K-Series to S-Series Differences

The S-series servers are supported by G-series RVUs, which introduce the following enhancements:

- Graphical user interface with extensive online help
- Online configuration
- Integrated software products

See a comparison diagram of K-series and S-series servers.

S-Series Software Package and Management Tools

The following software is included with the S-series servers. Click on the links for general information about how you can use these software packages.

- Gxx version of the NonStop Kernel operating system that supports ServerNet technology
- Guardian or Open System Services (OSS)
- The Distributed Systems Management/Software Configuration Manager (DSM/SCM)
- OSM or TSM software
- Subsystem Control Facility (SCF), which includes several subsystems
- Factory Default IP Addresses for S-Series Servers

K-Series Software Obsolete in S-Series	Function	S-Series Equivalent
Communications Management Interface (CMI)	Configures new device, starts new device	Subsystem Control Facility (SCF) for ATP6100, CP6100, and EnvoyACP/XF
Communications Subsystem Manager (CSM)	Configures processes for communications subsystems	Concentrator Manager (ConMgr)
Configuration Utility Program (COUP)	Adds controller, adds device, configures system variables and generic processes online	Subsystem Control Facility (SCF)
Expand-over-LAN	Networks NonStop systems	Expand-over-IP

Install	Installs new operating system or SPR and configures software	Distributed Systems Management/Software Configuration Manager (DSM/SCM)
Multilan Server	Accesses NonStop resources from LAN workstations	NetBIOS (NBT)
		NetBIOS over NonStop IPX/SPX (NBX) product
Peripheral Utility Program (PUP)	Manages storage devices, including disks and tapes	Subsystem Control Facility (SCF)
Remote Console Process (RCP)	Provides a remote system console	OSM or TSM software
Remote Maintenance Interface (RMI)	Loads processors, performs system coldloads	OSM or TSM software and system console
Remote Operations Facility (ROF)	Establishes a connection to a remote system	OSM or TSM software
SNAX/CDF	Configures processes for communications subsystems	SNAX/APN
Syshealth	Provides troubleshooting, maintenance, and service	OSM or TSM software
HP Tandem LAN Access Method (TLAM)	Configures LAN access connections and facilities	ServerNet LAN Systems Access (SLSA) via SCF
HP Tandem Maintenance and Diagnostic System (TMDS)	Provides online diagnostics to analyze the status of hardware devices	OSM or TSM software
TAPECOM	Performs routine tape and tape-drive management operations	MEDIACOM

For a table of K-series to S-series migration-related information and tasks and the

documentation that supports them, see Sources of Information.

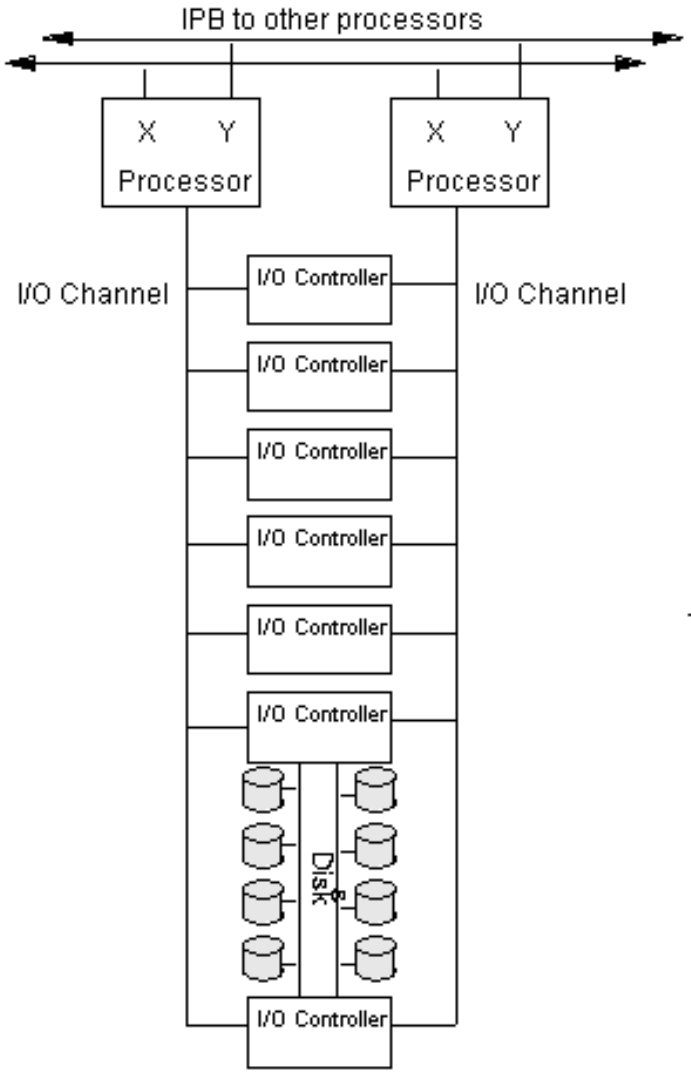
Note: G-series RVUs support S-series servers and D-series RVUs support NonStop K-series servers.

Related topics:

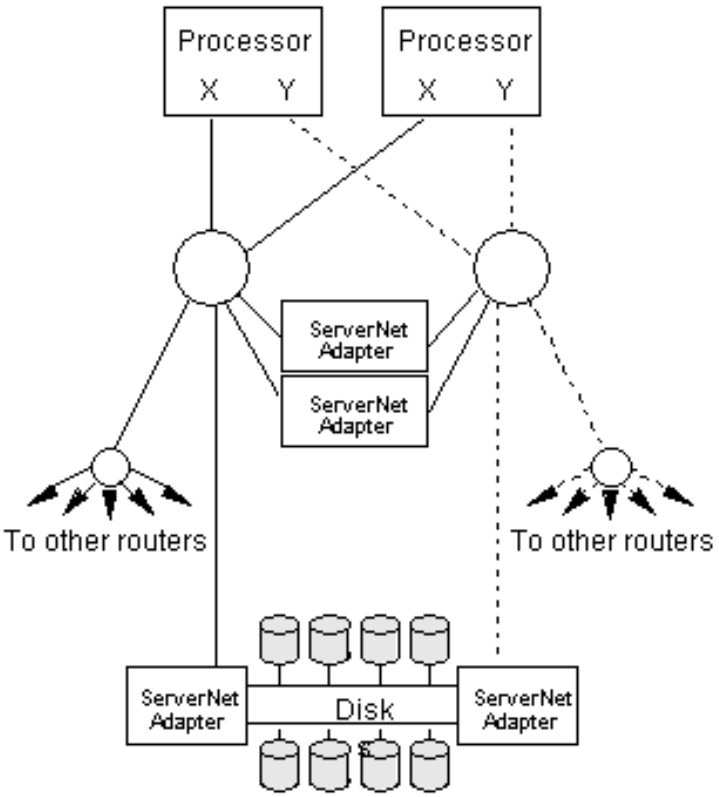
- K-Series to S-Series Migration Considerations
- Recommended Software Product Revisions (SPRs)
- Fallback Considerations

Comparison Diagram of K-Series and S-Series Servers

Map: K-Series to S-Series Differences.



