your system, boot the system and retrieve the UDI. Once you have determined the license level you require, order a Product Authorization Key (PAK) for that license level. Cisco will email you the PAK. Use this PAK and the UDI to receive a license file.

Finally, you load the license file to the directory where you plan to boot the Cisco IOS image and complete the installation of Cisco IOS on your platform.

In summary in order to use the full functionality of Cisco IOS on your hardware platform, you follow this procedure to acquire, receive, and use a license file for the Cisco 5921 ESR. More detailed steps follow this summary.

1. After receiving the tar file from your Cisco account representative untar the tar file and load the components to your hardware platform (page 5-27).
2. Boot the Cisco IOS software on your hardware platform and retrieve the UDI and required license level (page 5-31).
3. Purchase a Cisco IOS license (page 5-33).
4. Use the Package Activation Key (PAK) and UDI to receive a license file from Cisco.com (page 5-34).
5. Upload the license file to the directory where you boot the Cisco IOS software (page 5-34).
6. Boot Cisco IOS and verify that the license is active (page 5-34). The license file must be installed first before it will be active.

Classic License PIDs

License PID	Name used within IOS	Enforced Rate in Mbps
L-FL-5921-XL0-K9	c5921-x86-level0	5
L-FL-5921-XL1-K9	c5921-x86-level1	10
L-FL-5921-XL2-K9	c5921-x86-level2	25
L-FL-5921-XL3-K9	c5921-x86-level3	50
L-FL-5921-XL4-K9	c5921-x86-level4	100
L-FL-5921-XL5-K9	c5921-x86-level5	200
L-FL-5921-XL6-K9	c5921-x86-level6	500

Permanent License Reservation (PLR)

Note PLR support began with the 15.6(3)M1 release, with Unlimited throughput level. Starting with15.9(3)M, support was added for different throughput levels.

The PLR licensing feature allows the use of reserved licenses in a disconnected mode. You can optionally request permanent licenses for each 5921. Permanent licenses do not require periodic access to the License Authority. PLR is supported on the universalk9 and universalk9_npe images.

You can have a pool of PLR entitlements from which to draw from in the virtual account on the Cisco Smart Software Manager (CSSM). When the CSSM validates the reservation request code generated by a product instance doing Universal PLR, it will allocate one of these entitlements to the product instance and generate a string (authorization code) to enter (install) on that product instance.

When you install the authorization code, generated by the CSSM, on the product instance any entitlement request made by the product to the Smart Agent will be authorized.

Prerequisites for the Reservation Request Process

Before you begin, make sure the following apply:

- The device has no connectivity to Cisco.
- Smart Licensing on the device should be enabled and not registered.
- Evaluation period is in use or expired.

Enabling Permanent License Reservation

Smart Licensing can be enabled in configuration mode with the following commands:

Enable/Disable Smart Licensing:

```
Router(config)#[no] license smart enable
```

Enable/disable license reservation feature:

```
Router(config)#[no] license smart reservation
```

Configuring the Throughput Level

You must configure the device ***plr-throughput*** level with the appropriate throughput before generating a reservation request code.

```
Router(config)#[no]license platform plr-throughput level ?
    c5921-x86-UPL-100Mbps  100 Mbps throughput rate
    c5921-x86-UPL-10Mbps   10 Mbps throughput rate
    c5921-x86-UPL-200Mbps  200 Mbps throughput rate
    c5921-x86-UPL-50Mbps   50 Mbps throughput rate
```

Note Required throughput level should be configured prior to generating a reservation request code. Otherwise, it generates a request code with default/unlimited PLR throughput level.

Step 1 Get a reservation request code.

```
Router# license smart reservation request
```

Prints the "Reservation Request Code" that you will need to paste into the CSSM portal. The evaluation period will continue counting down until an Authorization Code is installed.

Step 2 Get the authorization code.

a. Enter the request code into the CSSM portal.

b. CSSM portal provides an authorization code.

Step 3 Install the authorization code.

```
Router# license smart reservation install <Reservation Authorization Code>
```

Step 4 The device is now registered and the evaluation period stops counting down.

There is a cancel CLI to abort an unregistered request code.

```
Router# license smart reservation cancel
```

Note Before generating a new request code, if any changes in the throughput level are made, first cancel the existing request using “license smart reservation cancel. Otherwise the request code will not be updated.

Reservation Return Process

This is done if the product previously used the reservation CLI to reserve and install an authorization code.

Step 1 Use the following command to generates a code to be entered into the CSSM portal.

```
Router# license smart reservation return
```

Note The Agent will transition to the unidentified state and will use the evaluation period if any time is left.

Step 2 Paste the return code into the CSSM portal appropriately and remove the license. All entitlements are now returned to the virtual account.

Step 3 The device will now go back to the unidentified state and the evaluation period will be consumed again.

Step 4 After the reservation return is complete, you will not be able to use the original “Reservation Authorization Code” to re-authorize the device.

Verifying the License State

Use the following command is used to view and verify the current license state:

```
Router#show license tech support
Smart Licensing Tech Support info
Smart Licensing Status
======================

Smart Licensing is ENABLED
License Reservation is ENABLED

Registration:
  Status: REGISTERED - UNIVERSAL LICENSE RESERVATION
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on May 13 04:45:04 2019 PST

License Authorization:
  Status: AUTHORIZED - RESERVED on May 13 04:45:04 2019 PST

Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 90 days, 0 hours, 0 minutes, 0 seconds

License Usage
=============
Handle: 2
    License: <empty>
    Entitlement Tag:
regid.2019-03.com.cisco.c5921-x86-UPL-10Mbps,1.0_d1f80a70-6249-4f66-86d8-8327f0c649b7
    Description: <empty>
    Count: 1
    Version: 1.0
    Status: AUTHORIZED(3)
    Status time: Mon May 13 04:45:04.000
    Request Time: May 13 04:45:04 2019 PST

Product Information
===================
UDI: PID:CISCO5921-K9,SN:9W01262VYXA

Agent Version
=============
Smart Agent for Licensing: 1.6.10_rel/106
Component Versions: SA:(1_6_rel)3.2.18, SI:(rel22)1.5.8, CH:(rel4)2.0.1,
PK:(rel18_twig_2)1.0.1

Upcoming Scheduled Jobs
=======================
Current time: May 13 05:23:06 2019 PST
Reservation request in progress warning: Not Available
Daily: May 14 02:59:45 2019 PST (21 hours, 36 minutes, 39 seconds remaining)
Reservation configuration mismatch between nodes in HA mode: May 14 03:00:45 2019 PST (21
hours, 37 minutes, 39 seconds remaining)
Init Flag Check: Not Available
Evaluation Expiration Check: Not Available
Evaluation Expiration Warning: Not Available

License Certificates
====================
Production Cert: True
Not registered. No certificates installed

HA Info
==========
RP Role: Active
Chassis Role: Active
Behavior Role: Active
RMF: True
CF: True
```

```
CF State: Stateless

Other Info
==========
Software ID:
regid.2019-03.com.cisco.ESR_5921_PLR_10mbps,1.0_685caebd-4239-4f7b-94aa-9934874b0225
Agent State: authorized
TS enable: True
Transport: Callhome
Locale: en_US.UTF-8
Debug flags: 0xF
Privacy Send Hostname: True
Privacy Send IP: True
Build type:: Production
sizeof(char)  : 1
sizeof(int)   : 4
sizeof(long)  : 4
sizeof(char *): 4
sizeof(time_t): 4
sizeof(size_t): 4
Endian: Big
Write Erase Occurred: False
XOS version: 0.12.0.0
Reservation enable: True
Reservation in progress: False
Reservation in progress start time: May 13 04:44:17 2019 PST
Reservation type: Universal
Reservation request code: <empty>
Reservation authorization code: BAACyp-mqBXi4-DYgsGf-JDZTRr-aYQaFV-Y5m1
Reservation return code: <empty>
Config Persist Received: True
Message Version: 1.1
routingReadyByEvent: True
systemInitByEvent: True
bypassUDICheck: False
loggingAddTid: False
platformOverrideEvent: UnknownPlatformEvent
WaitForHaRole: False
standbyIsHot: False
chkPtType: 2
delayCommInit: False
roleByEvent: True
maxTraceLength: 150
traceAlwaysOn: False
debugFlags: 0
```

The following commands shows licensing output.

The following output shows the pre-15.9(3)M licensing:

```
Router# show platform software license
Packet forwarding: Enabled

Unique Device Identifier: CISCO5921-K9:9T45105QNO4

License features supported:
       Feature              Rate     Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps    -
  c5921-x86-UPL           Unlimited  In Use
```

The following output shows 15.9(3)M and above licensing:

```
Router# show platform software license
Packet forwarding: Enabled

Current PLR enforcement forwarding rate: 10 Mbps

Unique Device Identifier: CISCO5921-K9:9T45105QNO4

License features supported:
       Feature              Rate     Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps    -
  c5921-x86-UPL           Unlimited    -
  c5921-x86-UPL-10Mbps      10 Mbps  In Use
  c5921-x86-UPL-50Mbps      50 Mbps    -
  c5921-x86-UPL-100Mbps    100 Mbps    -
  c5921-x86-UPL-200Mbps    200 Mbps    -
  c5921-x86-UPL-500Mbps    500 Mbps    -
```

Loading the Cisco 5921 ESR Components

Previous versions of the c5921 IOS image were intended to work with 32bit operating systems. The c5921 works on a 64bit Linux Operating systems only when 32bit libraries installed.

