Huawei IN200 NIC

User Guide

Issue	06
Date	2019-07-27

HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://e.huawei.com

About This Document

Purpose

This document describes the IN200 PCIe NIC physical structure, features, specifications, and installation and management methods, as well as how to install and use management tools of the IN200.

Intended Audience

This document is intended for:

- Enterprise administrators
- Enterprise end users

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.

Symbol	Description
NOTE	Calls attention to important information, best practices and tips. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

Change History

Changes between document issues are cumulative. The latest document issue contains all the changes made in earlier issues.

Issue	Date	Description
06	2019-07-27	This issue is the sixth official release. Modified **1.5 System Requirements**.
05	2019-07-15	This issue is the fifth official release. ● Modified **1.3 Features**. ● Modified **1.5 System Requirements**. ● Modified the instance of installing and upgrading the NIC driver in **2.2 Maintaining the NIC Driver**. ● Added preparation before installing the RoCE Driver to **2.3 Maintaining the RoCE Driver**. ● Added **4 Configuring QoS**. ● Added **5 Configuring RoCE Bonding**. ● Added **6.1.4.44 Setting NIC Packet Data to Be Sent and Triggering NIC Packet Sending When AC Power Is Lost (reboot_notice)** ~ **6.1.4.47.3 Setting DWRR Scheduling (-t -p)**.
04	2019-05-10	This issue is the fourth official release. ● Added information about driver maintenance on Windows in **2.2 Maintaining the NIC Driver**. ● Added information about installing, upgrading, and uninstalling the hinicadm tool on Windows in **6 Management Tools**. ● Added **2.3 Maintaining the RoCE Driver**. ● Added **3 Configuring SR-IOV**.
03	2019-01-30	This issue is the third official release.

Issue	Date	Description
02	2018-08-10	This issue is the second official release.
01	2018-05-20	This issue is the first official release.

Contents

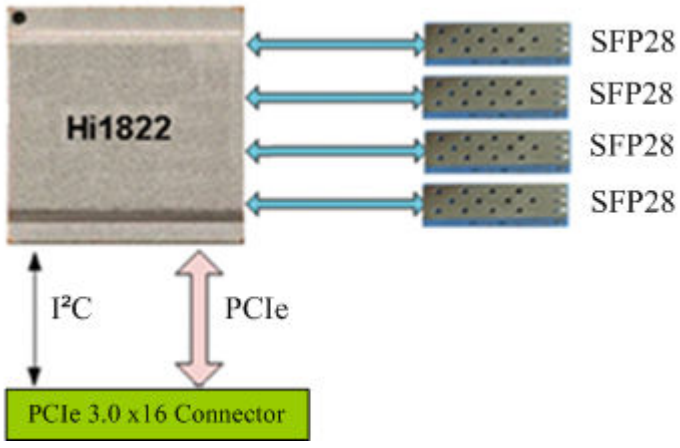
1 Getting to Know the IN200

1.1 Overview

The IN200 Ethernet NIC (IN200 for short) is a PCIe card for Huawei servers. It supports four 25GE SFP28 optical ports as external service ports.

The IN200 is a PCIe card based on the Huawei HiSilicon Hi1822 NIC chip. It supports PCIe 3.0 x16 and Inter-integrated Circuit (I^2C) channel, supports System Management Bus (SMBus), and Management Component Transport Protocol (MCTP) out-of-band management. **Figure 1-1** shows the architecture of the IN200.

Figure 1-1 IN200 architecture

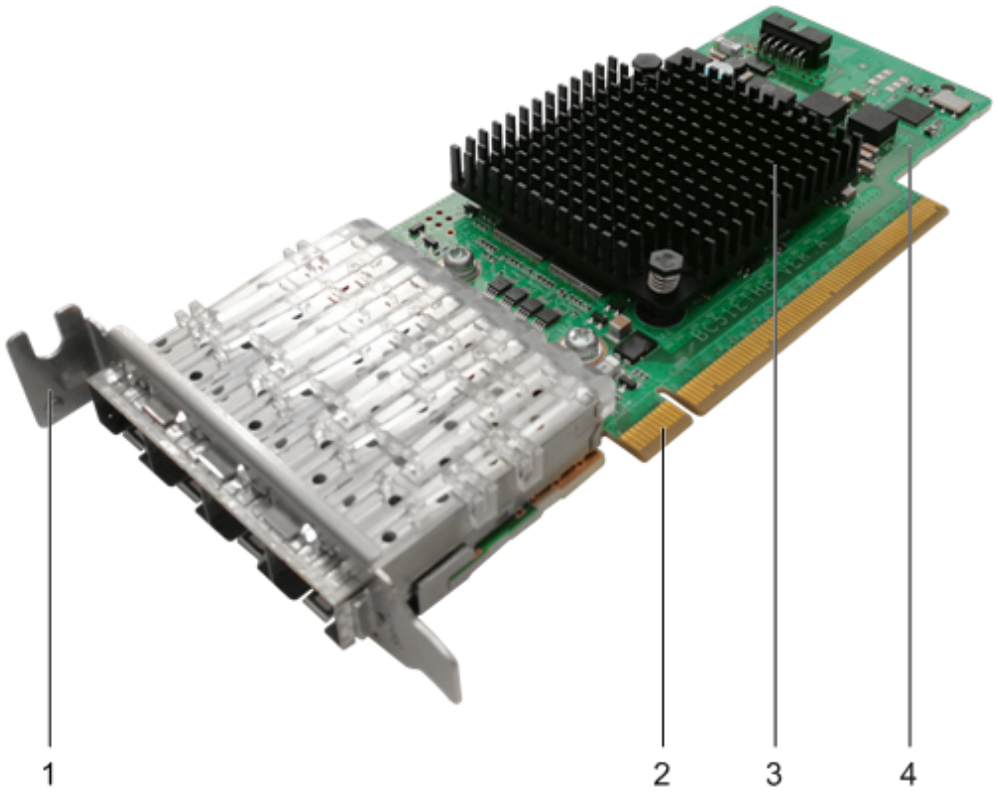


1.2 Physical Structure

Components

Figure 1-2 shows the components of the IN200.

