Initial System Configuration

This chapter describes how to configure initial system parameters for the ASR 5500.

It includes the following sections:

Basic Configuration

After power is applied to the chassis and the ASR 5500 has successfully booted, the command line interface (CLI) appears on a terminal connected to the Console port of the Master MIO.

The initial configuration requires completing the following tasks via the CLI:

- Configuring a context-level security administrator and hostname.
- Configuring the Ethernet interface(s) on the MIO/UMIO.
- Configuring the system for remote CLI access via Telnet, SSH, or FTP (secured or unsecured).

Important In release 20.0 and higher Trusted StarOS builds, telnet and FTP are disabled. For additional information, see the *System Administration Guide*.

Context-level Security Administrator and Hostname

Important You must configure a context-level security administrator during the initial configuration. After completing the initial configuration process and ending the CLI session, if you have not configured a security administrator CLI access will be locked.

Step 1 At the CLI prompt, enter **config**.

```
[local]asr5500# config
[local]asr5500(config)#
```

Step 2 Enter the context configuration mode by entering **context local**.
The local context is the system's management context. Contexts allow you to logically group services or interfaces. A single context can consist of multiple services and can be bound to multiple interfaces. Enter **context local** at the CLI prompt.

```
[local]asr5500(config) context local
[local]asr5500(config-ctx)#
```

Step 3 Enter the following command to configure a context-level security administrator for the system:
administrator *name* **{ password** *password*
| encrypted password *enc_password* **} [ftp] [**
no-cli]
[timeout-absolute *absolute_time* **] [**
timeout-idle *idle_time* **]**

Keyword/Variable	Description
name	Specifies the security administrator's name as an alphanumeric string of 1 through 32 characters that is case sensitive.
password *password*	Specifies the password for the security administrator as an alphanumeric string of 1 through 63 characters that is case sensitive.
encrypted password	Specifies the encrypted password for the security administrator. This keyword is only used by the system when you save configuration scripts. The system displays the encrypted keyword in the configuration file as a flag indicating that the variable following the keyword is the encrypted version of the plain text password. Only the encrypted password is saved as part of the configuration file.
ftp	Specifies that the security administrator is allowed to access the system with the File Transfer Protocol (FTP). This option is useful to upload files (configuration or software images). **Note:** In release 20.0 and higher Trusted StarOS builds, FTP is disabled.
no-cli	Specifies that the security administrator cannot access the system's command line interface (CLI). **Note**: Use this keyword in conjunction with the **ftp** keyword to allow access to the system with FTP only.

Keyword/Variable	Description
timeout-absolute *absolute_time*	Specifies the maximum amount of time that the operator can maintain a session with the system. *absolute_time* is measured in seconds. Use any integer from 0 through 300000000. The default is 0. When the absolute timeout value is reached, the operator session is automatically terminated.
timeout-idle *idle_time*	Specifies the maximum amount of time that an operator session can remain idle before being automatically terminated. The *idle_time* is measured in seconds. Use any integer from 0 through 300000000. The default is 0.

For example:

```
[local]asr5500(config-ctx)# administrator Secure1 301delta timeout-idle 120
```

Important For additional information on configuring system administrators, refer to the *System Administration Guide*.

Step 4 Enter **exit** at the prompt to exit the context configuration mode.

```
[local]asr5500(config-ctx)# exit
[local]asr5500(config)#
```

Step 5 *Optional:* Enter **system hostname** *hostname* to configure a hostname by which the system will be recognized on the network. host_name is the name by which the system will be recognized on the network. The *hostname* can be up to 63 alphanumeric characters and is case sensitive.

Important The new *hostname* replaces the default hostname "asr5500" that appears in the CLI prompt. It also becomes the system hostname parameter for SNMP.

For example:

```
[local]asr5500(config)# system hostname node1033
[local]node1033(config)#
```

MIO/UMIO Port Numbering

The two 1 GbE ports on the MIO/UMIO cards in slots 5 and 6 can only be used as management ports. 10 GbE ports can only be used for non-local contexts (service ports). MIO/UMIO port numbers are non-contiguous.

Important For lab environments where network booting of the chassis is desirable, Ethernet 1 port on an MIO/UMIO can be used to network boot the chassis. Other MIO/UMIO ports cannot be used for network booting.