Starting with this release (15.9(3)M), Cisco added the c5921x86 image with 32bit libraries as part of package for 64bit Linux Operation system. This enables the c5921 to run on 64bit operating systems without pre-installed 32bit libraries.

Installation on a 32 bit system OR a 64 bit system where there are manually pre-installed 32 bit libraries

Software Components

There are two different images for the 5921. The Enterprise Based Image, and the Universal Image.

Note The Universal Image was previously known as the Advanced Enterprise Image.

For details on both images, see Software Components, page 6-39

Contact your Cisco account representative to obtain the Cisco 5921 ESR tar file.

To load the Cisco 5921 ESR components on your platform, perform the following procedure:

Step 1 Create a directory to hold the Cisco 5921 installation:

```
mkdir -p /opt/cisco/c5921
```

Step 2 Extract the contents of the tar file to the installation directory:

```
tar -xf c5921i86-universalk9-ms.[RELEASE_LABEL] -C /opt/cisco/c5921
```

Step 3 Change directory to the location where the files were extracted:

```
cd /opt/cisco/c5921/c5921i86-universalk9-ms.[RELEASE_LABEL]
```

Step 4 List the contents of the c5921 directory. The output should show the following entries:

```
#ls -tlr
swruser@swr-host:~/c5921i86-universalk9-ms_release_REL-[RELEASE_LABEL]$ ls -tlr
FAQ_C5921.txt
c5921i86-universalk9-ms.SPA
c5921i86-universalk9-ms.md5
README_C5921.txt
RELEASE_NOTES_C5921.txt
SWROPTIONS.example.txt
swr-application.1
swr_reload
swr_reload.1
libdyncs.so
swroptions.1
swrvcon
swrvcon.1
c5921-swr-init.sh
```

Table 5-1 Components in the Cisco 5921 ESR tar file

Component File	Description
swrvcon swrvcon.1	Linux application and corresponding man page that provides a virtual console to access Cisco IOS. For more information on swrvcon, see the swrvcon.1 man page.
swr_reload swr_reload.1	The purpose of swr_reload is to launch and reload the software router which we specify in SWROPTIONS file, the command-line option, or from the current working directory. Use the Control-C command to exit the swr_reload application. For more information on swr_reload, see the swrnetio.1 man page.
c5921i86-universalk9-ms swr-application.1	Application that contains the Cisco IOS code. Provides information about this application. Also see the swr-application.1 man page.
libdyncs.so	Application-specific shared library
RELEASE_NOTES_C5921.txt	Engineering release notes for the Cisco 5921 ESR.
README_C5921.txt	Introduction to the Cisco 5921 ESR and a description of the provided components.
SWROPTIONS.example.txt	Provides a sample of the possible content to the SWROPTIONS file.
FAQ_C5921.txt	Questions and answers for common issues.
C5921-swr-init.sh	5921 SWR startup script.

Step 5 Copy all of the files in the untarred directory to the /opt/cisco/*c5921* directory.

Step 6 Keep the original files unaltered as a backup.

Installation on a 64 bit system which does not have 32 bit libraries integrated

Software Components

There are two different images for the 5921. The Enterprise Based Image, and the Universal Image.

For details on both images, see Software Components, page 6-39

Contact your Cisco account representative to obtain the Cisco 5921 ESR tar file.

To load the Cisco 5921 ESR components on your platform, perform the following procedure:

Step 1 Create a directory to hold the Cisco 5921 installation:

```
mkdir -p /opt/cisco/c5921
```

Step 2 Extract the contents of the tar file to the installation directory:

```
tar -xf c5921x86-universalk9-tar.SPA.<RELEASE_LABEL> -C /opt/cisco/c5921
```

Step 3 Change directory to the location where the files were extracted:

```
cd /opt/cisco/c5921/c5921x86-universalk9-SPA.<RELEASE_LABEL>
```

Step 4 List the contents of the c5921 directory. The output should show the following entries:

```
#ls -tlr
swr_reload.1
swr_reload
c5921x86-universalk9-ms.SPA
c5921x86-universalk9-ms.md5
swrvcon.1
swrvcon
libdyncs.so
swrvcon.sh
swr_reload.sh
SWROPTIONS.example.txt
swroptions.1
swr-application.1
RELEASE_NOTES_C5921.txt
README_C5921.txt
FAQ_C5921.txt
c5921-swr-init.sh

ls -tlr libc_2_*

libc_2_23:
ld-linux.so.2
librt.so.1
libpthread.so.0
libnsl.so.1
libm.so.6
libgcc_s.so.1
libdl.so.2
libc.so.6

libc_2_29:
ld-linux.so.2
librt.so.1
libpthread.so.0
libnsl.so.1
libm.so.6
libgcc_s.so.1
```

```
libdl.so.2
libc.so.6
```

Table 5-2 Components in the Cisco 5921 ESR tar file

Component File	Description
c5921-swr-init.sh	5921 SWR startup script.
swr_reload.sh	Introduced to use swr_reload by loading appropriate libraries.
swrvcon.sh	Introduced to use swrvcon by loading appropriate libraries.
swrvcon swrvcon.1	Linux application and corresponding man page that provides a virtual console to access Cisco IOS. For more information on swrvcon, see the swrvcon.1 man page.
swr_reload swr_reload.1	The purpose of swr_reload is to launch and reload the software router which we specify in SWROPTIONS file, the command-line option, or from the current working directory. Use the Control-C command to exit the swr_reload application. For more information on swr_reload, see the swrnetio.1 man page.
c5921x86-universalk9-ms.SPA c5921x86-universalk9-ms.md5	Application that contains the Cisco IOS code.
libdyncs.so libc_2_29 libc_2_23	Application-specific shared library
RELEASE_NOTES_C5921.txt	Engineering release notes for the Cisco 5921 ESR.
README_C5921.txt	Introduction to the Cisco 5921 ESR and a description of the provided components.
SWROPTIONS.example.txt	Provides a sample of the possible content to the SWROPTIONS file.
FAQ_C5921.txt	Questions and answers for common issues.

Step 5 Copy all of the files in the untarred directory to the /opt/cisco/*c5921* directory. To make sure subdirectory contents copied proper, please use -r option while copy. For example:

```
#cp -r /opt/cisco/c5921/c5921i86-universalk9-ms.158/* /opt/cisco/c5921/
```

Step 6 Keep the original files unaltered as a backup.

Loading the c5921 using the init Script:

Make sure the /opt/cisco/c5921/ directory and all the dependent files are available.

Step 1 At the Linux prompt, enter **./c5921-swr-init.sh start** to boot Cisco IOS.

Step 2 **./swrvcon.sh 100**(session id) for virtual console access.

Step 3 **./c5921-swr-init.sh stop** to stop/kill Cisco IOS(c5921).

Note The overall difference between 5921i86 and 5921x86 installation is:
swrvcon.sh should be used in place of swrvcon.
swr_reload.sh should be used inplace of swr_reload.
All options remain the same.

Loading the c5921 using the swr_reload.sh script:

Make sure the /opt/cisco/c5921/ directory and all the dependent files are available.

Step 1 Open the SWROPTIONS.example.txt file with vi or another Linux text editor.

Step 2 Specify the image name using the 'launchapp' and 'soft-rommon' keywords.
soft-rommon=./
launchapp=<*image_name*>

Step 3 Map IOS Ethernet interfaces to linux physical interfaces using either tap interface or raw sockets.

Step 4 Enable interface monitoring using the **monitor-state=true** option.

Step 5 Specify the SWROPTIONS path using **-o <*SWROPTIONS file path*>.**

Step 6 Save your changes to the file.

Step 7 At the Linux prompt, enter **./swr_reload.sh -o /ios/swr/AUTO/SWROPTIONS <*instance_id*>**

Note The c5921 image cannot be loaded directly as ./x5921x86-universalk9-ms.SPA. You must use the swr_reload.sh executable.

Booting Cisco IOS for the First Time

The first time you boot Cisco IOS on your Linux platform, Cisco IOS is not fully functional. For full functionality you are required to install Cisco IOS with a license file from Cisco. To receive this license file, you first need to get the following two pieces of information from Cisco IOS.

- Unique Device Identifier (UDI)—The UDI is composed of a 12-character product identifier (PID) and an 11-character serial number.
- License level—The Cisco 5921 ESR has multiple license levels available to purchase based on total bandwidth usage.

Note Packet drops will be random, and will drop any packet that exceeds the licensed rate.

To boot Cisco IOS and acquire the UDI, perform the following procedure:

Step 1 Navigate to the directory where c5921 software router files are present, typically */opt/*cisco/*c5921*.

Step 2 Open the *SWROPTIONS.example.txt* file with **vi** or another Linux text editor.

Step 3 Define the home directory path using 'linux=<path>'

Step 4 Specify the image name using the 'launchapp' and 'soft-rommon' keywords.

```
soft-rommon=./
launchapp=<image_name>
```

Note 'launchapp' should always be placed below 'soft-rommon' option.

