ERICSSON

# Configuring Contexts and Interfaces

## SYSTEM ADMINISTRATOR GUIDE

83/1543-CRA 119 1170/1-V1 Uen H

# Contents

# 1 Configuring Contexts

This document provides overview of contexts, describes the tasks used to configure basic features for contexts and interfaces, and provides configuration examples.

For protocol- or feature-specific commands that appear in context configuration mode, see the *Command List*.

This document applies to both the Ericsson SmartEdge® and SM family routers. However, the software that applies to the SM family of systems is a subset of the SmartEdge OS; some of the functionality described in this document may not apply to SM family routers.

For information specific to the SM family chassis, including line cards, refer to the SM family chassis documentation.

For specific information about the differences between the SmartEdge and SM family routers, refer to the Technical Product Description *SM Family of Systems* (part number 5/221 02-CRA 119 1170/1) in the **`Product Overview`** folder of this Customer Product Information library.

## 1.1 About Contexts

One of the most advanced features of the SmartEdge router is the ability to support both a local context and multiple other contexts. A context is an instance of a virtual router, complete with its own management domain, authentication, authorization, and accounting (AAA) name space, IP address space, and routing protocols. The SmartEdge router can support over a thousand contexts. While these contexts share common resources, such as memory and processor cycles, each context is completely independent of all other contexts configured on the SmartEdge router. Contexts are conceptually similar to virtual routing and forwarding (VRF) instances, but are more powerful, and offer advanced capabilities not available in existing VRF implementations.

A context is not a dedicated, hard-wired set of physical ports, slots, CPUs, and memory. It is a logical construct that is created or deleted through configuration commands. The administrator has complete flexibility to determine which ports and circuits are associated with each context.

A physical circuit, on the other hand, refers to the physical communications channels through which packets are sent to or received by the SmartEdge router. A port, channel, or circuit is not considered part of any context. Examples of circuits, in the broadest sense of the term, include Ethernet, Packet over SONET/SDH (POS), and Layer 2 circuit endpoints, such as Asynchronous Transfer Mode (ATM), Frame Relay, and 802.1Q permanent virtual circuits (PVCs).

However, no traffic can flow over a circuit until it is associated with an interface through a configuration step called “binding”. The binding ties a particular circuit to a particular interface, and the circuit is said to be bound to that interface. The binding is simply a configuration statement provided as part of the circuit definition.

### 1.1.1 Local Context

A SmartEdge router with a single configured context is similar to traditional networking products. This is referred to as a “single-context configuration.” Every configuration includes the special context “local” that cannot be deleted. In single-context configurations, the local context is the only context.

### 1.1.2 Multiple Contexts

A SmartEdge router configured to support several contexts simultaneously is said to support multiple contexts. The software base is designed to support multiple contexts. All operating system features, such as the command-line interface (CLI), management features, such as the Simple Network Management Protocol (SNMP); troubleshooting features, such as ping, traceroute, debug, and system logging, IP addresses, interfaces, access control lists (ACLs) and routing protocol instances, are implemented on a per-context basis. When a new feature is added, it inherits the multicontext infrastructure, allowing the new functions to be used in a multicontext application.

Every context has its own complete implementation of IP routing protocols, including the Border Gateway Protocol (BGP), Open Shortest Path First (OSPF) protocol, Intermediate System-to-Intermediate System (IS-IS) protocol, and the complete IP multicast routing protocol suite. In particular, each BGP instance has its own autonomous system number (ASN), policies, and import and export properties, and each context can contain any mix of Interior Gateway Protocol (IGP) routing protocols. All routing protocols are implemented as multithreaded processes with multiinstance capability, which in combination with an intelligent scheduler, provides an efficient multicontext routing protocol implementation.

Each context has its own IP address space, which can overlap with the address space of other contexts. Every physical I/O channel—for ports, channels, subchannels, and ATM, Frame Relay, and 802.1Q PVCs—can be associated with a context through configuration commands and the binding process.