The MIO/UMIO is equipped with two daughter cards (DCs). Each DC supports ten 10 GbE ports.

Ports are specified in CLI commands by "x/yy" where x is the slot number (5 or 6) and yy the port number (1 to 29). For example, **show port info 5/20** [slot 5, port 20].

Table 1: MIO/UMIO Port Numbering 0

Port Number	Type	Connector	MIO DC	Notes
1	1000Base-T	RJ45	—	Management Port
2	1000Base-T	RJ45	—	Management Port
3	RS-232	RJ45	—	Console (serial)
4 – 9	—	—	—	Unassigned
10 – 19	10GbE	SFP+	Top	Service Port
20 – 29	10GbE	SFP+	Bottom	Service Port

The output of the **show port table** command reflects the port numbering scheme in the table above for MIO/UMIO cards equipped with two 10-port, 10 GbE daughter cards.

Configure the Ethernet Management Interface

IP Address Notation

When configuring a port interface via the CLI you must enter an IP address. The CLI always accepts an IPv4 address, and in some cases accepts an IPv6 address as an alternative.

For some configuration commands, the CLI also accepts CIDR notation.

Important Always view the online Help for the CLI command to verify acceptable forms of IP address notation.

IPv4 Dotted-Decimal Notation

An Internet Protocol Version 4 (IPv4) address consists of 32 bits divided into four octets. These four octets are written in decimal numbers, ranging from 0 to 255, and are concatenated as a character string with full stop delimiters (dots) between each number.

For example, the address of the loopback interface, usually assigned the host name localhost, is 127.0.0.1. It consists of the four binary octets 01111111, 00000000, 00000000, and 00000001, forming the full 32-bit address.

IPv4 allows 32 bits for an Internet Protocol address and can, therefore, support 2^{32} (4,294,967,296) addresses.

IPv6 Colon-Separated-Hexadecimal Notation

An Internet Protocol Version 6 (IPv6) address has two logical parts: a 64-bit network prefix and a 64-bit host address part. An IPv6 address is represented by eight groups of 16-bit hexadecimal values separated by colons (:).

A typical example of a full IPv6 address is 2001:0db8:85a3:0000:0000:8a2e:0370:7334.

The hexadecimal digits are case-insensitive.

The 128-bit IPv6 address can be abbreviated with the following rules:

- Leading zeroes within a 16-bit value may be omitted. For example, the address fe80:0000:0000:0000:0202:b3ff:fe1e:8329 may be written as fe80:0:0:0:202:b3ff:fe1e:8329
- One group of consecutive zeroes within an address may be replaced by a double colon. For example, fe80:0:0:0:202:b3ff:fe1e:8329 becomes fe80::202:b3ff:fe1e:8329.

IPv6 allows 128 bits for an Internet Protocol address and can support 2^{128} (340,282,366,920,938,000,000,000,000,000,000,000,000) internet addresses.

CIDR Notation

Classless Inter-Domain Routing (CIDR) notation is a compact specification of an Internet Protocol address and its associated routing prefix. It is used for both IPv4 and IPv6 addressing in networking architectures.

CIDR is a bitwise, prefix-based standard for the interpretation of IP addresses. It facilitates routing by allowing blocks of addresses to be grouped into single routing table entries. These groups (CIDR blocks) share an initial sequence of bits in the binary representation of their IP addresses.

CIDR notation is constructed from the IP address and the prefix size, the latter being the number of leading 1 bits of the routing prefix. The IP address is expressed according to the standards of IPv4 or IPv6. It is followed by a separator character, the slash (/) character, and the prefix size expressed as a decimal number.

The address may denote a single, distinct, interface address or the beginning address of an entire network. In the latter case the CIDR notation specifies the address block allocation of the network. The maximum size of the network is given by the number of addresses that are possible with the remaining, least-significant bits below the prefix. This is often called the host identifier.