Step 5 Map IOS Ethernet interfaces to linux physical interfaces using either tap interface or raw sockets.

Step 6 Enable interface monitoring using the 'monitor-state=true' option.

Step 7 Save the SWROPTIONS file, preferably under the default directory / (root).

Step 8 At the Linux prompt, enter ./swr_reload to boot Cisco IOS.

```
./swr_reload
```

Step 9 If the SWROPTIONS file is not placed in the default location / (root), then launch swr_reload by specifying the SWROPTIONS path using '-o <SWROPTIONS file path>'

```
./swr_reload -o /ios/swr/AUTO/SWROPTIONS
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

...<output truncated>...

Warning:  the compile-time code checksum does not appear to be present.
Linux 5921 (Intel-x86) processor with 933578K bytes of memory.
Processor board ID 200
8 Ethernet interfaces
512K bytes of NVRAM.


         --- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]:
Press RETURN to get started!
 ...<output truncated>...
Cisco IOS Software, Linux Software (C5921_I86-UNIVERSALK9-M), Version
15.2(20130516_1837)GC, EARLY DEPLOYMENT NIGHTLY BUILD, synced to
GGSG_GC_MAIN_POSTSYNC_PI19
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Thu 16-May-13 20:25 by sundanat
*Nov 19 08:13:41.047: %SNMP-5-COLDSTART: SNMP agent on host Router is undergoing a cold
start
*Nov 19 08:13:41.058: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is OFF
*Nov 19 08:13:41.058: %CRYPTO-6-GDOI_ON_OFF: GDOI is OFF
```

Step 10 When prompted about entering the initial configuration dialog, type **no** and press the **Enter** key.

Step 11 Type **enable** to enter privileged EXEC mode.

Step 12 Enter the **show license udi** command at the privileged EXEC prompt to display the UDI for your platform.

```
Router>enable
Router#show license udi
Device#   PID                   SN              UDI
-----------------------------------------------------------------------------
*0        CISCO5921-K9          9FKC6PI4VIA     CISCO5921-K9:9FKC6PI4VIA

Router#
```

The UDI has two parts: a PID and a serial number.

Step 13 Record the PID and serial number in the **show license udi** command output. You will need this information with the PAK that you receive from Cisco to get your license file.

Step 14 Enter the **show platform software license** at the privileged EXEC prompt to display the list of supported license features.

Pre 15.6(1)M output:

```
Packet forwarding: Enabled

Current enforcement forwarding rate: 32 Kbps

Unique Device Identifier: CISCO5921-K9:9VG0190PJDO

License features supported:
       Feature              Rate     Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps  In Use
  c5921-x86-evaluation      50 Mbps    -
  c5921-x86-level1          10 Mbps    -
  c5921-x86-level2          25 Mbps    -
  c5921-x86-level3          50 Mbps    -

Router#
```

Post 15.6(1)M output:

```
Packet forwarding: Enabled

Current enforcement forwarding rate: 100 Mbps

Unique Device Identifier: CISCO5921-K9:9NI4877PW72

License features supported:
       Feature              Rate     Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps    -
  c5921-x86-evaluation      50 Mbps    -
  c5921-x86-level0           5 Mbps    -
  c5921-x86-level1          10 Mbps    -
  c5921-x86-level2          25 Mbps    -
  c5921-x86-level3          50 Mbps    -
  c5921-x86-level4         100 Mbps  In Use
```

The outputs lists all the Cisco 5921 ESR licenses you may purchase for your particular platform.

Note Packet forwarding will be disabled and all interfaces will remain down until a valid license is installed.

Licensing the Software with Classic Licensing

Once you have the license level and the UDI, you can purchase a Cisco 5921 ESR license.

Contact your Cisco account representative to purchase the Cisco 5921 ESR. Cisco will send you an email with a Product Authorization Key (PAK) for the Cisco 5921 ESR.

Acquiring the License File

After you receive your PAK by email, you can use it and the UDI to acquire your Cisco 5921 ESR license.

To acquire your Cisco 5921 ESR license, perform the following procedure:

Step 1 Log into your Cisco.com account.

Step 2 Go to the following URL:

https://tools.cisco.com/SWIFT/LicensingUI/pakReg.Home

Step 3 In the Software License Tool, s**elect CISCO 5921 ESR** from the Product family drop-down list.

Step 4 Follow the instructions on the Product Registration Page.

You will need your PAK and the UDI (See: "Booting Cisco IOS for the First Time" section on page 5-31).

Step 5 Upon successful registration, Cisco will send you an email with your license file.

The license file is an attachment in the Software License email.

Uploading the Software License

The Software License email contains the license file as an attachment with a .lic filename extension. If this is not the case, please send an email to licensing@cisco.com.

To upload the software license to your platform, perform the following procedure:

Step 1 Verify that the Product Name, Product Description, Product UDI and Product Authorization Key listed in the Software License email are correct.

If anything is incorrect, send an email to licensing@cisco.com.

Step 2 Save the software license to your local computer.

Step 3 Copy the software license to the directory on the Linux machine from which you start the Cisco 5921 ESR, for example: */opt/cisco/c5921/*

Installing the Software License

After Cisco has emailed the Cisco 5921 ESR license to you and you have copied it to the directory from which you start the Cisco 5921 ESR, perform the following procedure to install the software license.

Step 1 At the Linux prompt, enter **./c5921-swr-init.sh start** to boot Cisco IOS.

Step 2 When prompted about entering the initial configuration dialog, type **no** and press the **Enter** key.

Step 3 Type **enable** to enter privileged EXEC mode.

```
Router>enable
Router#
```

Step 4 Enter the **license install** command with the name of the license file from the privileged EXEC prompt to install the Cisco 5921 ESR license. Note that "unix:" is the file system name.

```
Router#license install unix:9FKC6PI4VIA_20120830065202905.lic
Installing licenses from "unix:9FKC6PI4VIA_20120830065202905.lic"
Installing...Feature:c5921-x86-level2...Successful:Supported
1/1 licenses were successfully installed
0/1 licenses were existing licenses
0/1 licenses were failed to install

Router#
*Aug 30 13:55:10: %LICENSE_C5920-6-LICENSE_ACTIVATED: Installed license for feature
c5921-x86-level2 now in use.
*Aug 30 13:55:10: %LICENSE-6-INSTALL: Feature c5921-x86-level2 1.0 was installed in this
device. UDI=CISCO5921-K9:9FKC6PI4VIA; StoreIndex=0:Primary License Storage Router#

Router#
```

Step 5 Enter the **show license** or the **show platform software license** command at the privileged EXEC prompt to verify that the software license installed correctly.

Pre 15.6(1)M output:

```
Router#show license
Index 1 Feature: c5921-x86-default
          Period left: 8  weeks 3  days
          Period Used: 8  minutes 1  second
          License Type: EvalRightToUse
          License State: Active, In Use
          License Count: Non-Counted
          License Priority: Low
Index 2 Feature: c5921-x86-evaluation
Index 3 Feature: c5921-x86-level1
Index 4 Feature: c5921-x86-level2
          Period left: Life time
          License Type: Permanent
          License State: Active, In Use
          License Count: Non-Counted
          License Priority: Medium
Index 5 Feature: c5921-x86-level3
Router#
```

Post 15.6(1)M output:

```
Router#show license
Index 1 Feature: c5921-x86-default
        Period left: 6  weeks 1  day
        Period Used: 2  weeks 2  days
        License Type: EvalRightToUse
        License State: Active, In Use
        License Count: Non-Counted
        License Priority: Low
Index 2 Feature: c5921-x86-evaluation
Index 3 Feature: c5921-x86-level0
Index 4 Feature: c5921-x86-level1
Index 5 Feature: c5921-x86-level2
Index 6 Feature: c5921-x86-level3
Index 7 Feature: c5921-x86-level4
Index 8 Feature: c5921-x86-level5
Index 9 Feature: c5921-x86-level6
```

```
Router#show platform software license
```

Pre 15.6(1)M output:

```
Packet forwarding: Enabled

Current enforcement forwarding rate: 25 Kbps

Unique Device Identifier: CISCO5921-K9:9VG0190PJDO

License features supported:
       Feature              Rate      Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps    -
  c5921-x86-evaluation      50 Mbps    -
  c5921-x86-level1          10 Mbps    -
  c5921-x86-level2          25 Mbps  In Use
  c5921-x86-level3          50 Mbps    -

Router#
```

Post 15.6(1)M output:

```
Packet forwarding: Enabled

Current enforcement forwarding rate: 32 Kbps

Unique Device Identifier: CISCO5921-K9:94T3408DXHU

License features supported:
       Feature              Rate      Status
  ---------------------   ---------  ------
  c5921-x86-default          32 Kbps  In Use
  c5921-x86-evaluation      50 Mbps    -
  c5921-x86-level0           5 Mbps    -
  c5921-x86-level1          10 Mbps    -
  c5921-x86-level2          25 Mbps    -
  c5921-x86-level3          50 Mbps    -
  c5921-x86-level4         100 Mbps    -
```

Note If “write erase” and reload is done after doing “wr mem” at least once, the License throughput level has to be re-enabled after a reload.