Himalaya K-Series Servers



ServerNet Servers

The Distributed Systems Management/Software Configuration Manager (DSM/SCM)

Map: K-Series to S-Series Differences.

The Distributed Systems Management/Software Configuration Manager (DSM/SCM) provides tools for centralized planning, management, and installation of software on distributed (target) systems.

 **Caution**: DSM/SCM is preconfigured. Exercise care when changing $AUDIT and $DSMSCM disk names to prevent DSM/SCM from becoming unusable.

In your configuration process, you use DSM/SCM to plan new software configurations. Using its graphical user interface, you can select specific products from software inputs to create a new software configuration, based on a previous configuration, in which changed products are replaced by the new versions.

DSM/SCM runs on a central (host) system and performs the tasks of receiving, archiving, configuring, and packaging software for target sites. It also runs on each target system, where its primary function is to apply the software received from the central site. DSM/SCM:

- Activates new software configurations with minimal interruption of running applications.
- Provides default operator instructions with every software configuration placed on target systems. You can edit these instructions with site-specific information.
- Produces SQLCI reports that provide detailed information about the software configurations in the DSM/SCM environment. You can modify these reports and create new reports tailored to site specifications.
- Allows a site to run up to six software configurations on a single system, such as a production configuration and a test configuration.
- Manages software on multiple remote target systems.
- Enables you to scan through or print a report of softdocs to find dependencies for SPRs.

For further information on how to install and use DSM/SCM, see the DSM/SCM User's Guide.

OSM and TSM Packages

Map: K-Series to S-Series Differences.

The OSM and TSM packages provide troubleshooting, maintenance, and service tools. Both packages require that you use the HP NonStop System Console.

Beginning with the G06.21 RVU, OSM can be used instead of TSM to perform system management and monitoring tasks. Also beginning with the G06.21 RVU, support for new products and new functions are provided by OSM only.

Both the OSM and TSM packages allow you to:

- Start the NonStop S-series server (also known as system load or cold load).
- Display logical and physical views of the system configuration.
- Load new system configurations.
- Monitor system resources, both local and remote.
- Test disk drives and stop a disk drive.
- Browse event logs and send problem information to a service provider.

Also, for the products supported by each package, OSM and TSM allow you to:

- Identify system components and their status.
- Check the current status of system components.
- Test and maintain specified customer-replaceable units (CRUs) and field-replacement units (FRUs).
- Update firmware.
- Determine if hardware components need servicing.

For more information about the OSM package, see:

- OSM online help
- OSM User's Guide
- OSM Migration Guide

For more information about the TSM package, see:

- TSM Configuration Guide
- TSM Online User Guide
- TSM online help

NonStop System Console

Map: K-Series to S-Series Differences.

The HP NonStop system console provides a preconfigured workstation for the purpose of control, status, monitoring, and remote access of one or more S-series systems.

The system console provides OSM or TSM user interface applications for system maintenance and serviceability. User interfaces are also provided for system operator functions and for retrieving online documents.

For fault tolerance, two system consoles are recommended per NonStop S-series system site; one system console can provide console functions for more than one NonStop S-series system, and one system console must be configured per subnet to provide dial-in capabilities from a support center.

Although the applications that are hosted on the system console can be hosted on a variety of workstations from various suppliers, a specific configuration is provided to ensure a working environment for the critical functions. From a configuration and support standpoint, it is desirable that a well-known configuration be used.

Subsystem Control Facility (SCF)

Map: K-Series to S-Series Differences.

The Subsystem Control Facility (SCF) configures system, peripheral, and communication devices and subdevices and also configures processes and several system variables.

SCF configures and manages over 40 subsystems. Several of these subsystems use the system configuration database.

While loading the system using OSM or TSM, your system configuration is defined using the prebuilt SCF command files (see Initial Configuration of an HP NonStop S-Series Server). Most system configuration can be done online without stopping the NonStop Kernel operating system.

Configuration changes made online using SCF take effect as soon as the object (such as a process, disk, or data communications line) is started using the SCF START command. For information on saving new configurations, or more information about SCF and its supported subsystems, refer to one of the following manuals:

- For G06.17 and later: SCF Reference Manual for G-Series RVUs
- For G06.16 and earlier: SCF Reference Manual for G-Series Releases

Initial Configuration of an HP NonStop S-Series Server

Map: K-Series to S-Series Differences.

For G-series RVUs, many configuration tasks are completed by HP before the NonStop S-series server arrives at your site. HP provides you with the initial system configuration required to load the NonStop Kernel operating system, including:

- The system configuration database, $SYSTEM.ZSYSCONF.CONFIG, which contains configuration information received from the SCF subsystems about the placement and characteristics of all system components. The SYSnn.OSCONFIG file, which is built by SYSGENR for pre-NonStop S-series servers, still exists but contains only a few records.
- The primary Event Management Service (EMS) collector ($0), the central collection point for event messages, which record significant occurrences in the subsystem environment.
- Other processes that support the NonStop S-series server.

The contents of the $SYSTEM.ZSYSCONF.CONFIG file are described in the NonStop S-Series Planning and Configuration Guide. This initial CONFIG file is also saved on your system as the ZSYSCONF.CONF0000 file, which was built from the SCF0000 file. For sample displays from the CONF0000 file, see the SCF Reference Manual for G-Series RVUs (for G06.16 and earlier, see the SCF Reference Manual for G-Series Releases).

For the configuration tasks:

- In G06.21 and later, see the NonStop S-Series Hardware Installation and Fastpath Guide.
- In G06.20, see the Nonstop S-Series Hardware Installation Guide.
- In G06.19 and earlier, see the NonStop S-Series Planning and Configuration Guide.

The standard software configuration of a NonStop S-series server, as shipped by HP, consists of:

- Disk Volumes and Subvolumes
  - $SYSTEM Volume - contains files unique to the system, communications files, the operating system, SCF files, startup files, software product configuration files, and configuration files.
  - $DSMSCM Volume - contains Pathway control and log files, the NonStop SQL catalog, CIIN and CONFTEXT files, and DSM/SCM host and target

files.
    - $AUDIT Volume - contains DSM/SCM archive files and TMF audit trail files.
- Software Products and Documentation
    - Manufacturing installs customer-ordered software on NonStop S-series servers using DSM/SCM. To view the software products and documentation installed, use the DSM/SCM Planner Interface, as described in the DSM/SCM User's Guide.
    - Includes RVU README files, Softdocs, software in the archive, and information about target software configurations.
- Startup Files
    - CIIN File - at system start, reloads the remaining processors and starts the system console TACL process, accessed from the system console.
    - SCF0000 File - contains SCF commands that add and start key system processes and devices. SCF then saves this configuration, along with the base configuration, in the current SCF configuration file.
- STARTCOM and STARTSCF Files
    - For S-series servers with Ethernet 4 ServerNet adapters (E4SAs) and ServerNet Wide Area Network (SWAN) concentrators, starts TCP/IP processes, transfers control to the SCFSTART file (which configures the TCP/IP processes), and starts the SWAN, LISTNER, and TELSERV processes.