For example:

- the address specification 192.168.100.1/24 represents the given IPv4 address and its associated routing prefix 192.168.100.0, or equivalently, its subnet mask 255.255.255.0.
- the IPv4 block 192.168.0.0/22 represents the 1024 IPv4 addresses from 192.168.0.0 to 192.168.3.255.
- the IPv6 block 2001:DB8::/48 represents the IPv6 addresses from 2001:DB8:0:0:0:0:0:0 to 2001:DB8:0:FFFF:FFFF:FFFF:FFFF:FFFF.
- ::1/128 represents the IPv6 loopback address. Its prefix size is 128, the size of the address itself, indicating that this facility consists of only this one address. An application sending a packet to this address will get the packet back after it is looped back by the IPv6 stack. The equivalent IPv4 local host address is 127.0.0.1.

The number of addresses of a subnet defined by the mask or prefix can be calculated as $2^{\text{address size - mask}}$, in which the address size for IPv4 is 32 and for IPv6 is 128. For example, in IPv4, a mask of /29 gives: $2^{32-29} = 2^3 = 8$ addresses.

Configuring the Ethernet Management Interface

The procedure below describes how to configure an Ethernet management interface on the primary MIO/UMIO in slot 5.

Important Repeat the procedure below to configure the second Ethernet management interface on the MIO/UMIO in slot 5.

Step 1 Enter **config** to enter the configuration mode.
```
[local]host_name# config
```

Step 2 Enter **context local** to enter the context configuration mode. The following prompt appears:
```
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

Step 3 Enter **interface** *interface_name* to specify a name for the interface. *interface_name* is the name of the interface. The interface name can be between 1 and 79 alphanumeric characters and is case sensitive. The following prompt appears as the system enters the Ethernet Interface Configuration mode:
```
[local]host_name(config-ctx)# interface local01
[local]host_name(config-if-eth)#
```

Step 4 Configure the IP address for the interface.

- **For IPv4**: Configure an IP address for the interface configured in the previous step by entering **ip address** *ipaddress subnetmask*. *ip_address* specifies the IP address for the interface in dotted-decimal format. *subnetmask* specifies the subnet mask for the interface in dotted-decimal or CIDR format.
- **For IPv6**: Configure an IP address for the interface configured in the previous step by entering **ipv6 address** *ipv6address subnetmask*. *ipv6address* specifies the IP address for the interface in colon-separated-hexadecimal format. *subnetmask* specifies the subnet mask for the interface in colon-separated-hexadecimal or CDIR format.

For example:
```
[local]host_name(config-if-eth)# ip address 10.0.153.100 255.255.255.248
[local]host_name(config-if-eth)# ipv6 address 2620:0:60:48::10/64
```

Step 5 Enter **exit** to exit the Ethernet interface configuration mode. The following prompt appears:
```
[local]host_name(config-ctx)#
```

Step 6 If necessary, configure a static route to point the system to a default gateway by entering the following command: **{ ip | ipv6 } route** *gw_address interface_name*. *gw_address* specifies the IP address of the default gateway in IPv4 dotted-decimal format or IPv6 colon-separated-hexadecimal format). *interface_name* specifies the name of the interface that was configured in Step 2. Refer to the *System Administration Guide* for additional information.

Step 7 Enter exit at the prompt to exit the context configuration mode. The following prompt appears:
```
[local]host_name(config-ctx)# exit
[local]host_name(config)#
```

Step 8 Enter **port ethernet** *slot/port* to enter the Ethernet Port Configuration mode. *slot* is the actual chassis slot in which the MIO/UMIO card is installed. This could be either slot number 5 or 6. *port* is the physical port on the IO/UMIO that will be used. This will be either port 1 or 2. Port 1 is the top most port.
```
[local]host_name(config)# port ethernet slot/port
[local]host_name(config-port-slot/port)#
```

Step 9 Bind the port to the interface that you created in Step 2. Binding associates the port and all of its settings to the interface. Enter the following commands:

```
[local]host_name(config-port-slot/port)# bind interface interface_name local
[local]host_name(config-port-slot/port)# no shutdown
```

interface_name is the name of the interface that you configured in Step 3.

Step 10 *Optional* – Configure the port speed by entering the following command:
medium { auto | speed { 10 | 100 | 1000 } duplex {full | half} }

Keyword/Variable	Description
auto	**NOTE: Currently MIO/UMIO ports 1 and 2 support link speeds of 1000, 100, or 10 Mbps. The ports will auto-negotiate its speed based on the fastest link partner capability.**
speed	**NOTE: Currently for MIO/UMIO ports 1 and 2 the speed setting is ignored since the port always operates in auto mode.** The possible rates are: • **10** = 10 Mbps • **100** = 100 Mbps • **1000** = 1000 Mbps
duplex	You can implement either a full or half duplex mode. **NOTE: Ethernet networking rules dictate that if a device whose interface is configured to auto-negotiate is communicating with a device that is manually configured to support full duplex mode, the first device negotiates with the manually configured speed of the second device, but only communicates in half duplex mode.**

Step 11 Enter **exit** to exit the Ethernet Interface Configuration mode.

```
[local]host_name(config-port-slot/port)# exit
[local]host_name(config)#
```

Configuring the Management Interface with a Second IP Address

If necessary, you can configure a second IP address on the same MIO management interface.

Step 1 Enter **configure** to enter the configuration mode. The following prompt appears:

```
[local]host_name# config
[local]host_name(config)#
```

Step 2 Enter **context local** to enter the context configuration mode. The following prompt appears:

```
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

Step 3 Enter **interface** *interface_name* to specify the previously named interface.

```
[local]host_name(config-ctx)# interface local01
[local]host_name(config-if-eth)#
```

Step 4 Configure the second IP address for the interface.

- **For IPv4**: Configure an IP address for the interface configured in the previous step by entering **ip address** *ipaddress subnetmask*. *ip_address* specifies the IP address for the interface in dotted-decimal format. *subnetmask* specifies the subnet mask for the interface in dotted-decimal or CIDR format.
- **For IPv6**: Configure an IP address for the interface configured in the previous step by entering **ipv6 address** *ipv6address subnetmask*. *ipv6address* specifies the IP address for the interface in colon-separated-hexadecimal format. *subnetmask* specifies the subnet mask for the interface in colon-separated-hexadecimal or CDIR format.

For example:

```
[local]host_name(config-if-eth)# ip address 10.0.153.100 255.255.255.248
[local]host_name(config-if-eth)# ipv6 address 2620:0:60:48::10/64
```

Step 5 Leave the configuration mode by entering **end**:

```
[local]host_name(config-if-eth)# end
[local]host_name#
```

Step 6 Confirm the interface ip addresses by entering **show config context local**.

Configure the System for Remote Access

When the system is configured for remote access, an administrative user may access the system from a remote location over a local area network (LAN) or wide area network (WAN) via the following communication protocols:

- Telnet
- Secure Shell (SSH
- File Transfer Protocol (FTP) (secured or unsecured)
- Trivial File Transfer Protocol (TFTP)

Important For maximum security, use SSH v2.

Important In release 20.0 and higher Trusted StarOS builds, telnet and FTP are disabled. For additional information, see the *System Administration Guide*.

Step 1 At the Exec mode CLI command prompt, enter **config** followed by **context local** to enter the Context Configuration mode.

```
[local]host_name# config
[local]host_name(config)# context local
[local]host_name(config-ctx)#
```

Step 2 Go to a previously defined interface.

```
[local]host_name(config-ctx)# interface interface_name
```

Step 3 Enter **server telnetd** to allow Telnet access.

```
[local]host_name(config-ctx)# server telnetd
```

Important For maximum system security, you should not enable telnet. In release 20.0 and higher Trusted StarOS builds, telnet is disabled.

Step 4 Enter the following command sequence to allow SSH and SFTP access:

Important **v2-rsa** is the default SSH key type.

In StarOS 19.2 and higher, the **v1-rsa** keyword has been removed from and the **v2-dsa** keyword has been concealed within the Context Configuration mode **ssh generate** CLI command. A keyword that was supported in a previous release may be concealed in subsequent releases. StarOS continues to parse concealed keywords in existing scripts and configuration files created in a previous release. But the concealed keyword no longer appears in the command syntax for use in new scripts or configuration files. Entering a question mark (?) will not display a concealed keyword as part of the Help text. A removed keyword generates an error message when parsed.

```
[local]host_name(config-ctx)# ssh generate key type v2-rsa
```

Step 5 Configure the system to support SFTP:

```
[local]host_name(config-ctx)# server sshd
[local]host_name(config-sshd)# subsystem sftp
[local]host_name(config-sshd)# exit
```

For additional information about SSH, see Configuring SSH Options, on page 10.