Understanding Licensed Bandwidth Throttling

The Cisco 5921 ESR limits egress bandwidth based on the license level installed on the platform. The egress bandwidth is determined by adding the current outbound bandwidth on all interfaces (either physical or TAP). If this calculated bandwidth exceeds the licensed bandwidth, packets will be randomly dropped.

For calculating bandwidth, the Cisco 5921 ESR only uses the L2 header and payload. It does not take into account the 4 byte CRC that would be present on an Ethernet "wire," nor does it take into account the inter-frame gap and pre-amble that would typically be on the "wire." Most traffic generators do take these other items into account.   As a result, with small packets especially, the bandwidth calculated by IOS may be much lower than the bandwidth reported by the traffic generator.

Upgrading the c5921 IOS Image

Use the following steps to upgrade the c5921 IOS image in Linux.

Step 1 Navigate to the directory where the c5921 software router files are present, typically /opt/cisco/c5921 and do backup the IOS image file **c5921i86-universalk9-ms.SPA**.

```
#mv c5921i86-universalk9-ms.SPA c5921i86-universalk9-ms.SPA_backup
```

Step 2 Rename the library file **libdyncs.so** to another name.

```
#mv libdyncs.so libdyncs.so_backup
```

Step 3 Create a directory to hold the new cisco 5921 ESR upgrade image tar file.

```
#mkdir -p /opt/cisco/c5921/upgrade_image
```

Step 4 Extract the contents of the tar file to the installation directory.

```
#tar -xf c5921i86-universalk9-ms.<RELEASE_LABEL> -c /opt/cisco/c5921/upgrade_image
```

Step 5 List the contents of the c5921 directory. The output should show the following entries:

```
#ls -tlr
FAQ_C5921.txt
c5921i86-universalk9-ms.SPA
c5921i86-universalk9-ms.md5
README_C5921.txt
RELEASE_NOTES_C5921.txt
SWROPTIONS.example.txt
swr-application.1
swr_reload
swr_reload.1
libdyncs.so
swroptions.1
swrvcon
swrvcon.1
c5921-swr-init.sh
```

Step 6 Copy the image file **c5921i86-universalk9-ms.SPA** and library file, **libdyncs.so** files in the untarred directory to the /opt/cisco/c5921 directory.

```
#cp /opt/cisco/c5921/upgrade_image/c5921i86-universalk9-ms.SPA libdyncs.so
/opt/cisco/c5921/
```

Step 7 Validate the new upgrade image **c5921i86-universalk9-ms.SPA** and **libdyncs.so** file size present in the directory, typically /opt/cisco/c5921.

Step 8 Reload the router instance of c5921 by logging into router, or at the Linux prompt by entering these commands:

```
#./c5921-swr-init.sh stop ##### To stop c5921 instance
#./c5921-swr-init.start   ##### To restart c5921 instance
```

Step 9 From the privileged EXEC prompt of the c5921, enter the **show version** command to verify that the upgrade image software version installed correctly.

Note The same procedure applies to the downgrade process as well.

Warning **If the host linux kernel or OS upgrade performed where c5921 installed, then make sure after the upgrade process, linux host should retain the same c5921 installed directory/files to be intact so that c5921 will retain the previous license level installed. Therefore, ensure that you only change the two files mentioned above.**

6

Understanding the Cisco 5921 ESR

This chapter provides an introduction to the Cisco 5921 ESR, an overview of the Cisco 5921 ESR software, and a description of its components is provided.

This document contains the following sections:

Introduction

The Cisco 5921 ESR is a software-only product, designed to run on the Linux operating system. It is a Linux application, running entirely in user space, and utilizes the physical network interfaces provided by the host Linux system.

Note In this release of the software, only Ethernet interfaces are supported.

The Cisco 5921 ESR contains within it a complete implementation of the Cisco IOS software with the universalk9 feature set supported.

Software Components

There are two different images for the 5921. The Enterprise Based Image, and the Universal Image.

Note The Universal Image was previously known as the Advanced Enterprise Image.

5921 Enterprise Based Image

Starting with the 15.5(3)M release, a new Enterprise Based image will be available for the 5921 ESR. The feature set provides core routing functionality but removes the advanced IP Security features (IPSec, VPN), Unified Call Manager, and IP Mobility from the 5921 Advanced feature set. It will support three throughputs with Smart Licensing mode including 5, 10, 25, 50, 100, 200, and 500MB.

The image label is c5921i86-entbasek9-tar.SPA.155-3.M.

Note The CSL license is not supported in 15.5(3)M enterprise based image. The enterprise based image supports both node-lock-machine and node-lock-storage license schemes.

5921 Universal Image

The Universal Image is the legacy image that has been available in previous software releases. This does not indicate that it is down-level and should not be used. It simply indicates that it has supported previous versions and is still supported.

Software Image Components

The Cisco 5921 ESR software contains several components:

- c5921i86-universalk9-ms
  The Cisco 5921 ESR application itself, containing the IOS code.It runs as a Linux process.
- swrvcon
  Provides a virtual console for accessing the c5921i86-universalk9-ms process.
- swr_reload
  The application used to launch and reload software router.
- libdyncs.so
  Application-specific shared library

A complete description of these components follows.

c5921i86-universalk9-ms

c5921i86-universalk9-ms is the Cisco 5921 ESR application. It is delivered as a Linux executable application, and is run as a single user-space process. In this release, it is designed to run in the foreground. The standard in, standard out, and standard error of the process can be used to interact with the router. From the users point of view, it looks like an IOS router, and you can use the IOS CLI to manage the router.

Alternatively, c5921i86-universalk9-ms can be executed by the standard Linux init facilities. In this case, the standard out and standard error of the process go to /var/log/messages, allowing a limited form of post-facto debugging. There is no stdin for processes launched via Linux init scripts, and so the Cisco 5921 ESR application cannot accept commands in the way it can when launched in the foreground.

When c5921i86-universalk9-ms is launched via Linux init scripts, it is highly recommended that the "-L" command line option be used. This option allows the use of the swrvcon component for use as a remote console.

As a user-space application, the c5921i86-universalk9-ms accepts several command line options, which allows you to specify information such as how many ethernet ports are to be used, and other items. Provided with the distribution is a man page, swr-application.txt, which explains command line options.

c5921i86-universalk9-ms models its network interfaces as 4-port Ethernet line cards. There can be between 1 and 5 such 4-port Ethernet line cards, and the number can be specified in the c5921i86-universalk9-ms command line. In the IOS CLI, each Ethernet port is referred to using the standard IOS bay/unit notation. The 4 ports of the first line card are referred to in the CLI as e0/0, e0/1, e0/2, and e0/3. The 4 ports of the second line card are referred to as e1/0, e1/1, e1/2, and e1/3, and similarly for subsequent line cards.

The bay/unit notation is important when the SWROPTIONS file is discussed later in this document.

Refer to the swr-application.1 man pages provided in the tar file for more information and for command line syntax.

swrvcon

swrvcon is an optional virtual console to the 5921 router software. It allows you to connect and disconnect to the router console. The swrvcon feature will work only when router software is launched using its "-L" command line option. This provides a way for you to interact with the router software when it has been launched from the Linux init scripts.

Refer to the swrvcon.1 man page provided in the tar file for more information.

swr_reload

The swr_reload application is used to reload Cisco's IOS-Linux based software router. The purpose of swr_reload is to launch and reload the software router in the event that the underlaying Linux OS develops any problems.

Refer to the swr_reload.1 man page provided in the tar file for more information.

Highlighted Features

This section introduces the highlighted features introduced in Cisco IOS Releases:

Password Recovery

This section describes how to recover the enable password and the enable secret passwords. These passwords protect access to privileged EXEC and configuration modes. The enable password can be recovered, but the enable secret password is encrypted and must be replaced with a new password. Use this procedure in order to replace the enable secret password.

Step 1 Load SWR with command line option -x. For example:

```
Linux# ./c5921i86-universalk9-ms.SPA -x
```

Step 2 The router comes up with no startup configuration and gives the option:

```
Would you like to enter the initial configuration dialog? [yes/no]:
```

Step 3 Enter **No** at the prompt. The router boots IOS. All interfaces come up shutdown with default configuration.

Step 4 Copy the startup-config to running-config.

```
Router# copy startup-config running-config
```

Step 5 Set a new enable secret password, and un-shut all interfaces that should be up.

```
Router# conf t
Router# enable password <password>
Router# int e0/0
Router# no shut
Router# int 0/1
Router# no shut
```

```
Router# end
```

Step 6 Save the configuration, shutdown the router.

```
Router# wr
Router# Control C (returns the user to the Linux command prompt)
Linux# ./c5921-swr-init.sh stop
```

Step 7 Start the SWR and confirm the configuration is Ok, and the new enable password works.

```
Linux# ./c5921-swr-init.sh start
Linux# ./swrvcon 100
Router# enable
When prompted, enter the router enable password.
Router# show run
Router# show ip int br
```

Step 8 Repeat steps 1-7 for the enable secret password.

SWR Reload Application

The swr_reload application is optionally used to reload Cisco's IOS-Linux based software router in the event of a crash or a manually initiated reload. This application completely terminates and restarts all processes related to the SWR when a reload is initiated (versus simply starting the SWR process over without actually terminating it).

The swr_reload application reloads the software router through options in the SWROPTIONS file or through command-line options. The Control-C keyboard command should be used to exit the swr_reload application.