A context can have its own unique set of CLI administrators, each with their own (possibly overlapping) administrator names and passwords, and each authenticated through their own set of AAA databases. Each context can have its own SNMP community strings. This support allows Virtual Private Network (VPN) customers visibility into their own routing context for debugging and troubleshooting purposes.

### 1.1.3 Applications for Multiple Contexts

A simple yet powerful application for multiple contexts is olympic services, wherein a provider offers platinum, gold, and silver service classes to its customers, as a function of oversubscription (statistical gain) that is engineered at the access point. This setup takes advantage of the closed administrator group aspect of contexts, and less so of the ability of contexts to support multiple, overlapping address spaces.

Many service providers have different service offerings. For reasons ranging from mergers and acquisitions to organizational structure, these services often operate within their own, respective, autonomous systems. With conventional routers, an independent, physical router must be used for each autonomous system (AS), because conventional routers allow only a single routing instance in an AS.

However, each context in the SmartEdge router can have its own routing instance, for example BGP, and each BGP instance can optionally be a member of its own AS, with its own set of policies. The multiple context capability of the SmartEdge router allows a single router to replace multiple conventional routers in such an application. Each context appears as a virtual router, and thus the SmartEdge router can perform the functions of multiple routers simultaneously.

### 1.1.4 Multiple VPN Contexts

Provider Edge (PE) routers maintain a separate VPN context for each VPN connection. Each customer connection, such as an ATM, Frame Relay, or 802.1Q PVC, is mapped to a specific VPN context. Multiple ports on a PE router can be associated with a single VPN context; however, it is the ability of PE routers to maintain multiple VPN contexts that supports the per-VPN segregation of routing information.

### 1.1.5 Intercontext Interfaces

An intercontext interface allows routing protocols to exchange routing information between two or more contexts within the same physical SmartEdge router; this capability is similar to the exchange of routing information between two physical routers. An intercontext interface can be either a point-to-point intercontext interface or a point-to-multipoint (referred to as a LAN) intercontext interface. The point-to-point type links two intercontext interfaces of two different contexts; for this type of intercontext interface, there can be only two intercontext interfaces with the same ID on the SmartEdge router. The LAN type links multiple interfaces in multiple contexts. For LAN intercontext interfaces, the *id* argument specifies the group identifier for all the intercontext interfaces with the same ID that are linked together.

Use an intercontext interface only for:

- Intermediate System-to-Intermediate System (IS-IS) routing

- Intercontext static routes
- Interfacing to the default multicast domain tree (MDT) group in multicast virtual private networks (VPNs).

If you provide an IP address to an intercontext interface, the netmask `255.255.255.255` is not allowed.

### 1.1.6 Administrator Authentication to Local and Non-Local Contexts

Each context is configured with an AAA search list for authenticating administrators. The AAA search list determines the order in which administrators of a particular context are authenticated. At the logon prompt, the administrator provides a structured administrator name of the form ***admin-name@ctx-name***. The ***ctx-name*** portion of the administrator name string selects the context; the AAA search order for that context is used to authenticate the administrator.

**Note:** The separator character between the ***admin-name*** and the ***ctx-name*** arguments is configurable and can be any of %, -, @, _, \\, #, and /. For information about configuring the separator character, see the *Command List*. The default value is @, which is used throughout this document.

If Secure Shell (SSH) or Secure File Transfer Protocol (SFTP) access is requested without the context name being included in the user name, a user name belonging to the context where the accessed interface is defined is used.

If Telnet access is requested without the context name being included in the user name, a user name belonging to the local context is used.

The context of the data path through which an administrator's Telnet or SSH packets arrive and leave the SmartEdge router is not dependent on the context to which the administrator authenticates. For example, it is valid for an administrator whose workstation is connected to an Ethernet segment bound to the **corpA** context to log on to the SmartEdge router as **root@local**, thereby becoming a local administrator, even though the path through which Telnet or SSH packets arrive is through a port on the SmartEdge router that is bound to the **corpA** context.