Step 6 Enter **server ftpd** to allow FTP access.

Important For maximum system security, you should not enable FTP. In release 20.0 and higher Trusted StarOS builds, FTP is not supported

```
[local]host_name(config-ctx)# server ftpd
```

Step 7 Enter **server tftpd** to allow TFTP access.

```
[local]host_name(config-ctx)# server tftpd
```

Step 8 Enter **exit** to exit the context configuration mode.

```
[local]host_name(config-ctx)# exit
[local]host_name(config)#
```

Step 9 Enter **end** to exit the configuration mode.

```
[local]host_name(config)# end
[local]host_name#
```

Step 10 Proceed to Save the Basic Configuration, on page 16.

Configuring SSH Options

SSHv2 RSA is the only version of SSH supported under StarOS. Keywords previously supported for SSHv1 RSA and SSHv2 DSA have been removed from or concealed within the StarOS CLI.

Important A keyword that was supported in a previous release may be concealed in subsequent releases. StarOS continues to parse concealed keywords in existing scripts and configuration files created in a previous release. But the concealed keyword no longer appears in the command syntax for use in new scripts or configuration files. Entering a question mark (?) will not display a concealed keyword as part of the Help text. Removed keywords generate an error message when parsed.

Version 1 of the SSH protocol is now obsolete due to security vulnerabilities. The **v1-rsa** keyword has been removed for the Context Configuration mode **ssh** command. Running a script or configuration that uses the SSHv1-RSA key returns an error message and generates an event log. The output of the error message is shown below:

```
CLI print failure Failure: SSH V1 contains multiple structural vulnerabilities and is no
longer considered secure. Therefore we don't support v1-rsa SSH key any longer, please
generate a new v2-rsa key to replace this old one.
```

If the system boots from a configuration that contains the **v1-rsa** key, you can expect a boot failure when logging in through SSH. The workaround is to log in via the Console port, re-generate a new ssh v2-rsa key, and configure server sshd. It will then be possible to log in via ssh.

The **v2-dsa keyword** is now concealed for the Context Configuration mode **ssh** command

The **v1-rsa** keyword has been removed from the Exec mode **show ssh key** CLI command.

Setting SSH Key Size

The Global Configuration mode **ssh key-size** CLI command configures the key size for SSH key generation for all contexts (RSA host key only).

Step 1 Enter the Global Configuration mode.

`[local]`*host_name*`#` **configure**

`[local]`*host_name*`(config)#`

Step 2 Specify the bit size for SSH keys.

`[local]`*host_name*`(config)#` **ssh key-size { 2048 | 3072 | 4096 | 5120 | 6144 | 7168 | 9216 }**

The default bit size for SSH keys is 2048 bits.

Generating SSH Keys

The **ssh generate** command generates a public/private key pair which is to be used by the SSH server. The **v1-rsa** keyword has been removed from and the **v2-dsa** keyword concealed within the **ssh generate** CLI command. The only keyword available for generating SSH keys is **v2-rsa**.

Important The generated key pair remains in use until the command is issued again.

Step 1 Enter the context configuration mode:

`[local]`*host_name*`(config)#` **context** *context_name*

```
[local]host_name(config-ctx)#
```

Step 2 Generate an SSH key pair.

`[local]`*host_name*`(config-ctx)#` **ssh generate key type v2-rsa**

`[local]`*host_name*`(config-ctx)#`

Setting SSH Key Pair

The **ssh key** command sets the public/private key pair to be used by the system. The **v2-dsa** keyword is concealed in the **ssh key** command.

Specify the SSH key pair parameters.

`[local]host_name(config-ctx)#` **ssh key** *data* **length** *octets* **type v2-rsa**

Notes:

- *data* is the encrypted key expressed as an alphanumeric string of 1 through 1023 characters
- **length** *octets* is the length of the encrypted key in octets expressed as an integer from 0 through 65535
- **type** specifies the key type; **v2-rsa** is the only supported type.

Important *For releases prior to 20.0*, StarOS supports a maximum of 64 configurable authorized SSH keys. *For release 20.0 and higher*, StarOS supports a maximum of 200 configurable authorized SSH keys.