The following commands are supported on the Cisco 5921 ESR:

- `./swr_reload`
  Execute the image specified in `launchapp` and `soft-rommon` variables in SWROPTIONS file. If these variables are not defined in the SWROPTIONS file, load and execute the default image.
- `./swr_reload <options>`
  Pass command-line options to the image specified in the SWROPTIONS file. If the image is not specified in the SWROPTIONS file, pass the command-line options to default image.
- `./swr_reload <options> <executable>` (overrides the image listed in the SWROPTIONS file)
  Execute the image specified in the command-line option and pass all other command-line options to that image.

The following command line options are supported on the Cisco 5921 ESR:

Command Line Option	Description
`-h`	Displays the help for the command.
`-d`	Displays reload debug information.
`-F`	Used to launch an image from the command line.

Note See the man page for `swr_reload` for additional information and command examples.

Software Router Options File (SWROPTIONS)

The SWROPTIONS file processes the list of specified options to the SoftWare Router (SWR). When the image is loaded without any options, it will check for the SWROPTIONS file in the HOME directory. The alternate way to use the SWROPTIONS file is through the command line by using the `-o <path>` option along with the image name, where *`<path>`* is the path to the option file.

The SWR Options File is segregated into one unlabeled section and one or more labeled sections. A labeled section is optional, and does not need to be present. The file begins with the unlabeled section at the top, and is structured as follows:

Option	Description
`debug=<boolean>`	Governs the output of certain debug information (and is not controlled nor associated with any IOS commands). Default is FALSE.
`ethernet-slots=<n>`	Number of ethernet interfaces. Default value is 2.
`launchapp`	Name of SWR image to be launched (this option is applicable only for swr_reload utility).
`linux-vlan-stripped-from-pak=TRUE`	Required if VLAN Mode = enabled on linux
`local-console-disable=<boolean>`	Governs whether the router console output is directed to its process standard output (TRUE), or to the optional virtual console (FALSE). Default is FALSE.
`node-lock-type=machine`	Default value is "machine".
`node-lock-type=storage`	This will enable the c5921 to node-lock with HDD or compact flash devices. This will not invalidate the licenses if the storage media is transferred to different devices.
`ram=<n>`	Mega-bytes of router memory. Default value is 512.
`soft-rommon`	Number of ethernet “line cards.” Each “line card” consists of 4 actual ports. Default value is 2 line cards.

Note Node Lock does not work when operating in a virtual environment. Once the c5921 is installed with Node-lock-type as machine, it should not be changed to Node-lock-type as storage. Doing so will then invalidate the license installed on c5921 on the next reboot

The labeled section or sections follow the unlabeled section. Each labeled section is introduced by the syntactic form [section-name], where [section-name] is a word naming or labeling the section. This consists of two labels [filemap] and [interface].

The [filemap] label pertains to the mapping of an IOS file system to a linux directory and is structured as follows:

Option	Description
`ios=<ios-file-system>`	The name of the associated IOS file system, and must be specified. The colon character usually associated with IOS file systems should be omitted. Default is NULL.
`linux=<linux-directory>`	The name of the associated Linux directory, and must be specified. Default is NULL.
`access=<rw | ro>`	The allowed permissions granted to IOS accesses to the mapped directory: `rw` —Read and Write (default) `ro` —Read only.

The [interface] label pertains to the association of Linux network interfaces and IOS interfaces, and is structured as follows:

Option	Description
`linux=<opt-value>`	The name of the associated Linux interface, and must be specified. Default is NULL. Example: eth0
`ios=<opt-value>`	The name of the associated IOS interface, and must be specified. Default is NULL. Example: e0/0
`type=<opt-value>`	The type of packet capture: `raw` - Raw Packet Sockets. `tap` - TAP interface.
`monitor-state=(true|false)`	Enable/disable monitoring for the Linux physical interface. Default value is TRUE.
`promiscuous={true | false}`	Governs whether or not the SWR application captures packets in promiscuous mode (in which all packets on the wire are captured). Default is TRUE.

Using Command Line Options with the SWROPTIONS File

You can specify the required options from the command line overriding the options specified in the SWROPTIONS file.

Usage:

```
<image> [options] <application id>
<image-name> -<letter-key> <sub-option-key>=<sub-value>:<sub-option-key>=<sub-value>...
<application-id>
```

The command line options are as follows:

Option	Description
`-e<n>`	Number of ethernet “line cards.” Each “line card” consists of 4 actual ports. Default value is 2 line cards.
`-d`	Governs the output of certain debug information (and is not controlled nor associated with any IOS commands). Default is FALSE.
`-h`	Display command-line help.
`-m<n>`	Mega-bytes of router memory. Default value is 512
`-L`	Disable local console. Governs whether the router console output is directed to its process standard output (TRUE), or to the optional virtual console (FALSE). Default is FALSE.
`-v`	Display program version and quit. Default is FALSE
`-x`	Ignore the startup configuration on bootup. This option is used with password recovery.
`-i<sub-valued-option>`	Specifies an association between a Linux physical interface and an IOS interface, and also specifies characteristics of that association. More than one `-i` option can appear as long as they do not conflict.

The sub-valued-options are as follows:

Option	Description
`linux=<sub-value>`	The name of the associated Linux interface, and must be specified. Default is NULL, Example: eth0
`ios=<sub-value>`	The name of the associated IOS interface, and must be specified. Default is NULL, Example: eth0/0
`type=<sub-value>`	The packet of capture: `raw` - Use raw packet sockets. `tap` - Use tap interface.
`-o<path>`	If specified, read the SWROPTIONS File at the specified path. Default is $HOME/SWROPTIONS.
`-f<file-mapping>`	Specify file mapping. Default is unix:
`monitor-state=(true|false)`	Enable/disable monitoring for the Linux physical interface. Default value is TRUE.

In the event you wish to make use of the `-f <file-mapping>` sub-valued-option, the mappings are as follows:

Option	Description
`ios=<ios_file_sys_name>`	A unique filesystem name in the IOS filename space. This file system name excludes the ":" usually associated with IOS filesystem names.

Option	Description
linux=<linux_directory>	A path in the Linux filename space.
access=<rw | ro>	Optionally specifies the allowed permissions granted to IOS accesses to the mapped directory: rw—Read and Write (default), ro—Read Only.

Note See the man page for swroptions for additional information and command examples.

TAP Interfaces for Co-Located Applications

The Cisco 5921 ESR supports use of Linux TAP interfaces to allow applications on the local computer to both access the Cisco 5921 ESR itself (for management purposes), or for routing through the Cisco 5921 ESR to destinations in the network. Setting up a TAP interface involves:

1. Defining the interface to Linux
2. Supplying an IP address (or addresses) for the Linux "end" of the TAP
3. Associating the TAP interface with a Cisco 5921 ESR interface
4. Configuring the interface in the Cisco 5921 ESR

All of the TAP installation/configuration steps assumes that you are logged in as "root".

Explanation of each of these steps follows:

Defining the TAP Interface

Commands to define the TAP are dependent on the Linux operating system used. For example, to define the TAP on CentOS, the "tunctl" package must first be installed on the CentOS machine. Installation of the tunctl package can be accomplished by entering "yum install tunctl". After installation, the command "tunctl -p -t tap0" should be executed to define a persistent TAP interface to CentOS.

After initial definition of the TAP interface via "tunctl", the TAP is brought up by the "ifconfig" command:

```
ifconfig tap0 up
```

Supplying an IP Address

The TAP interface can be envisioned as a software version of an Ethernet cable, or as a software-only 2 port switch. As such, both "ends" of the cable (or both ports of the switch) need IP addresses. In the case of Cisco 5921 ESR support, one "end" of the cable is Linux, the other is the Cisco 5921 ESR process itself.

Supply an appropriate IP address (or addresses) using the "ifconfig" command. For example, "ifconfig tap0 10.10.10.1/24" would assign the 10.10.10.1 address to the Linux "end" of the tap, with a 24-bit network mask (e.g., 255.255.255.0).

In order to make the tap0 interface persistent across Linux reboots, some additional Linux configuration needs to be applied. For example, for CentOS enter the following commands:

```
cd /etc/sysconfig/network-scripts
vi ifcfg-tap0
```

and add the following information to the file:

```
DEVICE=tap0
ONBOOT=yes
BOOTPROTO=none
IPADDR="10.10.10.1"
NETMASK="255.255.255.0"
TYPE=Tap
```

For other distributions of Linux consult your Linux documentation.

Associating the TAP interface

The association of the TAP to a Cisco 5921 ESR interface is done via the SWROPTIONS file. The form of the entry for TAP interfaces is:

```
SWROPTIONS:
     [interface]
     linux=tap0
     ios=e0/0
     type=tap
```

An example for configuring "tap" via the command line would be:

```
c5921i86-universalk9-ms -i linux=tap0:ios=eth0/0:type=tap
```

Configuring the Cisco 5921 ESR Interface

From the perspective of the Cisco 5921 ESR, the interface associated with the TAP is configured as any other Ethernet interface.

For example, the Cisco 5921 ESR configuration might contain:

```
interface e0/4
   ip address 10.10.10.2 255.255.255.0
   end
```

Note that, in this example, the Cisco 5921 ESR "end" of the TAP interface has its own address (10.10.10.2), and this address is in the same subnet as the Linux "end" of the TAP, which is 10.10.10.1.