### 1.1.7 Administrator Privileges for Local and Non-Local Contexts

With regard to the operating system concept of multiple contexts, there are two types of administrators:

- Local—An administrator authenticated to the local context. The local administrator has a structured administrator name of the form ***admin-name*** **`@ local`**.

- Non-local—An administrator authenticated to any context other than the local context. An example of a non-local administrator has an administrator name of the form ***admin-name@ctx-name*** is **joe@vpn1**, where **vpn1** is the name of the context.

  An administrator authenticated to the local context, given appropriate administrator privileges, can configure all functions on the SmartEdge router, including functions for each context, and global entities, such as ports, port profiles, SNMP, and so on.

  Non-local administrators have no configuration mode privileges and have restricted exec mode privileges. An exec command is accessible to a non-local administrator if its purpose is to provide information about, or to generate limited troubleshooting for, the context to which the administrator is authenticated. For example, when an administrator authenticated as **fred@corpA** runs the **`show ip route`** command (in global configuration mode), the output displays only the IP routing table for the context **corpA** and not for any other context.

**Note:** In addition to context authentication, the SmartEdge router software supports privilege levels that affect an administrator's access to the operating system CLI. Both administrators and commands have default privilege levels that you can modify. For details, see the **`privilege max`** and **`privilege`** commands in *Command List*.

### 1.1.8 Subscriber Domains and Domain Aliases

A subscriber domain is the name of the context in which the subscriber is configured or a domain alias for that context (as defined by the **`domain`** command). Use subscriber domains as one way to control which subscribers can connect to each context. If enabled by the **`service domain-wildcard`** command, the subscriber domain alias can be specified using the asterisk (*) wildcard character.

## 1.2 Configuring Contexts

**Note:** In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the *Command List*.

To configure the basic features for a context and accounts for the administrators who manage them, perform the tasks described in the following sections:

For more information about configuring administrator accounts, including how to configure authentication, session limits, and command authorization, see the *Command List*.

### 1.2.1 Enable Multiple-Context Service

To configure any context other than the local context, you must enable multiple-context service; perform the task described in Table 1.

*Table 1 Enable Multiple-Context Service*

| Task | Root Command | Notes |
|---|---|---|
| Enable multiple-context service. | *service multiple-contexts* | Enter this command in global configuration mode. |

### 1.2.2 Configure a Context

To configure a context, perform the tasks described in Table 2

*Table 2 Configure a Context*

| # | Task | Root Command | Notes |
|---|---|---|---|
| 1. | Create or modify a context and access context configuration mode with one of the following tasks: | | |
| | Create or modify a standard context and access context configuration mode. | *context* | Enter this command in global configuration mode. |
| | Create or modify a VPN context and access context configuration mode. | *context vpn-rd* | Enter this command in global configuration mode. |
| 2. | Specify a privilege level password in the local database for the **`enable`** command with one of the following tasks: | | |
| | Configure a password that the system will encrypt. | *enable password* | Enter this command in context configuration mode. |
| | Configure a password in encrypted form. | *lsp* | Enter this command in context configuration mode. |
| 3. | Specify how the system performs privilege level authentication. | *enable authentication* | Enter this command in context configuration mode. |
| 4. | Specify general attributes for the context (all attributes are optional): | | |
| | Specify falling-threshold parameters for IP pools in the context. | *ip pool (context configuration)* | Enter this command in context configuration mode. |
| | Create one or more unique domain aliases for a context. | *domain (context)* | Enter this command in context configuration mode. |

*Table 2 Configure a Context*

| # | Task | Root Command | Notes |
|---|---|---|---|
| | Enable the use of the asterisk (*) wildcard character in subscriber domain aliases. | *service domain-wildcard* | Enter this command in global configuration mode |
| | Apply an existing bulkstats schema profile to the context. | *bulkstats schema* | Enter this command in context configuration mode. |

### 1.2.3 Configure an Administrator Account in a Context

To configure an administrator account in a context, perform the tasks described in Table 3.