For details:

- In G06.21 and later, see the NonStop S-Series Hardware Installation and FastPath Guide.
- In G06.20 and earlier, see Appendix A in the NonStop S-Series FastPath Guide.

Example: SCF0000 File for G06

This example is based on a four processor NonStop S-series server with one I/O enclosure and one ServerNet wide area network (SWAN) concentrator.

```
================================================================================
== S7000/S70000 Configuration File
==
== This file contains the necessary SCF commands to configure this
== system as delivered from HP Manufacturing.
==
==
================================================================================

================================================================================
== General commands.
================================================================================

LOG SCFLOG
ALLOW ALL ERRORS

================================================================================
================================================================================

ASSUME PROCESS $ZZKRN

================================================================================
== QIO
================================================================================

ADD PROCESS #QIOMON, AUTORESTART 10, CPU ALL,
DEFAULTVOL $SYSTEM.SYSTEM, &
HOMETERM $ZHOME, OUTFILE $ZHOME, NAME $ZM, PRIORITY 199, &
PROGRAM $SYSTEM.SYSTEM.QIOMON, STARTMODE KERNEL

================================================================================
== LAN Manager
================================================================================

ADD PROCESS #ZZLAN, AUTORESTART 10, PRIMARYCPU 0, BACKUPCPU 1, &
DEFAULTVOL $SYSTEM.SYSTEM, HOMETERM $ZHOME, OUTFILE $ZHOME, &
NAME $ZZLAN, PRIORITY 180, PROGRAM $SYSTEM.SYSTEM.LANMAN, &
STARTMODE KERNEL, STARTUPMSG"<BCKP-CPU>"

================================================================================
== WAN Manager
================================================================================

ADD PROCESS #ZZWAN, NAME $ZZWAN, PRIORITY 180, AUTORESTART 10, &
PROGRAM $SYSTEM.SYSTEM.WANMGR, PRIMARYCPU 0, BACKUPCPU 1, &
TYPE SUBSYSTEM-MANAGER, STARTMODE KERNEL, HOMETERM $ZHOME, &
OUTFILE $ZHOME, STARTUPMSG"<BCKP-CPU>"

================================================================================
== SCP
================================================================================

ADD PROCESS #SCP, NAME $ZNET, PRIORITY 175, AUTORESTART 10, &
PROGRAM $SYSTEM.SYSTEM.SCP, PRIMARYCPU 0, BACKUPCPU 1, TYPE OTHER, &
STARTMODE SYSTEM, HOMETERM $ZHOME, &
```

```
OUTFILE $ZHOME, STARTUPMSG "<BCKP-CPU>; AUTOSTOP -1"

================================================================================
== CLCI TACL
================================================================================

ADD PROCESS #CLCI-TACL, NAME $CLCI, PRIORITY 199, AUTORESTART 10, &
PROGRAM $SYSTEM.SYSTEM.TACL, PRIMARYCPU 0, BACKUPCPU 1, TYPE OTHER, &
STARTMODE MANUAL, HOMETERM $YMIOP.#CLCI, INFILE $YMIOP.#CLCI, &
OUTFILE $YMIOP.#CLCI, STARTUPMSG "<BCKP-CPU>"

== Start QIO, WAN Manager, etc.
================================================================================
START #QIOMON
START #ZZLAN
START #ZZWAN
START #SCP
DELAY 60

================================================================================ ==

== This section adds a MIOE adapters into slots 1.1.50 and 1.1.55.
== TSM will configure the TCP/IP processes for these shared adapters.
==
================================================================================

ASSUME SUBSYS $ZZLAN

ADD ADAPTER MIOE0, TYPE MIOE, LOCATION (1,1,50), ACCESSLIST (0,1)
START ADAPTER MIOE0, SUB ALL
ADD LIF LANX, PIF MIOE0.0.A
START LIF LANX

ADD ADAPTER MIOE1, TYPE MIOE, LOCATION (1,1,55), ACCESSLIST (1,0)
START ADAPTER MIOE1, SUB ALL
ADD LIF LANY, PIF MIOE1.0.A
START LIF LANY

================================================================================
==
== The following section was taken from TSMAUTO's ADDTOSCF & ADDTCPIP
==
================================================================================
== This section adds the TSM processes that configure and start up
== TCP IP's for the LANX and LANY LIF's. These are TACL's that will
== use IN files INIT0 and INIT1 to do the deed.

ADD PROCESS $ZZKRN.TSM-ZTCP0, &
AUTORESTART 0, &
DEFAULTVOL $SYSTEM.ZSERVICE, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $TSMM0, &
CPU 0, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.TACL, &
INFILE $SYSTEM.ZTSM.INIT0, &
STARTMODE SYSTEM, &
STARTUPMSG "01", &
USERID SUPER.SUPER
```

```
START PROCESS $ZZKRN.TSM-ZTCP0

ADD PROCESS $ZZKRN.TSM-ZTCP1, &
AUTORESTART 0, &
DEFAULTVOL $SYSTEM.ZSERVICE, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $TSMM1, &
CPU 1, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.TACL, &
INFILE $SYSTEM.ZTSM.INIT1, &
STARTMODE SYSTEM, &
STARTUPMSG "0", &
USERID SUPER.SUPER

START PROCESS $ZZKRN.TSM-ZTCP1

== This section adds the TSM processes themselves that need to be
== started at boot time.

== ADD $ZLOG to SCF database and issue command to start process
ADD PROCESS $ZZKRN.TSM-ZLOG, &
AUTORESTART 5, &
DEFAULTVOL $SYSTEM.ZSERVICE, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $ZLOG, &
PRIMARYCPU 0, &
BACKUPCPU 1, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.EMSACOLL, &
STARTMODE SYSTEM, &
STARTUPMSG "LOGSUBVOL $SYSTEM.ZSERVICE,SECURITY NNOO,TMDS,LOGPREFIX S, &
SUPPRESS (N=25, T1=100, T2=120, T3=300, S=6),BACKUP <BCKP-CPU>"

START PROCESS $ZZKRN.TSM-ZLOG

== Add $ZSPE process to SCF database and issue start command

ADD PROCESS $ZZKRN.TSM-SP-EVENT, &
AUTORESTART 5, &
CPU FIRST, &
DEFAULTVOL $SYSTEM.ZSERVICE, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $ZSPE, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.ZSPE, &
STARTMODE SYSTEM

START PROCESS $ZZKRN.TSM-SP-EVENT

== Add $ZTSMS snmp process to SCF database and issue start command
== startup paramets are network name and subagent timeout interval

ADD PROCESS $ZZKRN.TSM-SNMP, &
AUTORESTART 5, &
CPU FIRST, &
DEFAULTVOL $SYSTEM.ZSERVICE, &
HIGHPIN ON, &
```

```
HOMETERM $ZHOME, &
NAME $ZTSMS, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.SNMPAGT, &
OUTFILE $ZHOME, &
STARTUPMSG "TCPIP^PROCESS^NAME $ZTCP0, SUBAGENT^TIMEOUT 12", &