Entering the above configuration, and starting the interface allows the Cisco 5921 ESR to communicate with the local Linux system. As a test of the Cisco 5921 ESR TAP support, it should now be possible to login to the Linux box, and ping the Cisco 5921 ESR address (in this example, 10.10.10.2) from a shell process.

Standard Linux Packages

If using the TAP interface, you will also need the following standard Linux packages:

- bridge-utils
- uml-utilities (Ubuntu)
- tunctl (CentOS and most other Distributions)

Installation

The Cisco 5921 ESR is delivered as a tar file:

c5921i86_universalk9-ms_REL-[RELEASE_LABEL].tar

with the following contents:

FAQ_C5921.txt

c5921i86-universalk9-ms.SPA

c5921i86-universalk9-ms.md5

README_C5921.txt

RELEASE_NOTES_C5921.txt

swr-application.1

swr_reload

swr_reload.1

libdyncs.so

swroptions.1

SWROPTIONS.example.txt

swrvcon

swrvcon.1

c5921-swr-init.sh

The files from the tar file need merely be extracted, using:

```
tar -xf c5921i86-universalk9-ms_REL-[RELEASE_LABEL].tar
```

Node Locking

The storage devices that hold the 5921 image can be moved from one device to another. This has ramifications for the licensing of the 5921 image. The SWR Options File mentioned earlier has a new option "**node-lock-type=storage**" which is used to enable the ability to transfer licenses.

Note Node Lock does not work when operating in a virtual environment.

There are two typical situations where you need to take action in order to preserve the system state during a hard drive swap:

Situation 1

The Linux Ethernet interface numbering will be changed on moving the hard disk from one device to another. The original numbering would be something like eth0, eth1, eth2 etc. After a move to a new device, the Ethernet numbering will be changed to something like eth3, eth4, eth5 etc.

The file "**/etc/udev/rules.d/70-persistent-net.rules**" is created by the system during boot up which will have the mac-address and Ethernet numbering. After swapping the hard drive to a different machine, the new MAC address will be added and increments the interface numbering in the above file

The solution to fixing this problem is to delete this file before swapping the hard drive into a new system. The file "/etc/udev/rules.d/70-persistent-net.rules" is created by the system on boot up which will have the mac-address and Ethernet numbering. On swapping the hard drive to a different machine, the new MAC address will be added and increments the interface numbering in the "**/etc/udev/rules.d/70-persistent-net.rules**" file.

The following are examples of what you may see with improper Ethernet interface numbering:

Before swap:

```
[root@bara-1 AUTO]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:22:4D:88:D9:BA
          inet addr:10.78.176.107  Bcast:10.78.176.127  Mask:255.255.255.192
          inet6 addr: fe80::222:4dff:fe88:d9ba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:675040 errors:0 dropped:0 overruns:0 frame:0
          TX packets:332217 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:955303216 (911.0 MiB)  TX bytes:23037014 (21.9 MiB)
          Interrupt:16 Memory:d0120000-d0140000

eth1      Link encap:Ethernet  HWaddr 00:22:4D:88:D9:C9
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:17 Memory:d0020000-d0040000
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:356 errors:0 dropped:0 overruns:0 frame:0
          TX packets:356 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27912 (27.2 KiB)  TX bytes:27912 (27.2 KiB)
[root@bara-1 AUTO]#
```

After swap:

```
eth2      Link encap:Ethernet  HWaddr 90:E2:BA:17:54:F6
          inet6 addr: fe80::92e2:baff:fe17:54f6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19997 errors:0 dropped:0 overruns:0 frame:0
          TX packets:707 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1646234 (1.5 MiB)  TX bytes:77904 (76.0 KiB)

eth3      Link encap:Ethernet  HWaddr 00:22:4D:88:DA:5A
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4348 (4.2 KiB)  TX bytes:6957 (6.7 KiB)
          Interrupt:177 Memory:d0120000-d0140000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1910 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1910 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3010538 (2.8 MiB)  TX bytes:3010538 (2.8 MiB)
[root@bara-1 AUTO]#
```

Situation 2

After swapping the hard drive and booting the device, fsck (file system check error) errors are seen on the new device. The solution is to manually run the fsck command on the new device.

Further Reading and Resources

In the tar file you will find:

swr-application.1 (man page)

swr_reload.1 (man page)

swrvcon.1 (man page)

swroptions.1 (man page)

Included are the following additional files:

FAQ_C5921.txt (Frequently Asked Questions)

README_C5921.txt (Introduction to the Cisco 5921 ESR and a description of the components.)

RELEASE_NOTES_C5921.txt (Software distribution release notes.)

Cisco IOS Configuration Guides:

http://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-15-2m-t/products-installation-and-configuration-guides-list.html

Information on the CentOS, the performance reference platform operating system, is at:

http://www.centos.org

Cisco Feature Navigator allows you to quickly find the right Cisco software release for the features you want to run on your network. You can search by feature, technology, image, platform and more.

http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp

7

Creating the Cisco 5921 ESR Reference Platform

Since the Cisco 5921 ESR can execute on a variety of different hardware platforms and Linux distributions, recreating the problems that customers are experiencing can be quite challenging. Cisco customer support cannot maintain, on site, all of the different hardware platforms used by all system integrators. Therefore, all potential problems or other issues that require Cisco customer support should first be replicated by the system integrator on a "reference platform."

Note A reference platform is defined as a mutually agreed upon production PC system that is currently available for retail sale and fully supported by its manufacturer but is not any form of development board or engineering prototype.

The system integrator will also need a USB CD-ROM drive for initial configuration of the system, i.e., installing the operation system from a "Live CD".

The reference system must be loaded with the CentOS-6.4 i386 (32-bit) Linux distribution.

Information and download of CentOS can be found at *http://www.centos.org*

8

Troubleshooting

Frequently Asked Questions

- Table 8-1 “General FAQs”
- Table 8-2 “Support FAQs”
- Table 8-3 “Configuration and Installation FAQs”

Table 8-1 General FAQs

Question	Answer
What type of product is the Cisco 5921 ESR?	The Cisco 5921 ESR is an x86 32-bit Linux application which provides Cisco IOS routing capability. It runs on the host hardware and Linux operating system provided by an Integrator.
Are there any hardware options or features that are required for the Cisco 5921 ESR application?	There are no hardware requirements. This is a Linux software application. The Integrator is responsible for providing the hardware.
When selecting the platform, what are the hardware requirements?	From the Cisco 5921 ESR perspective, the CPU must be of the Intel x86 compatible family with minimum 512 Mbyte RAM with at least two ethernet interfaces. In addition, the distribution of Linux selected may impose more requirements.
What Linux distributions are supported?	There are no Linux restrictions as the Integrator is responsible for building the system and verifying its functionality. Cisco has tested on Ubuntu, Fedora, CentOS, and OpenWRT. Cisco has also tested in desktop configurations and embedded configurations. See *http://en.wikipedia.org/wiki/Linux_distribution*
What are the baseline Linux requirements?	The selected Linux kernel must be 2.6.32 or greater. While the software application is a 32-bit application, it will work on a 64-bit Linux system provided the 32-bit libraries are installed.
What are the Cisco 5921 ESR application Linux dependencies?	Standard Linux tools can be used to display dependencies. Cisco 5921 ESR application specific shared library "libdyncs.so" is provided as part of the Cisco 5921 applicatoin package. Both Cisco 5921 ESR application and "libdyncs.so" library must be present in the same directory. "libdyncs.so" must be exported using "export LD_LIBRARY_PATH=<path_of_cisco_5921_ESR_app>" An example from Cent is shown below: `$ export LD_LIBRARY_PATH=/opt/cisco/c5921`

Question	Answer

	ldd c5921i86-universalk9-ms.SPA
	linux-gate.so.1 => (0x008af000)
	libnsl.so.1 => /lib/libnsl.so.1 (0x00621000)
	libdl.so.2 => /lib/libdl.so.2 (0x0091a000)
	libpthread.so.0 => /lib/libpthread.so.0 (0x00a03000)
	librt.so.1 => /lib/tls/librt.so.1 (0x00905000)
	libdyncs.so => /opt/cisco/c5921/libdyncs.so (0x00fd3000)
	libm.so.6 => /lib/libm.so.6 (0x0038e000)
	libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x00c8e000)
	libc.so.6 => /lib/libc.so.6 (0x00174000)
	lib/ld-linux.so.2 (0x00bcc000)
	
	$ ldd swrvcon
	linux-gate.so.1 => (0x00cfd000)
	libc.so.6 => /lib/libc.so.6 (0x004e5000)
	/lib/ld-linux.so.2 (0x00a2f000)
	
	$ ldd swr_reload
	linux-gate.so.1 => (0x00b76000)
	libpthread.so.0 => /lib/libpthread.so.0 (0x00e16000)
	libc.so.6 => /lib/libc.so.6 (0x0079a000)
	/lib/ld-linux.so.2 (0x0052f000)
	Internet search "linux ldd" for further details.