*Table 3 Configure an Administrator Account in a Context*

| # | Task | Root Command | Notes |
|---|---|---|---|
| 1. | Create an administrator logon account and access administrator configuration mode. | *administrator* | Enter this command in context configuration mode. |
| 2. | Specify general attributes for the account, enter these commands in administrator configuration mode (all attributes are optional). | | |
| | Assign a full name or textual description for the administrator. | *full-name* | |
| | Specify the initial privilege level for exec sessions initiated by the administrator. | *privilege start* | |
| | Specify the maximum privilege level for the administrator. | *privilege max* | |
| | Specify public key authentication for the administrator who is accessing the SmartEdge router CLI through SSH. | *public-key* | |

### 1.2.4 Example: Administrator Privileges

The following example displays the creation of an administrator account with the administrator name **super** and the password **icandoanything**. When the administrator logs on to the system, the initial privilege level is **10**. The administrator can modify the privilege level up to the maximum of **15:**

```
[local]Redback#configure
```

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#administrator super password
icandoanything

[local]Redback(config-administrator)#full-
name "Fred P. Lynch x.1234"

[local]Redback(config-administrator)#privilege start 10
[local]Redback(config-administrator)#privilege max 15
[local]Redback(config-administrator)#enable password
pwd_for_priv_level_15
[local]Redback(config-ctx)#
```

Because this account is created in the **local** context, this administrator is able to view and modify the entire system configuration and view all running information on the system.

The next time the administrator **super** logs on to the system with the **icandoanything** password, the administrator is at privilege level 10. To enter privilege level 15, the administrator needs to issue the following commands with the password chosen to enter privilege 15 (in this example, the chosen administrator password is **pwd_for_priv_level_15**). This password will not be displayed at the CLI:

```
[local]Redback>enable
Password <enter the password, pwd_for_priv_level_15>
[local]Redback#
```

### 1.2.5 Example: Public Keys

The following example configures a public RSA key for the administrator, **jewel:**