STARTMODE SYSTEM

START PROCESS $ZZKRN.TSM-SNMP

== Add $ZTSM tsm process to SCF database and issue start command

ADD PROCESS $ZZKRN.TSM-SRM, &
AUTORESTART 5, &
CPU FIRST, &
DEFAULTVOL $SYSTEM.ZSERVICE, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $ZTSM, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.SRM, &
OUTFILE $ZHOME, &
STARTMODE APPLICATION

START PROCESS $ZZKRN.TSM-SRM

== Add $ZCVP0 tsm/cev process to SCF database and issue start command

ADD PROCESS $ZZKRN.CEV-SERVER-MANAGER-P0, &
AUTORESTART 5, &
CPU FIRST, &
DEFAULTVOL $SYSTEM.ZCEV, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $ZCVP0, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.CEVSMX, &
OUTFILE $ZHOME, &
STARTMODE APPLICATION, &
STARTUPMSG "TCPIP $ZTCP0"

START PROCESS $ZZKRN.CEV-SERVER-MANAGER-P0

== Add $ZCVP1 tsm/cev process to SCF database and issue start command
ADD PROCESS $ZZKRN.CEV-SERVER-MANAGER-P1, &
AUTORESTART 5, &
CPU FIRST, &
DEFAULTVOL $SYSTEM.ZCEV, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $ZCVP1, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.CEVSMX, &
OUTFILE $ZHOME, &
STARTMODE APPLICATION, &
STARTUPMSG "TCPIP $ZTCP1"

START PROCESS $ZZKRN.CEV-SERVER-MANAGER-P1

== Add the Routing Distributor ($ZRD9) to SCF database and issue start command
ADD PROCESS $ZZKRN.ROUTING-DIST, &
```

```
AUTORESTART 0, &
CPU FIRST, &
DEFAULTVOL $SYSTEM.ZSERVICE, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $TSMRD, &
PRIORITY 150, &
PROGRAM $SYSTEM.SYSTEM.TACL, &
INFILE $SYSTEM.ZTSM.INITRD, &
STARTMODE APPLICATION, &
TYPE OTHER, &
USERID SUPER.SUPER

START PROCESS $ZZKRN.ROUTING-DIST

== Configuring SNMP for the Primary HSSC TSM clients

ASSUME PROCESS $ZTSMS
ADD PROFILE $ZTSMS.#TSMPRI, HOSTADDR "192.231.36.1", ACCESS READWRITE, &
COMMUNITY "tsm192.231.36.1"
START PROFILE $ZTSMS.#TSMPRI

ADD TRAPDEST $ZTSMS.#TSMPRI, HOSTADDR "192.231.36.1", NETWORK $ZTCP0, &
COMMUNITY "tsm192.231.36.1"
START TRAPDEST $ZTSMS.#TSMPRI

ADD ENDPOINT $ZTSMS.#TSMEND, NETWORK $ZTCP1
START ENDPOINT $ZTSMS.#TSMEND

ADD TRAPDEST $ZTSMS.#TSMPRIB, HOSTADDR "192.231.36.1", NETWORK $ZTCP1,&
COMMUNITY "tsm192.231.36.1"
START TRAPDEST $ZTSMS.#TSMPRIB


==========================================================================
== Storage Devices
==========================================================================

ASSUME SENDTO STORAGE
==**********************************************************************
==
== Group 1
==
==**********************************************************************
ADD DISK $D0101, PRIMARYLOCATION (01,1,1), MIRRORLOCATION (01,1,2)
ADD DISK $D0103, PRIMARYLOCATION (01,1,3), MIRRORLOCATION (01,1,4)
ADD DISK $D0105, PRIMARYLOCATION (01,1,5), MIRRORLOCATION (01,1,6)
ADD DISK $D0107, PRIMARYLOCATION (01,1,7), MIRRORLOCATION (01,1,8)
ADD DISK $DSMSCM, PRIMARYLOCATION (01,1,13), MIRRORLOCATION (01,1,14)
ADD DISK $AUDIT, PRIMARYLOCATION (01,1,15), MIRRORLOCATION (01,1,16)
ADD DISK $D0117, PRIMARYLOCATION (01,1,17), MIRRORLOCATION (01,1,18)
==**********************************************************************
==
== Group 2
==
==**********************************************************************
ADD DISK $D0201, PRIMARYLOCATION (02,1,1), MIRRORLOCATION (02,1,2)
ADD DISK $D0203, PRIMARYLOCATION (02,1,3), MIRRORLOCATION (02,1,4)
ADD DISK $D0205, PRIMARYLOCATION (02,1,5), MIRRORLOCATION (02,1,6)
ADD DISK $D0207, PRIMARYLOCATION (02,1,7), MIRRORLOCATION (02,1,8)
ADD DISK $D0211, PRIMARYLOCATION (02,1,11), MIRRORLOCATION (02,1,12)
```

```
ADD DISK $D0213, PRIMARYLOCATION (02,1,13), MIRRORLOCATION (02,1,14)
ADD DISK $D0215, PRIMARYLOCATION (02,1,15), MIRRORLOCATION (02,1,16)
ADD DISK $D0217, PRIMARYLOCATION (02,1,17), MIRRORLOCATION (02,1,18)
==*****************************************************************
==
== Group 11
==
==*****************************************************************
ADD DISK $D1101, PRIMARYLOCATION (11,1,1), MIRRORLOCATION (11,1,2)
ADD DISK $D1103, PRIMARYLOCATION (11,1,3), MIRRORLOCATION (11,1,4)
ADD DISK $D1105, PRIMARYLOCATION (11,1,5), MIRRORLOCATION (11,1,6)
ADD DISK $D1107, PRIMARYLOCATION (11,1,7), MIRRORLOCATION (11,1,8)
ADD DISK $D1111, PRIMARYLOCATION (11,1,11), MIRRORLOCATION (11,1,12)
ADD DISK $D1113, PRIMARYLOCATION (11,1,13), MIRRORLOCATION (11,1,14)
ADD DISK $D1115, PRIMARYLOCATION (11,1,15), MIRRORLOCATION (11,1,16)
ADD DISK $D1117, PRIMARYLOCATION (11,1,17), MIRRORLOCATION (11,1,18)
==*****************************************************************
START DISK $DSMSCM
START DISK $AUDIT
START DISK $D*
==*****************************************************************
== Add Tape drive(s)
ADD TAPE $T0150 ,LOCATION (01,1,50)
ADD TAPE $T0155 ,LOCATION (01,1,55)
ADD TAPE $T0250 ,LOCATION (02,1,50)
ADD TAPE $T0255 ,LOCATION (02,1,55)
ADD TAPE $T1150 ,LOCATION (11,1,50)
ADD TAPE $T1155 ,LOCATION (11,1,55)
START TAPE $*

================================================================================
==
== This section adds 2 E4SA (4-Port Ethernet) adapters as follows:
==
== SLOT    NAME   CPUS    LIF   TCPIP     IP Address
== ------  -----  ------- ----- ------ ---------------
== 1.1.53  E0153  (00,01) L018  $ZB018 192.231.036.099
==                        L019  $ZB019 192.231.036.100
==                        L01A  $ZB01A 192.168.002.099
==                        L01B  $ZB01B 192.168.002.100
==
== 1.1.54  E0154  (01,00) L01C  $ZB01C 192.168.003.099
==                        L01D  $ZB01D 192.168.003.100
==                        L01E  $ZB01E 192.168.004.099
==                        L01F  $ZB01F 192.168.004.100
==
==
==                      SUPPORT NOTE S97028
== S7000/S70000 DEFAULT NAMING CONVENTION FROM MANUFACTURING
==                           02 JUN 97
==
==
== SUMMARY:
== A naming convention for peripheral and communication devices is now
== being used during Manufacturing build of S7000/S70000 systems. The
== naming convention relates the logical names to the physical location
== of the devices. Knowledge of this naming convention could be helpful
== during initial installation and bringup of a S7000/S70000 system.
== Please note that these conventions do not apply to the TSM related
== processes (MIOE0, MIOE1, LANX, LANY, $ZTCP0, $ZTCP1, $ZTNP0, $ZTNP1,
== $ZPRP0, and $ZPRP1).
```