Can the Cisco 5921 ESR application run in the foreground as well as the background?	Yes, both modes are supported. The Integrator will determine which is appropriate depending upon requirements.
When the Cisco 5921 ESR application is initiated for the first time, does it go through the IOS initial configuration process?	Yes, the Integrator will be prompted to enter the initial configuration dialog: --- System Configuration Dialog --- Would you like to enter the initial configuration dialog? [yes/no]:
What is the IOS feature baseline?	The Cisco 5921 ESR application is built from the IOS GC and T train. Note that some features that are included in the feature list may not be included, or supported at this time. See the Cisco Feature Navigator at *http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp* for supported features.
In IOS terms, what is the image name of the IOS Linux application?	c5921i86-universalk9-ms

Question	Answer
What other Cisco 5921 ESR application requirements exist, if any?	• swrvcon is an optional application that provides virtual access to the IOS console (CLI) • swr_reload is an application that provides a launch option and reloads SWR if it becomes inoperable.
Are IOS features configured via CLI commands?	Yes. The IOS CLI is available from the console window or the virtual console application, swrvcon. Once the configuration has been modified it can be saved via the standard IOS commands. The configuration is saved within the Linux file system based upon the designated router ID.
My single board computer has a spare serial port. Can the Cisco 5921 ESR be used to make this port an AUX port for connection to a modem?	No, serial ports are not supported. The Cisco 5921 ESR only supports ethernet interfaces.
Is it possible to share the ethernet interfaces between Linux and the Cisco 5921 ESR application?	This is not supported. An interface should either be managed by the Cisco 5921 ESR application or by Linux, never by both. Any interface that is not assigned to the Cisco 5921 ESR application are available to Linux. Do not assign an IP address in Linux to an interface that is managed by the Cisco 5921 ESR.
My single board computer has a USB port. Can this be used in conjunction with an eToken device?	Yes, when Linux is used to manage the eToken. IOS eToken commands and functionality can not be used for Linux eTokens. Internet search "linux etoken" for further details.
My single board computer has a USB port. Can this be used this in conjunction with an eToken device?	Yes when Linux is used to manage the eToken. IOS eToken commands and functionality can not be used for Linux eTokens. Internet search "linux etoken" for further details.
When needing to install a new Cisco 5921 ESR application, is the new image or application FTPed to IOS as typically done with other IOS products?	No, as a Linux application, the Integrator would simply need to stop the application, then copy the new Cisco 5921 ESR files over the previous ones.
There are three Linux applications in the Cisco 5921 ESR distribution. When updating the system can one application be upgraded and the others remain?	No. The three applications are considered a single package and should all be installed and upgraded as a group.

Question	Answer
If the selected processor supports hardware assisted encryption, can the Cisco 5921 ESR application make use of this to increase encryption performance?	No, the Cisco 5921 ESR application only supports the software based encryption built into the application.
What kind of performance can one expect from the Cisco 5921 ESR application?	The achievable performance is dependent upon several factors. The most important will be the CPU features such as processor speed, cache, and memory. Enabled IOS features will also impact performance, i.e., encryption, access control lists, tunneling, NAT, and firewall.

Table 8-2 Support FAQs

The IOS show process command displays a long list of IOS tasks, but these tasks do not show up in the Linux top display. Why?	The Cisco 5921 ESR application runs the entire IOS image as a single Linux process.
When the software application is initiated from the command line with a new router ID parameter, the previously saved configuration is not found. Why?	The router ID parameter is used by the Cisco 5921 ESR application to uniquely manage various files. The nvram data is stored in a file based upon the router ID so when the router ID is changed, the associated nvram files is expected. Example of nvram filename for router ID 100: nvram_00100.
All of my interfaces show up as 10 Mbps ethernet even though I have 10/100/1000 interfaces in my system. Am I really limited to 10Mbit/s	No. If you are using RAW interfaces, the monitor-state option in SWROPTIONS file needs to be in TRUE state to get the interface speed setting from Linux to IOS. If you use TAP interface, the default 10 Mbps speed only will show up in IOS even though your underlying physical adapter is 100/1000 Mbps. This is the limitation for TAP in showing up as 10 Mbps default value but in reality this interface can carry more than 10 Mbps based on the underlying physical adapter.
All of my interfaces show "duplex half" even though the interfaces on my system are full duplex. Am I really limited to half duplex operation	No, you aren't limited to half duplex operation. IOS interfaces might not be mapped to any Linux interfaces and/or the monitor-state in SWROPTIONS file might not be set to True. This might be the reason that your IOS interfaces are showing as default half-duplex.
"Show interface" always shows the hardware as "AmdP2" despite the manufacturer of the adapter. Why?	"AmdP2" is the name of the driver used at the logical level by the Cisco 5921 ESR application. The underlying interface is not queried by IOS, so the actual hardware type is not displayed.
I cannot copy files as on other Cisco routers to devices like "flash:" or "disk:". When I issue "dir all" in IOS, I see "unix:", What is that?	Since there is no flash or hard drive dedicated to IOS as on Cisco Hardware Routers, by default, only a Unix file system is available to the Cisco 5921 ESR application. Therefore, "unix:" was created to allow reading/writing of files to the unix filesystem from IOS. "unix:" is mapped to the current working directory where the Cisco 5921 ESR application was launched.

Table 8-3 Configuration and Installation FAQs

Must the Cisco 5921 ESR applications be managed by root or can the files be managed in other directories?	Cisco 5921 ESR does not require root management, but it does require sudo privilege to execute when not root. Internet search "linux sudo" for further details.
From the Linux perspective, do the Linux logging facilities continue to function?	Absolutely, the Linux variable data directory, /var, remains intact. The standard Linux /var files can be reviewed to gather system information. It is also important to note that standard Linux utilities and tools continue to be available to facilitate file rotation, reporting, etc. The /var/log directory may be most useful. It is also possible to configure IOS to redirect logging and debug information to the Linux syslog on the local machine using the Tap interface or to a remote host. Internet search "linux variable log files" for further details.
Seeing TCP performance degradation while using linux driver optimizations.	GRO should be disabled. Refer to CSCvr00520 in the Cisco Bug Search Tool:

9

Command Reference

This appendix provides command reference documentation.

List of Commands

license clear

To remove a license from license storage, use the **license clear** command in privileged EXEC mode. This command does not have a **no** form.

license clear *feature-name*

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Usage Guidelines

The **license clear** command verifies that the license is valid and explicitly installed. Only explicit licenses that have been added using the **license install** command are removed.

Use the **show license** command to determine the exact name for the *feature-name*.

Examples

The following example shows how to clear the license *c5921-x86-level2*:

```
Router#license clear c5921-x86-level2
Feature: c5921-x86-level2
    1   License Type: Permanent
        License State: Active, In Use
        License Addition: Exclusive
        License Count: Non-Counted
        Comment:
        Store Index: 1
        Store Name: Primary License Storage

Are you sure you want to clear? (yes/[no]): yes
Router#
```

license install

To install a stored license file, use the **license install** command in privileged EXEC mode. This command does not have a **no** form.

license install *filesystem***:***filename*

Syntax Description

filesystem:filename	The *filesystem:filename* argument indicates the location within a device where the Cisco IOS software license is stored.

Command Default

No license is installed.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Usage Guidelines

You will be prompted to accept the Cisco licensing end user license agreement (EULA) when installing the license.

Examples

The following example shows how to install a license with the name `unix:9FKC6PI4VIA_20120717083214630.lic` in the unix file system:

```
Router#license install unix:9FKC6PI4VIA_20120717083214630.lic
Installing licenses from "unix:9FKC6PI4VIA_20120717083214630.lic"
Installing...Feature:c5921-x86-level3...Successful:Supported
1/1 licenses were successfully installed
0/1 licenses were existing licenses
0/1 licenses were failed to install

*Jul 17 18:01:43.016: %LICENSE_C5920-6-LICENSE_ACTIVATED: Installed license for feature
c5921-x86-level3 now in use.
*Jul 17 18:01:43.047: %LICENSE-6-INSTALL: Feature c5921-x86-level3 1.0 was installed in
this device. UDI=CISCO5921-K9:9FKC6PI4VIA; StoreIndex=0:Primary License Storage

*Jul 17 18:01:45.017: %LINK-3-UPDOWN: Interface Ethernet0/0, changed state to up
*Jul 17 18:01:45.017: %LINK-3-UPDOWN: Interface VoIP-Null0, changed state to up
*Jul 17 18:01:46.018: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed
state to up
*Jul 17 18:01:46.018: %LINEPROTO-5-UPDOWN: Line protocol on Interface VoIP-Null0, changed
state to up
Router#
```

show license

To display information about a Cisco IOS software license, use the **show license** command in either user EXEC or privileged EXEC mode.

show license EULA | **agent** {**counters** | **session**} | **all** | **detail** *feature-name* | **feature** | **counters** | **session** | **statistics** | **status** }

Syntax Description

EULA	Shows the software license End User License Agreement.
agent *{counters | session}*	Shows information about a license agent including counters and session. This keyword is available only in privileged EXEC mode.
all	Shows detailed information on installed licenses.
detail *feature-name*	Shows license feature detail information.
feature	Shows a list of licensed features available in an image.
counters	Shows statistics counters for the license agent.
session	Shows session information related to a license agent.
statistics	Shows license statistic information.
status	Shows supported license types, supported license operations, and device status.