Specifying SSH Encryption Ciphers

The SSH Configuration mode **ciphers** CLI command configures the cipher priority list in sshd for SSH symmetric encryption. It changes the cipher options for that context.

Step 1 Enter the SSH Configuration mode.
`[local]`*host_name*`(config-ctx)#` **server sshd**

Step 2 Specify the desired encryption algorithms.
`[local]`*host_name*`(config-sshd)#` **ciphers** *algorithm*

Notes:

- *algorithm* is a string of 1 through 511 alphanumeric characters that specifies the algorithm(s) to be used as a single string of comma-separated variables (no spaces) in priority order from those shown below:
  - **blowfish-cbc –** symmetric-key block cipher, Cipher Block Chaining, (CBC)
  - **3des-cbc –** Triple Data Encryption Standard, CBC
  - **aes128-cbc –** Advanced Encryption Standard (AES), 128-bit key size, CBC
  - **aes128-ctr –** AES, 128-bit key size, Counter-mode encryption (CTR)
  - **aes192-ctr –** AES, 192-bit key size, CTR
  - **aes256-ctr –** AES, 256-bit key size, CTR
  - **aes128-gcm@openssh.com –** AES, 128-bit key size, Galois Counter Mode [GCM], OpenSSH
  - **aes256-gcm@openssh.com –** AES, 256-bit key size, GCM, OpenSSH
  - **chacha20-poly1305@openssh.com –** ChaCha20 symmetric cipher, Poly1305 cryptographic Message Authentication Code [MAC], OpenSSH

The default string for algorithm is:

```
blowfish-cbc,3des-cbc,aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,
aes256-gcm@openssh.com,chacha20-poly1305@openssh.com
```

Step 3 Exit the SSH Configuration mode.
`[local]`*host_name*`(config-sshd)#` **end**
`[local]`*host_name*`#`

Set System Timing

Setting the System Clock and Time Zone

Use the following command sequence to configure the system clock and time zone:

```
[local]host_name# clock set YYYY:MM:DD:HH:mm or YYYY:MM:DD:HH:mm:ss
[local]host_name# config
[local]host_name(config)# clock timezone timezone [ local ]
[local]host_name(config)# end
[local]host_name#
```

Important See the online help for the **clock timezone** command for a complete list of supported time zones. The optional **local** keyword indicates that the time zone specified is the local timezone.

Important Daylight Savings Time is automatically adjusted for time zones supporting it.

Save your configuration as described in Save the Basic Configuration, on page 16.

Enter **show clock** to verify that you configured the time and time zone correctly:

```
[local]host_name# show clock
Wednesday October 10 13:08:27 us-eastern 2012
```

Configuring Network Time Protocol Support

This section describes how to enable the use of the Network Time Protocol (NTP) on the ASR 5500 chassis.

Overview of NTP Support

Many of the services offered by the ASR 5500 platform require accurate timekeeping derived through NTP. If the time reference(s) used by StarOS are not accurate, the services may be unreliable. For this reason it should be assumed that normal system operation requires that NTP be configured.

The system uses NTP to synchronize internal clocks on the chassis to external time sources (typically GPS NTP sources, or other Stratum 2 or 3 servers, switches or routers).

By default, NTP is not enabled externally and should be configured when the system is initially installed. When enabled, the active MIO/UMIO will synchronize with external sources. If not enabled, the active MIO/UMIO will use its local clock as a time source. In the event of an NTP server or network outage, an already running MIO/UMIO will continue to use NTP to maintain time accuracy, but in a holdover mode.

All cards with CPUs synchronize to the active MIO/UMIO internally. This occurs even if an external NTP server is not configured. In the event of a MIO/UMIO switchover, all other cards will start synchronizing with the newly active MIO/UMIO automatically.

The system should have:

- NTP enabled.

- NTP configured for use in the local context only. Use of other contexts (which can be specified in the enable configurable) will cause issues.
- NTP configured for three external NTP servers. With three or more servers, outlyers and broken or misconfigured servers can be detected and excluded. Generally, the more servers the better (within reason).