==
== Once a customer system has been installed, these names can be changed
== to fit the needs of the customer. Special attention should be placed
== on systems using the sockets library to interface to TCP/IP. Unless
== specifically named, the default TCP/IP process name is $ZTC0.
==
== DETAIL:
== Following are the rules and some examples:
==
== Disks: $D<cabid><slot#>
==   ex. $D3217 (Second I/O cab attached to Third cpu cab, slot 17)
==
== Tapes: $T<cabid><slot#>
==   ex. $T0150 (First cpu cab, slot 50)
==
== LAN (E4SA) Adapters: E<cabid><slot#>
==   ex. E8254 (Second I/O cab attached to Eighth cpu cab, slot 54)
==
== lif 's: L<cabid><portid>
==   ex. L51A (First I/O cab attached to Fifth cpu cab, slot 53,
==   port 2)
==
== TCP Process Names: $ZB<cabid><portid>
==   ex. $ZB51A (TCP process for above LIF)
==
== Telnet Process Names: $ZN<cabid><portid>
==   ex. $ZN51A (Telnet process for above TCP)
==
== Listner Process Names: $ZP<cabid><portid>
==   ex. $ZP51A (Listner process for above TCP)
==
== TFTP Process Names for T/3880 SWAN Concentrator: $ZF<cabid><portid>
== ex. $ZF51A (TFTP process for above TCP)
==
== WANBOOT Process Names for T/3880 SWAN Concentrator: $ZW<cabid><portid>
==   ex. $ZW51A (WANBOOT process for above TCP)
==
== SWAN Adapter Names: S<adapter#>
== ex. S19 (Nineteenth SWAN adapter)
==
==
== where:
==
== <cabid> = Cabinet ID. 01-08 for cpu cabinets; 11,21,31,41,51,61,71
==   and 81 for the first I/O cabinet attached to a cpu
==   cabinet; 12,22,32,42,52,62,72 and 82 for the second I/O
==   cabinet attached to a cpu cabinet
==
== <slot#> = Actual physical slot# in the cabinet (Disk: 1-8 & 11-18,
==   Tape: 50 & 55, LAN: 51-54)
==
== <portid>= Combination of slot# and port# mapped in the
==   following way:
==
==   Slot#   Port#   <portid>
==   51   0   0
==   51   1   1
==   51   2   2
==   51   3   3
==   52   0   4
==   52   1   5

```
==                      52      2       6
==                      52      3       7
==                      53      0       8
==                      53      1       9
==                      53      2       A
==                      53      3       B
==                      54      0       C
==                      54      1       D
==                      54      2       E
==                      54      3       F
==
== <adapter#> = incremental number from 00-99
==
================================================================================
================================================================================
== Add an E4SA ADAPTER to slot 01.1.53 as E0153 with LIFs: L018 - L01B
================================================================================
ASSUME PROCESS $ZZLAN
ADD ADAPTER E0153, TYPE E4SA, LOCATION (1,1,53), ACCESSLIST (1,0)
START ADAPTER E0153, SUB ALL
ADD LIF L018, PIF E0153.0.A
ADD LIF L019, PIF E0153.0.B
ADD LIF L01A, PIF E0153.0.A
ADD LIF L01B, PIF E0153.0.B
START LIF L018
START LIF L019
START LIF L01A
START LIF L01B
================================================================================
== Add an E4SA ADAPTER to slot 01.1.54 as E0154 with LIFs: L01C - L01F
================================================================================
ASSUME PROCESS $ZZLAN
ADD ADAPTER E0154, TYPE E4SA, LOCATION (1,1,54), ACCESSLIST (0,1)
START ADAPTER E0154, SUB ALL
ADD LIF L01C, PIF E0154.0.A
ADD LIF L01D, PIF E0154.0.B
ADD LIF L01E, PIF E0154.0.A
ADD LIF L01F, PIF E0154.0.B
START LIF L01C
START LIF L01D
START LIF L01E
START LIF L01F
================================================================================
== All the SWANs are connected to E4SA's in slots 53 and 54 in a cabinet.
================================================================================
================================================================================
==
== This section adds the SWAN concentrators. These are configured
== as ATP terminals for testing. Depending on the number of SWANs
== ordered, they are configured the following standard way:
==
== # of SWANs    Port#'s  E4SA loc.     Primary TCP Secondary TCP
== 1st 25 SWANs   8 & C   Cab 1 slot 53   $ZB018       $ZB01C
== 2nd 25 SWANs   9 & D   Cab 1 slot 53   $ZB019       $ZB01D
== 3rd 25 SWANs   A & E   Cab 1 slot 54   $ZB01A       $ZB01E
== 4th 25 SWANs   B & F   Cab 1 slot 54   $ZB01B       $ZB01F
==
== In order to successfully run SWAN's the TCP processes need to
== be correctly configured and started elsewhere on a per boot
== basis. The files STARTCOM and STARTSCF are provided on the
== $SYSTEM.ZSYSCONF subvolume as an example of one method of doing
```

```
== so.
== Sierra Naming Convention for Manufacturing
== ------------------------------------------------------------
== The new symbolic process names being proposed are as follows :
== 1. $ZZKRN.SWAN<a>-SNMP ( SNMP agent symbolic process name)
== 2. $ZTMX<a> (SNMP agent symbolic process name)
== 3. $ZZWAN.#TMX<n> ((SNMP trap multiplexor symbolic process name)
==
==
== where:
==
== <a> = A thru Z (currently A thru H will be used for the eight groups
== of swans required to configure upto 200 )
== <n> = 1 thru 9 (currently 1 thru 8 will be used for the eight groups
== of swans required to configure upto 200 )
==
==
== Each group of 25 swans are connected across two TCPIP processes. Each == of
these group requires an SNMP trap multiplexor process with a unique
== SNMP agent process.For example the first group will have swans from 1
== thru 25, the second Group will swans from 26 thru 50 and so on. Each