Command Modes

User EXEC

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Usage Guidelines

Use this command to display license information and help with troubleshooting issues related to Cisco IOS software licenses.

This command also displays the features that are available but not licensed to execute. The command output is grouped according to how the features are stored in license storage.

Examples

The following example shows output for the **show license** command:

```
Router#show license
Index 1 Feature: c5921-x86-evaluation
Index 2 Feature: c5921-x86-level1
Index 3 Feature: c5921-x86-level2
Index 4 Feature: c5921-x86-level3
Index 5 Feature: c5921-x86-level4
Index 6 Feature: c5921-x86-level5
Index 7 Feature: c5921-x86-level6
```

```
        Period left: Life time
        License Type: Permanent
        License State: Active, In Use
        License Count: Non-Counted
        License Priority: Medium
Router#
```

show license file

To display all the installed Cisco IOS license files, use the **show license file** command in privileged EXEC mode.

show license file

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Examples

The following example shows output for the **show license file** command:

```
Router#show license file
License Store: Primary License Storage
  Store Index: 0
    License: 11 c5921-x86-level2 1.0 LONG NORMAL STANDALONE EXCL INFINITE_KEYS
              INFINITE_KEYS NEVER NEVER NiL SLM_CODE CL_ND_LCK NiL *1YZ6XX23GP
             7EZBU400 NiL NiL NiL 5_MINS <UDI><PID>CISCO5921-K9</PID><SN>9FKC6
             PI4VIA</SN></UDI> cNLHxz4sLXyD2mpPrjiBLgMnDZuHTdFrpd1tsKyXKOLMNWb
             IUyXcJj3B7Sm7CiH5feiqm8lGflckHxtlrl3ED0VjDFL7995uVPzNWrYawVpDfvXE
             Uq33rjfCbdGPJ49iHVVW$<WLC>AQEBIf8B//+bOa/pH0fJvDEkKA7vF4b2rmpacnz
             qwBWiX3Yc+NCc7p40fiKqLMFI4fwuHFTm7zNHebPkGlARtYd1UQO7GJ3KnufZ9oZ6
             JdFniDf5HrQ8DrXdpCz5RgZE+y8fbN200xiXA5cB3fwcJqoPIFZm2HmD1qFfsyTAz
             uio66t6Xk5y8xo1lbVhvoh/FZfy5iRY3oE=</WLC>
    Comment:
       Hash: cmaFOn/1HXsNtUUVzHuI0eS9S48=
Router#
```

show license udi

To display Unique Device Identifier (UDI) information for the router, use the **show license udi** command in Privileged EXEC mode.

show license udi [**history**]

Syntax Description

history	(Optional) keyword to display the history of UDIs associated with the router.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Usage Guidelines

The Cisco 5921 ESR uses virtual UDIs because it is not a hardware-based platforms with a fixed UDI.

The **show license udi** command displays the complete UDI for the Cisco 5921 ESR. The UDI comprises a generated serial number, and a product ID number. You use the UDI in the license registration tool to acquire a license.

The **show license udi history** command displays all invalidated UDIs.

Examples

The following examples display output for the **show license udi** command:

```
Router# show license udi
Device#    PID                SN                 UDI
---------------------------------------------------------------------------------
*0         CISCO5921-K9       9I8DGGX8Z2G        CISCO5921-K9:9I8DGGX8Z2G
```

The following example displays the output for the **show license udi history** command:

```
router# show license udi history
Device#    PID                SN                 UDI
---------------------------------------------------------------------------------
*0         CISCO5921-K9       9I8DGGX8Z2G        CISCO5921-K9:9I8DGGX8Z2G

Invalidated UDIs:
------------------------------------------------
1. CISCO5921-K9 : 9UD82385M5P
```

Table 9-1 describes the significant fields shown in the examples.

Table 9-1 Field Descriptions for the show license udi Command Output

Field	Description
PID	Product ID number
SN	Generated serial number

Field	Description
UDI	Unique Device Identifier
Invalidated UDIs	Unique Device Identifiers that have been invalidated due to new UDI generation.

show platform software enforcement

This command is used to determine how much outbound bandwidth is currently being used by the system and to determine the high water mark for bandwidth usage. show platform software license.

show platform software enforcement

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Usage Guidelines

Use this command for detailed information about bandwidth usage. The command's output will also show the user how close the current bandwidth is in relation to the installed license. Use the "clear" form of this command to clear the statistics.

Examples

The following example displays information about a license.

```
Router#show platform software enforcement

License enforcement statistics:
Switched:300930 packets, 37299710 bytes, rate 198000 bps
Dropped:0 packets, 0 bytes, rate 0 bps
Total:300930 packets, 37299710 bytes, rate 198000 bps
Highest total rate: 199000 bps (02:36:44.905 EST Mon Apr 12 2010) Notification if traffic
load reaches 100% of enforced rate
Statistics last cleared: never
```

show platform software interface-status

This command is used to determine the physical interface status as reported to the Cisco 5921 ESR by Linux. This command only has meaning if "monitor-status" is enabled on a given interface.

show platform software interface-status

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Usage Guidelines

Use this command to determine physical interface status as report by Linux.

Examples

```
Router#show platform software interface-status

Status of Monitored Interfaces
------------------------------
Linux Interface   IOS  Running  Carrier  MTU    Speed     Duplex   Autoneg
eth1              0/1  Yes      Yes      1500   1 Gbps    full     Yes
eth0              0/0  Yes      Yes      1500   1 Gbps    full     Yes
```

show platform software license

To display Cisco 5921 ESR specific information use the **show platform software license** command in the privileged EXEC mode.

show platform software license

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(4)GC	This command was introduced.

Usage Guidelines

You can use this command to determine which feature licenses are required for your system, and verify that an installed license is activated. The command displays the following information:

- State of packet forwarding (enabled of disabled) based on license state
- Current enforcement forwarding rate
- Unique Device Identifier (UDI) generated for your Cisco 5921 ESR instance
- List of the supported licenses, based on the hardware capabilities rating of your hardware

Examples

The following example displays information about a license.

```
Router#show platform software license
Packet forwarding: Enabled

Current enforcement forwarding rate: 200 Mbps

Unique Device Identifier: CISCO5921-K9:92K4304QKB3

License features supported:
       Feature              Rate     Status
  ---------------------   ---------  ------
  c5921-x86-default          8 Kbps    -
  c5921-x86-evaluation      50 Mbps    -
  c5921-x86-level0           5 Mbps    -
  c5921-x86-level1          10 Mbps    -
  c5921-x86-level2          25 Mbps    -
  c5921-x86-level3          50 Mbps    -
  c5921-x86-level4         100 Mbps    -
  c5921-x86-level5         200 Mbps  In Use
  c5921-x86-level6         500 Mbps    -

Router#
```

10

Glossary Terms and Acronyms

This appendix provides terms and acronyms used in this guide and their description or definition:

- Table 10-1, “Glossary”
- Table 10-2, “Acronyms and Abbreviations”

Table 10-1 Glossary

Glossary Term	Description
Cisco 5921 ESR	Special version of IOS built into a Linux application
customer	End user of the system provided by the Integrator
integrator	Engineering personnel responsible for integrating the Cisco 5921 ESR application with Linux and other system collateral
Router ID	ID of the c5921i86-universalk9-ms process
swrvcon	Linux Application that provides a virtual console to access Cisco IOS
TAP	As in network tap, simulates a link layer device, and operating with layer 2 packets
tar	File format used commonly to collect many files into one larger file for distribution or archiving
tar file	Collection of files in the tar file format. Also known as a tarball.
Unique Device Identifier	29 character value composed of an 18 character product identifier, and an 11 character serial number

Table 10-2 Acronyms and Abbreviations

Acronym	Definition
ASCII	American Standard Code for Information Interchange
aux	auxiliary
CCP	Cisco Configuration Professional
CentOS	Community Enterprise Operating System
CLI	Command Line Interface
CME	Communications Manager Express
CPU	Central Processing Unit
CSL	Cisco Software Licensing
CSM	Cisco Security Manager
DHCP	Dynamic Host Configuration Protocol

Table 10-2 Acronyms and Abbreviations

Acronym	Definition
DLEP	Dynamic Link Exchange Protocol
DMVPN	Dynamic Multipoint Virtual Private Network
EFT	Early Field Trial
EIGRP	Enhanced Interior Gateway Routing Protocol
ESR	Embedded Services Router
EULA	End User License Agreement
FAQ	Frequently Asked Questions
FTP	File Transfer Protocol
GETVPN	Group Encrypted Transport VPN
GRE	Generic Routing Encapsulation
IOS	Internetwork Operating System
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
L2TP	Layer 2 Tunneling Protocol
LMS	LAN Management Solution
MIB	Management Information Base
OID	Object Identifiers
OSPF	Open Shortest Path First
PAK	Product Authorization Key
QoS	Quality of Service
RAR	Radio Aware Routing
SNMP	Simple Network Management Protocol
SRST	Survivable Remote Site Telephony
SSH	Secure Shell
swr	software router
syslog	system log
TFTP	Trivial File Transfer Protocol
man	manual
NVRAM	Nonvolatile Random Access Memory
UDI	Unique Device Identifier
VLAN	Virtual Local Area Network