```
[local]Redback(config-administrator)#public-key RSA

 Enter public key for the user

$0531362763821938699612467 61 admin@local
% adding public key 1024 35 13877892548755011249626406025749447395347 7
80214577723471190493135601780425356384229093001105445048536324328024 64
00199717731319844418831089264593496852809170833789839891527385879500 64
52667325324989385497793626010262714937340759030252164573952312178584 14
47489051486168865249795082968405313627638219386996124676 1 to user jewel
```

## 1.3 Performing Context Operations

Context operations tasks are listed in Table 4. Enter **show** commands in any mode; enter all other commands in exec mode.

*Table 4 Context Operations Tasks*

| Task | Root Command |
|---|---|
| Terminate one or all of an administrator's remote (Telnet or Secure Shell [SSH]) terminal sessions. | *clear administrator* |

*Table 4 Context Operations Tasks*

| Task | Root Command |
|---|---|
| When entered from exec mode, this command displays context-specific information without entering context configuration mode. When entered from global configuration mode, this command creates a new context or specifies an existing context, and enters context configuration mode. | *context* |
| Enable the generation of general debug messages for the current context. | *debug context* |
| Display all administrator sessions on a system. | *show administrators* |
| Display configuration information for a specified context. | *show configuration context* |
| Display a list of configured context names. | *show context* |
| Display the status of the IP addresses in the specified IP pool, in all IP pools in the specified interface, or in all IP pools in the current context or range. | *show ip pool* |
| Display an administrator's public keys. | *show public- key* |

# 2 Interfaces

This document provides an overview of interfaces, describes the tasks used to configure basic features for interfaces, and provides configuration examples and detailed descriptions of the commands used to configure these features through the operating system.

Protocol-specific information and commands appear in the document for each specific protocol.

**Note:** In the following descriptions, the term controller card applies to the Cross-Connect Route Processor (XCRP4) Controller card, including the controller carrier card unless otherwise noted.

The term controller carrier card refers to the controller functions on the carrier card within the SmartEdge 100 chassis. The term I/O carrier card refers to the traffic card functions on the carrier card; these functions are compatible with the similar functions that are implemented on the traffic card that are supported on all other SmartEdge routers.

## 2.1 About Interfaces

Within the SmartEdge router, an interface is a logical entity that provides higher-layer protocol and service information, such as Layer 3 addressing. Interfaces are configured as part of a context and are independent of physical ports and circuits. The separation of the interface from the physical layer allows for many of the advanced features offered by the SmartEdge router. For higher-layer protocols to become active, you must bind a physical port or circuit to an interface.

With Dynamic Host Configuration Protocol (DHCP) relay enabled on an interface, the SmartEdge router can examine all responses from a DHCP relay server and note the bindings among the assigned IP address, the requesting Ethernet medium access control (MAC) address, and the circuit from which the request was received.

The result is a behavior similar to that of secured Address Resolution Protocol (ARP). Because an entry is automatically placed in the SmartEdge router host table for this binding, the need to use secured ARP for the binding is eliminated. This ensures that the address cannot be spoofed and that traffic cannot be redirected.

The SmartEdge router supports the following types of interfaces:

- Bridged interface—Allows circuits, such as Ethernet ports or Asynchronous Transfer Mode (ATM) permanent virtual circuits (PVCs) with RFC 1483 bridged encapsulation to be bridged. A bridged interface is associated with a bridge in this context by using the **`bridge`** command (in interface

configuration mode). For more information on the **bridge** command (in interface configuration mode), see *Configuring Bridging*.

- Intercontext interface—Allows the Intermediate System-to-Intermediate System (IS-IS) routing protocol to exchange routing information between two or more contexts within the same physical SmartEdge router; this capability is similar to the exchange of routing information between two physical routers. An intercontext interface can be either a point-to-point intercontext interface or a point-to-multipoint (referred to as a LAN) intercontext interface:
  - The point-to-point type links two intercontext interfaces of two different contexts; for this type of intercontext interface, there can be only two intercontext interfaces with the same ID on the SmartEdge router.
  - The LAN type links multiple interfaces in multiple contexts. For LAN intercontext interfaces, the ***id*** argument specifies the group identifier for all the intercontext interfaces with the same ID that are linked together.