== group is added between two TCPIP processes, which are called TCPIP on
== PATH A and TCPIP on PATH B.
==========================================================================
==========================================================================
== Configure the ConMgr processes.
==========================================================================
ADD PROCESS $ZZWAN.#0, IOPOBJECT $SYSTEM.SYS00.CONMGR
ADD PROCESS $ZZWAN.#1, IOPOBJECT $SYSTEM.SYS00.CONMGR

==========================================================================
== Configure the TFTP server processes.
==========================================================================

ADD PROCESS $ZZWAN.#ZF018, IOPOBJECT $SYSTEM.ZTCPIP.TFTPSRV, &
TCPIP $ZB018, CPU 00, ALTCPU 01, TYPE (0,48), STARTUP "$SYSTEM.CSS00"

ADD PROCESS $ZZWAN.#ZF01C, IOPOBJECT $SYSTEM.ZTCPIP.TFTPSRV, &
TCPIP $ZB01C, CPU 00, ALTCPU 01, TYPE (0,48)

==========================================================================
== Configure the WANBoot processes.
==========================================================================

ADD PROCESS $ZZWAN.#ZW018, IOPOBJECT $SYSTEM.SYS00.WANBOOT, &
CPU 00, ALTCPU 01, TYPE (0,30), STARTUP "TCPIP $ZB018"

ADD PROCESS $ZZWAN.#ZW01C, IOPOBJECT $SYSTEM.SYS00.WANBOOT, &
CPU 00, ALTCPU 01, TYPE (0,30), STARTUP "TCPIP $ZB01C"

==========================================================================
== Add an SNMP agent for each group of swans
==========================================================================
ADD PROCESS $ZZKRN.SWANA-SNMP, &
AUTORESTART 5, &
CPU FIRST, &
DEFAULTVOL $SYSTEM.ZZSWANA, &
HIGHPIN ON, &
HOMETERM $ZHOME, &
NAME $ZTMXA, &
PRIORITY 150, &
```

```
PROGRAM $SYSTEM.SYSTEM.SNMPAGT, &
OUTFILE $ZHOME, &
STARTUPMSG "TCPIP^PROCESS^NAME $ZB018", &
STARTMODE SYSTEM
START PROCESS $ZZKRN.SWANA-SNMP
==========================================================================
== Configure a SNMP Trap multiplexor for the SWAN TCPIP's. This is
== to catch SWAN hardware event traffic that is to be routed to TSM.
==========================================================================

ADD PROCESS $ZZWAN.#ZTMX1 ,IOPOBJECT $SYSTEM.SYS00.SNMPTMUX, CPU 00, &
ALTCPU 01, TYPE (0,49), STARTUP " -a $ZTMXA -t ($ZB018,$ZB01C)"
START PROCESS $ZZWAN.#ZTMX1
==========================================================================
== Define an ATP6100 profile for the lines on SWAN S01.
==========================================================================
ADD PROFILE $ZZWAN.#ATPTERM, FILE $SYSTEM.SYS00.PATPTERM
==========================================================================

== For each of the SWAN's, you must make sure that the TRACKID is
== correctly set to the same trackid that is on the printed label
== on the rear of each of the SWAN boxes.
==========================================================================
==========================================================================
== Configure the SWAN concentrator objects for SWAN S01.
==========================================================================

ADD ADAPTER $ZZWAN.#S01,TRACKID "fill01", &
SNMPCODE $SYSTEM.CSS00.C7849P00, &
KERNELCODE $SYSTEM.CSS00.C7953P00, &
TCPIP $ZB018, HOSTIP 192.168.05.83, &
ALTTCPIP $ZB01C, ALTHOSTIP 192.168.07.83

ADD SERVER $ZZWAN.#S01.1
ADD SERVER $ZZWAN.#S01.2
ADD SERVER $ZZWAN.#S01.3

ADD PATH $ZZWAN.#S01.1.A, IPADDRESS 192.168.05.101
ADD PATH $ZZWAN.#S01.1.B, IPADDRESS 192.168.07.101
ADD PATH $ZZWAN.#S01.2.A, IPADDRESS 192.168.05.102
ADD PATH $ZZWAN.#S01.2.B, IPADDRESS 192.168.07.102
ADD PATH $ZZWAN.#S01.3.A, IPADDRESS 192.168.05.103
ADD PATH $ZZWAN.#S01.3.B, IPADDRESS 192.168.07.103

==========================================================================
== Define the WAN lines as ATP6100 lines for SWAN S01.
==========================================================================

ADD DEVICE $ZZWAN.#S1P1, PROFILE ATPTERM, &
IOPOBJECT $SYSTEM.SYS00.OATPCSS, &
CPU 00, ALTCPU 01, TYPE (53,00), RECSIZE 80, CLIP 1, LINE 0, &
ADAPTER S01, STARTDOWN

ADD DEVICE $ZZWAN.#S1S1, PROFILE ATPTERM, &
IOPOBJECT $SYSTEM.SYS00.OATPCSS, &
CPU 00, ALTCPU 01, TYPE (53,00), RECSIZE 80, CLIP 1, LINE 1, &
ADAPTER S01, STARTDOWN

ADD DEVICE $ZZWAN.#S1P2, PROFILE ATPTERM, &
IOPOBJECT $SYSTEM.SYS00.OATPCSS, &
CPU 00, ALTCPU 01, TYPE (53,00), RECSIZE 80, CLIP 2, LINE 0, &
ADAPTER S01, STARTDOWN
```

```
ADD DEVICE $ZZWAN.#S1S2, PROFILE ATPTERM, &
IOPOBJECT $SYSTEM.SYS00.OATPCSS, &
CPU 00, ALTCPU 01, TYPE (53,00), RECSIZE 80, CLIP 2, LINE 1, &
ADAPTER S01, STARTDOWN

ADD DEVICE $ZZWAN.#S1P3, PROFILE ATPTERM, &
IOPOBJECT $SYSTEM.SYS00.OATPCSS, &
CPU 00, ALTCPU 01, TYPE (53,00), RECSIZE 80, CLIP 3, LINE 0, &
ADAPTER S01, STARTDOWN

ADD DEVICE $ZZWAN.#S1S3, PROFILE ATPTERM, &
IOPOBJECT $SYSTEM.SYS00.OATPCSS, &
CPU 00, ALTCPU 01, TYPE (53,00), RECSIZE 80, CLIP 3, LINE 1, &
ADAPTER S01, STARTDOWN

==========================================================================
== At this point, if TCP/IP processes were available, start commmands
== should be issued to bring up the SWAN's. The following commented
== lines are examples. These commands are invoked in the STARTSCF
== file.
==========================================================================

== START PROCESS $ZZWAN.#0
== START PROCESS $ZZWAN.#1
== START PROCESS $ZZWAN.#ZB*
== START PROCESS $ZZWAN.#ZF*
== START PROCESS $ZZWAN.#ZW*

== START ADAPTER $ZZWAN.#*,SUB ALL
== START DEVICE $ZZWAN.#*
EXIT
```