Important Do not configure any external NTP servers using the **prefer** keyword. The NTP clock selection algorithms already have the built-in ability to pick the best server. Use of **prefer** usually results in a poorer choice than NTP can determine for itself.

Important Do not change the **maxpoll**, **minpoll**, or **version** keyword settings unless instructed to do so by Cisco TAC.

Basic NTP Configuration

Important Configure the system clock and time zone prior to implementing NTP support. This greatly simplifies the time zone shift that must be corrected by the NTP server. See Setting the System Clock and Time Zone, on page 13.

Use the following example to configure the necessary NTP association parameters:

```
[local]host_name# config
[local]host_name(config)# ntp
[local]host_name(config-ntp)# enable
[local]host_name(config-ntp)# server ip_address1
[local]host_name(config-ntp)# server ip_address2
[local]host_name(config-ntp)# server ip_address3
[local]host_name(config-ntp)# end
[local]host_name#
```

By default *context_name* is set to *local*. This is the recommended configuration.

A number of options exist for the **ntp server** command. Refer to the *NTP Configuration Mode Commands* chapter in the *Command Line Interface Reference* for more information.

Important Configure the system with at least three (preferably four) NTP servers.

Save the configuration as described in Save the Basic Configuration, on page 16.

Configuring NTP Servers with Local Sources

NTP can use network peers, local external clocks (such as GPS devices), or a local clock with no external source.

A local clock with no external source is usually a last-resort clock when no better clock is available. It is typically configured on a site's intermediate NTP server so that when a WAN network outage occurs, hosts within the site can continue to synchronize amongst themselves.

You can configure this in ntpd or on many commercially available NTP devices. This local clock should always have a high stratum number (8+) so that under normal conditions (when real sources are available) this local clock will not be used.

Using a Load Balancer

The NTP daemon and protocol assume that each configured server is running NTP. If an NTP client is configured to synchronize to a load balancer that relays and distributes packets to a set of real NTP servers, the load balancer may distribute those packets dynamically and confuse the NTP client. NTP packets are sensitive to latency and jitter. Relaying them through a load balancer can confuse the NTP client and is not a supported practice.

Verifying the NTP Configuration

To verify the NTP Configuration, enter the **show ntp associations** command at the Exec mode. The output displays information about all NTP servers.

The table below lists and briefly describes the parameters that appear in the output of the **show ntp associations** command.

Table 2: Output Parameters for show ntp associations

Column Title	Description
remote	Lists the current NTP servers. One of these characters precedes each IP address to show the server's current condition: • () Rejected/No response • X False tick • (.) Excess • - Outlyer • + Candidate • # Selected • * System peer • (o) PPS peer
refid	Last reported NTP reference to which the server is synchronizing.
st	NTP server stratum level.
t	Communication type: broadcast, multicast, etc.
when	Number of seconds since the last contact.
poll	Polling interval between the system and the NTP server.

Column Title	Description
reach	Octal value of the reachability shift register indicating which responses were received for the previous eight polls to this NTP server.
delay	Round-trip delay (in milliseconds) for messages exchanged between the system and the NTP server.
offset	Number of milliseconds by which the system clock must be adjusted to synchronize it with the NTP server.
jitter	Jitter in milliseconds between the system and the NTP server.

Enable CLI Timestamping

To display a timestamp (date and time) for every command that is executed on the CLI, enter the **timestamps** command at the root prompt for the Exec mode:

```
[local]host_name# timestamps
```

Immediately after you execute the command, the date and time appear.

Save the configuration as described in Save the Basic Configuration, on page 16.

Save the Basic Configuration

Save this basic system configuration information to a file locally. The following procedure saves the configuration file to flash memory in the MIO/UMIO.

Step 1 You must be at the root prompt for the Exec mode to save the configuration file.

```
[local]host_name#
```

Step 2 To save your current configuration, enter the following command:

```
[local]host_name# save configuration /flash/system.cfg
```

This completes the basic configuration process.

Additional Configuration Tasks

Establishing the basic configuration allows an operator to access the ASR 5500 for management purposes. Additional configuration settings are required for full operational deployment within a provider network. To complete these tasks, refer to the following documents:

- *System Administration Guide*
- *Command Line Interface Reference*

- *Administration Guide* specific to the type of product being deployed.
- StarOS *Release Notes*