- Loopback interface—Has no explicit association with any circuit in the system. This feature is useful in applications that require an IP address in a particular context, but not necessarily a physical connection, because a loopback interface is always up. For example, loopback interfaces can be useful for routing protocols, because the interface is not associated with a physical port that can go down. You cannot configure secondary IP addresses for a loopback interface.
- Multibind interface—Allows multiple circuits to be bound to the interface. This feature is useful when the interface is used for subscriber circuits. You can also specify that a multibind interface act as a last-resort interface.
- Last-resort interface, which is a type of multibind interface—Acts as a fallback for any incoming subscriber circuit for which the subscriber record does not include an IP address that is assigned to any other interface. If a subscriber session is established, and there is no valid interface to which it can bind, the session binds to the last-resort interface.

Each interface must have an IP address you can explicitly specify, using the **ip address** command (in interface configuration mode), or implicitly, using the **ip unnumbered** command (in interface configuration mode). When specified implicitly, the interface borrows the IP address from the interface specified by the command. The IP address is used as the source address for routing updates and packets, thus conserving network and address space. Last-resort interfaces must always be configured using the **ip unnumbered** command.

**Note:** When IP Version 6 (IPv6) addresses are not referenced or explicitly specified, the term IP address can refer generally to IP Version 4 (IPv4) addresses, IPv6 addresses, or IP addressing. In instances where IPv6 addresses are referenced or explicitly specified, the term IP address refers only to IPv4 addresses. For a description of IPv6 addressing and the types of IPv6 addresses, see RFC 3513, `Internet Protocol Version 6 (IPv6) Addressing Architecture`.

IPv6 is a new version of the Internet Protocol, designed as the successor toIPv4. IPv6 is fully described in RFC 2460, `Internet Protocol, Version 6 (IPv6) Specification`. The changes from IPv4 to IPv6 include:

- Increase in address size from 32 bits to 128 bits
- Simplified header
- Extensible header with optional extension headers
- Designed to co-exist with IPv4
- Uses multicast addresses instead of broadcast addresses

**Note:** When adding interfaces to a context follow these limitations:

- Do not configure more than one interface IP address in a context on the same subnet.
- The host portion of an interface IP address cannot be 0 or the subnet for a broadcast IP address.

## 2.2 Configuring Interfaces

### 2.2.1 Configuration Guidelines

Consider the following guidelines for interfaces, IP addresses, and IP pools:

- A standard (one that is not a last-resort interface) multibind interface must have an IP address assigned explicitly, using the **`ip address`** command (in interface configuration mode).
- A last-resort multibind interface must be configured as unnumbered, using the **`ip unnumbered`** command (in interface configuration mode).
- The interface from which the IP address is borrowed for an unnumbered interface must be in the same context as the unnumbered interface.
- An IP address can be of any class: A, B, or C.
- You can't configure more than one interface IP address in a context on the same subnet.
- The host portion of an interface IP address cannot be 0 or the broadcast address for a subnet.
- Only standard and last-resort multibind interfaces support IP pools.
- IP pools can be named or unnamed.
- Last-resort interfaces support up to 2,048 IP pools, which can be named or unnamed; standard multibind interfaces can be configured to have only a single IP pool, which can be either named or unnamed.

- An IP pool can have the same name as an interface within a context, but the name must be unique among IP pools within that context.

- The IP addresses in a named IP pool are reserved; they can be assigned only to subscribers that have been configured to use this specific IP pool. The assignment can be made either by the **`ip address`** command (in subscriber configuration mode) or by the vendor-specific attribute (VSA) 36 provided by Ericsson AB, IP-Address-Pool-Name.

- For a standard multibind interface, the specified IP address for a pool must be within the subnet specified by the primary IP address for the interface, and the prefix length for the pool must be either the same length or larger than that specified for the interface. Standard network subnetting rules apply for creating the range of IP addresses for the pool.

- For a last-resort multibind interface, the specified IP address and subnet range for any pool cannot overlap the subnet range assigned to any other interface with the exception of loopback interfaces. IP addresses that are assigned to loopback interfaces and that overlap the subnet range for an IP pool in a last-resort multibind interface are marked as reserved in the IP pool.