Sources of Information for K-Series to S-Series Differences

This is part of the K-Series to S-Series Differences procedure.

For information on:	Refer to:
Learning about the major new hardware and software features introduced in each release version update (RVU)	Interactive Upgrade Guide
Determining what hardware and software is supported for NonStop S-series systems	Interactive Upgrade Guide
Finding out what hardware and software is not supported for NonStop S-series systems	For specific products, see the manual or softdoc for that product
	For all products, see the Software and Hardware Product Maintenance Lists and the Interactive Upgrade Guide
Finding information on the initial configuration of a system and subsystems	Gxx.xx Software Installation and Upgrade Guide For G06.17 and later: NonStop S-Series Planning and Configuration Guide For G06.16 and earlier: Himalaya S-Series Planning and Configuration Guide
Planning new hardware	For G06.17 and later: NonStop S-Series Planning and Configuration Guide For G06.16 and earlier: Himalaya S-Series Planning and Configuration Guide

Configuring new hardware	For G06.21 and later: NonStop S-Series Hardware Installation and FastPath Guide For G06.20: NonStop S-Series Hardware Installation Guide For G06.17, G06.18, and G06.19: NonStop S-Series Planning and Configuration Guide For G06.16 and earlier: Himalaya S-Series Planning and Configuration Guide
Determining power requirements, cooling, and other installation considerations	For G06.17 and later: NonStop S-Series Planning and Configuration Guide For G06.16 and earlier: Himalaya S-Series Planning and Configuration Guide Power Requirements for NonStop Servers also describes power and environmental requirements for NonStop Servers.
Installing NonStop S-series servers, the system console, and peripheral devices other than printers and terminals	For G06.21 and later: NonStop S-Series Hardware Installation and FastPath Guide For G06.17 through G06.20: NonStop S-Series Hardware Installation Guide For G06.16 and earlier: Himalaya S-Series Hardware Installation Guide
Migrating to specific RVUs	Interactive Upgrade Guide
Configuring hardware and software	For G06.21 and later: NonStop S-Series Hardware Installation and FastPath Guide For G06.17 through G06.20: NonStop S-Series FastPath Guide For G06.16 and earlier: Himalaya S-Series FastPath Guide
Installing and upgrading to a specific RVU	Gxx.xx Software Installation and Upgrade Guide
Determining migration considerations for optional products	Gxx.xx Software Installation and Upgrade Guide
Migrating the network connections	Interactive Upgrade Guide

Verifying the functions of the operational environment	For G06.20 and later: NonStop S-Series Hardware Installation and FastPath Guide For G06.20: NonStop S-Series Hardware Installation Guide For G06.17, G06.18, and G06.19: NonStop S-Series Planning and Configuration Guide For G06.16 and earlier: Himalaya S-Series Planning and Configuration Guide
Falling back to the previous G-series RVU	Gxx.xx Software Installation and Upgrade Guide
Falling back to a previous D-series RVU	Interactive Upgrade Guide

If you need more information, see:

- NonStop K- to S-Series Migration (http://nonstop.compaq.com/page/KtoS_Migration.html)
- Education and Training (http://education.nonstop.compaq.com/index.htm)
- Customer Support (http://nonstop.compaq.com/page/CustomerSupport)

K-Series to S-Series Migration Considerations

Map: K-Series to S-Series Differences.

Sources of Information for K-Series to S-series Differences refers you to NonStop publications that provide details on performing migration-related tasks. The following information is only a summary of some of the more critical aspects of migrating a system.

Converting Obsolete Commands

Because CMI, PUP, TAPECOM, and COUP are not used in G-series RVUs, you must update any files on your system that might contain obsolete commands:

- Function keys
- Command files
- IN files
- TACL macros

Converting Commands to SCF

From within the Subsystem Control Facility (SCF), if you type a HELP PUP or HELP COUP command, SCF displays information to help you convert the command to its SCF equivalent.

MIGSCF, a nonsupported software tool, can help convert PUP and COUP commands to SCF, when equivalent commands exist and if the procedures for configuring devices or processes are not different. For more information about installing and using MIGSCF, see the SCF Reference Manual for G-Series RVUs (for G06.16 and earlier, see SCF Reference Manual for G-Series Releases). Information about MIGSCF is also available in a file on the SUT named $SYSTEM.ZMIGSCF.MANUAL.

Comparing DSM/SCM to Install

You use DSM/SCM to install G-series RVUs of the operating sysem. If you are converting from a D-series system and are familiar with the Install product, it is important that you understand the significant differences between the phases of Install and DSM/SCM. For an overview of the installation steps involved:

- For G06.18 and later, refer to the G06.nn Software Installation and Upgrade Guide.
- For G06.17 and earlier, refer to the G-Series Highlights and Migration Planning Guide.

Creating Additional Startup Files

With the NonStop S-series server, HP ships several startup command files that automatically start devices and processes. To further automate the system startup

sequence, you can create additional startup files for the system, subsystems, system software, processes, communication lines, and applications. For examples of startup files:

- For G06.21 and later, see the NonStop S-Series Hardware Installation and FastPath Guide.
- For G06.20, see the NonStop S-Series Hardware Installation Guide.
- For G06.19 and earlier, see the NonStop S-Series Planning and Configuration Guide.

Configuring Devices and System Attributes Online

System files are handled differently in G-series systems. For the location of system configuration information and information on the contents of the $SYSTEM.SYSnn subvolume for G-series systems, see the Interactive Upgrade Guide, or for G06.19 and earlier, the G-Series Highlights and Migration Planning Guide.

For K-series servers, SYSGENR uses the CONFTEXT file to define hardware and build the processes that are specified in the configuration file into the operating system. For NonStop S-series servers, SYSGENR uses the ALLPROCESSORS section of the CONFTEXT file only and this ALLPROCESSORS section contains only a few entries.

All the other entries that would be in the ALLPROCESSORS section for K-series servers are either configured using SCF or configured automatically for S-series servers. In some cases, how an attribute is configured depends on which G-series RVUs you are running.

When you use DSM/SCM, it automatically runs SYSGENR when you change anything in the CONFTEXT file (however, CONFTEXT now contains so little information that the need for modification is rare).

SCF is the S-series equivalent for most of the configuration that is done using CONFTEXT for K-series servers. For example, many K-series communications controllers (which were configured in CONFTEXT) are replaced by the S-series SWAN concentrator or adapters. To configure the SWAN concentrator, use the SCF interface to the WAN subsystem. Then, use the SCF interface for various different subsystems depending on which communication subsystem you are configuring. For example, for an Expand-over-FOX line, use the SCF interface to the Kernel subsystem, the SCF interface to the ServerNet/FX adapter subsystem, and the SCF interface to the WAN subsystem.

SCF does not build the processes that are specified in the configuration file into the operating system. The configuration information is not located in the operating system, but instead it is stored in the configuration database files (located in the $SYSTEM.ZSYSCONF subvolume), which enables you to change the system configuration online.

Changing the System Name, System Node Number, and Time Attributes

After the first system load using the START SYSTEM action in OSM or TSM, you must use the SCF Kernel subsystem to change the initial values to values appropriate for your site for these system attributes:

System Attribute	Initial Value
SYSTEM_NAME	\NONAME
SYSTEM_NUMBER	254
DAYLIGHT_SAVINGS_TIME	NONE
TIME_ZONE_OFFSET	0:00

Caution: After you have named your system, you should avoid renaming it later because a name change affects many products. For example, changing the system name affects your SQL database. To change the system name in your SQL database, you must run the ZDSMSCM.INITENV macro. You can run the ZDSMSCM.INITENV macro **only once.** When you first obtain a new system, decide on a name **before** you do any other changes. Later on, you must use SQL commands to update your SQL databases, as described in the SQL/MP Installation and Management Guide.

For details on setting key system attributes:

- For G06.21 and later, see the NonStop S-Series Hardware Installation and FastPath Guide.
- For G06.20 and earlier, see the NonStop S-Series FastPath Guide.

Configuring IP Addresses for the Dedicated Service LAN

NonStop S-series servers are shipped with an initial set of IP addresses configured by HP for the NonStop S-series server and the system console. The IP addresses enable connections from the system console to different parts of your NonStop server. For more information, see:

- For G06.21 and later, see the NonStop S-Series Hardware Installation and FastPath Guide.
- For G06.20, see the NonStop S-Series Hardware Installation Guide.
- For G06.19 and earlier, see the NonStop S-Series Planning and Configuration Guide.

Related topics:

- Factory Default IP Addresses for NonStop S-Series Servers
- Verifying the Initial SLSA Subsystem Configuration

Recommended Software Product Revisions (SPRs)

Map: K-Series to S-Series Differences.

Note: Use our Feedback form to let us know of other considerations about SPRs that you would like for us to add to this page.

SPRs provide new features between release version updates (RVUs) or correct software problems from a previous RVU. You must apply any fallback SPRs on the RVU you might have to fall back to.

Caution: Many SPRs supported for D-series RVUs will not function properly on a G-series system. Do not apply any SPR supported for C-series or D-series RVUs without first verifying that the SPR can be installed on a G-series system.

Installing SPRs involves the following steps:

1 Before installing an SPR, determine if it can be installed on a NonStop S-series server running your current RVU.

2 Use DSM/SCM to install the SPRs.

3 If necessary, use the OSM Service Connection or the TSM Service Application to update your service processor (SP) firmware.

4 Load the system again. Some SPRs, when applied, need to have SYSGEN run. DSM/SCM does this for you.

Note: When upgrading from some RVUs, you might need to install specific SPRs on that RVU before you perform a Build/Apply for the later RVUs. See the Gxx.xx Software Installation and Upgrade Guide for detailed instructions concerning the RVUs you are working with.

Fallback Considerations

Map: K-Series to S-Series Differences.

Note: Use our Feedback form to let us know of other fallback considerations that you would like for us to add to this page.

Fallback procedures help you work with system limitations you might encounter when installing software or falling back to your previous release version update (RVU).

Source for General Instructions

For general upgrade installation instructions and fallback alerts, see the Gxx.xx Software Installation and Upgrade Guide for your current G-series RVU and the RVU you might have to fall back to. This manual also provides a list of software product revisions (SPRs) that you need to install to fall back from one G-series RVU to a previous one.

Always Save the Configuration Database

The record format of the system configuration database changes with various RVUs. To ensure system configuration integrity, always save the current system configuration database before installing software or making any changes to your system configuration.

Caution: Do not attempt a fallback without previously studying the in-depth procedures provided for the RVUs involved. Falling back involves a thorough research of all the fallback SPRs as well as the versions of your applications and configuration files.

The following summary of fallback steps provides a general overview on what is needed to fall back to a previous software RVU.

1. Use the DSM/SCM Target Interface to back out of the revision and install the fallback SPRs.
2. Start the DSM/SCM client software and monitor the status of the backout process. (Wait for backout completion before exiting DSM/SCM.)
3. Use the OSM Service Connection or the TSM Service Application to update your service processor (SP) firmware to the previous firmware version.

   (To fall back to either the G03.00 or the G05.00 RVU, use the multiple SP firmware update to update to the previous version of the SP firmware.)
4. Run ZPHIRNM from the TACL prompt and specify the SYSnn you want to fall back to.

5 Stop applications and then stop the system by halting the system processors (you do not need to power down).

6 Load the system from a saved system configuration database.

See the following for more information on starting the system using OSM:

- NonStop S-Series Operations Guide
- OSM online help
- OSM User's Guide

See the following for more information on starting the system using TSM:

- NonStop S-Series Operations Guide
- TSM online help
- TSM Online User Guide (also located in the TSM Client program group).

    - For TSM client 2000A and later: Start> Programs> Compaq TSM> TSM Documentation> TSM Online User Guide

    - For TSM client 10.0 and earlier: Start> Programs> TSM Client> TSM Documentation> TSM Online User Guide