- Depending on the value of the ***`netmask`*** or ***`prefix-length`*** argument for the IP address assigned to the interface and the range of IP addresses assigned to a pool in that interface, the IP address assigned to the interface and its network (.0) and broadcast (.255) IP addresses need not overlap the IP addresses assigned to the pool. If they do overlap the range of IP addresses assigned to the pool, they are excluded from the pool.

- The maximum number of IP addresses in a pool is 65,536 addresses; therefore, the minimum values for the ***`netmask`*** and ***`prefix-length`*** arguments is 255.255.0.0 and 16, respectively.

- For pools with Class A or Class B addresses:

  All IP addresses in the assigned range are included in the pool except the interface, network (.0), and broadcast (.255) IP addresses assigned to the interface when they overlap with the pool IP addresses.

- For pools with Class C addresses:

  By default, all network (.0) and broadcast (.255) IP addresses are excluded from the pool, even if the pool is supernetted; to include any intervening network and broadcast IP addresses in any IP pool configured with Class C addresses in the context, you must use the **`ip pool`** command (in context configuration mode) with the **`options use-class-c-bcast-addrs`** construct.

### 2.2.2 Configure Basic Features for an Interface

To configure the basic features for an interface, perform the tasks described in Table 5; enter all commands in interface configuration mode, unless otherwise specified.

*Table 5 Configure Basic Features for an Interface*

| # | Task | Root Command | Notes |
|---|---|---|---|
| 1. | Create a new interface, or modify an existing one, and access interface configuration mode. | *interface (context)* | Enter this command in context configuration mode. |
| 2. | Associate a text description with the interface. | *description (interface)* | |
| 3. | Specify that the Don't Fragment (DF) flag in received packets be ignored. | *ip clear-df* | |
| 4. | Specify that the Internet Control Message Protocol (ICMP) Destination Unreachable packet-too-big message be suppressed. | *ip icmp* | |
| 5. | If the interface is not bridged, configure IP addresses for the interface with one of the following tasks: | | |
| | Assign a primary or secondary IP address. | *ip address (interface)* | This command is not used for last-resort interfaces. |
| | Assign a primary or secondary IPv6 address. | *ipv6 address* | |
| | Create a pool of IP addresses for the interface. | *join-group* | |
| | Select a fixed IP address as the source address for one or more protocols. | *ip source-address* | Use this command only with loopback interfaces. |
| | Enable IP processing on an interface without assigning it an explicit IP address. | *ip unnumbered* | This command is required for last-resort interfaces. |
| | Enable IPv6 processing on an interface without assigning it an explicit IPv6 address. | *ipv6 unnumbered* | |
| 6. | Set the maximum transmission unit (MTU) for traffic sent on the circuit to which the interface is bound. | *ip mtu* | |

*Table 5 Configure Basic Features for an Interface*

| # | Task | Root Command | Notes |
|---|---|---|---|
| | Set the MTU for traffic sent on the circuit to which the interface is bound. | *ipv6 mtu* | |
| 7. | Set the maximum segment size (MSS) for TCP sessions. | *ip tcp mss* | |
| 8. | If the interface is bridged, bind it to an existing bridge group. | *bridge* | For a description of this command, see *Configuring Bridging*. |

### 2.2.3 Examples: Configuring Interfaces

The following example creates the **enet71** interface, assigns it an IP address, and binds it to an Ethernet port:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface enet71
[local]Redback(config-if)#ip address 10.1.2.1
255.255.255.0
[local]Redback(config-if)#exit
[local]Redback(config)#port ethernet 7/1
[local]Redback(config-port)#bind interface enet71 local
```

The following example creates a loopback interface (**loop-lo2**) and an unnumbered interface (**unnum2**). The unnumbered interface borrows its IP address from the loopback interface. Do not bind a circuit to the loopback interface:

```
[local]Redback(config-ctx)#interface loop-lo2 loopback
[local]Redback(config-if)#ip address 11.1.2.3/32
[local]Redback(config-if)#interface unnum2
[local]Redback(config-if)#ip unnumbered loop-lo2
```

The following example assigns an IPv6 address to the **enet1** interface:

```
[local]Redback(config-ctx)#interface enet1
[local]Redback(config-if)#ipv6 address 7001::1/64
```

## 2.3 Performing Interface Operations Tasks

Interface operations tasks are listed in Table 6. Enter **show** commands in any mode; enter all other commands in exec mode.

*Table 6 Interface Operations Tasks*

| Task | Root Command |
|---|---|
| Enable the generation of debug messages for all configured interfaces in the current context. | *debug if* |
| Display information about interfaces, including the interface bound to the Ethernet management port on the controller card. | *show ip interface* |
| Display the status of the IP addresses in the specified IP pool, in all IP pools in the specified interface, or in all IP pools in the current context or range. | *show ip pool* |