Figure 1-2 IN200 components



1	Bracket	2	PCIe connector
3	Hi1822+heat sink	4	Mainboard

Table 1-1 describes the components of the IN200.

Table 1-1 IN200 component descriptions

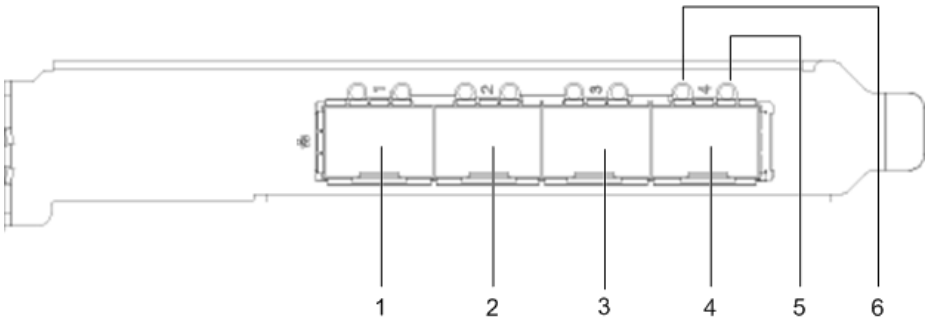
Component	Description
Mainboard	Includes a NIC module, network port module, and power module.
Hi1822	A NIC chip, implementing NIC functions.

Component	Description
Heat sink	Cools the NIC chip.
Bracket	A NIC bracket (half- or full-height).
PCIe connector	Connects to the PCIe slot of the server.

Panel

Figure 1-3 shows the panel and indicators on the IN200.

Figure 1-3 Panel



1	SFP28 optical port 1	2	SFP28 optical port 2
3	SFP28 optical port 3	4	SFP28 optical port 4
5	Active/Link indicator	6	Speed indicator

Indicators

The indicators display the working status of the IN200. **Table 1-2** describes the indicators on the IN200 panel.

Table 1-2 Indicator description

Indicator	Meaning	Color	Description
Active/Link indicator	Network connection status indicator	Green	● Off: No link is established. ● Steady on: A link is established and no data is being transmitte. ● Blinking: A link is established and data is being transmitted.

Indicator	Meaning	Color	Description
Speed indicator	Network data transmission status indicator	Yellow and green	● Off: No link is established. ● Steady yellow: A 10 Gbit/s link is established. ● Steady green: A 25 Gbit/s link is established.

1.3 Features

The features of the IN200 are as follows:

- Half-height half-length PCIe x16 card with a half-height or full-height bracket, applicable for various application scenarios.
- Huawei HiSilicon Hi1822 NIC chip, up to four 25GE SFP28 ports, and excellent compatibility with the x86 architecture.
- Supports 25GE, 10GE, and GE modes.
- Supports for IPv4 and IPv6, TCP Checksum Offload, UDP Checksum Offload, TCP Segmentation Offload (TSO), Large Receive Offload (LRO), and Receive Side Scaling (RSS).
- Supports Configuration of interrupt aggregation parameters and parameter self-adaption.
- Supports 802.1Q VLAN acceleration and filtering.
- Supports Virtual eXtensible Local Area Network (VXLAN)/Network Virtualization using Generic Routing Encapsulation (NVGRE) offload.
- Supports Pause frames, Priority-based Flow Control (PFC), and Enhanced Transmission Selection (ETS).
- Supports NetQueue.
- Supports Single-Root I/O Virtualization (SR-IOV).
- Supports PF-passthrough VMs.
- Supports PF hybrid mode, unicast list filtering, multicast list filtering, and full multicast mode.
- Supports VF unicast list filtering, multicast list filtering, and full multicast mode.
- Supports VF QinQ mode.
- Supports **Auto**, **Enable**, and **Disable** VF link status.
- Supports VF QoS configuration.
- Supports VF MAC address management.
- Supports VF spoofchk.
- Supports Virtual Ethernet Bridge (VEB), which allows internal exchange between functions.
- Supports RoCEv2 offload.
- Supports Preboot Execution Environment (PXE) in Unified Extensible Firmware Interface (UEFI) mode, VLAN configuration, secure boot, and port configuration in the Basic Input and Output System (BIOS).

- Supports Data Plane Development Kit (DPDK).
- Supports MCTP.
- Supports Collection of out-of-band chip logs.
- Supports NIC management tools on the CLI.
- Supports In-band one-click logs collection.
- Supports Loopback tests.
- Supports Port location indicators.
- Supports Ethernet port Forward Error Correction (FEC) mode configuration.
- Supports Ethernet port auto-negotiation.

NOTE

- The IN200 supports the Legacy and UEFI modes. The PXE is not supported in Legacy mode.
- The GE mode supports only GE optical modules, basic packet receiving and sending, and stateless offload, and does not support features such as SR-IOV, DPDK, and RoCE.

1.4 Technical Specifications

Table 1-3 lists the basic technical specifications of the IN200.

Table 1-3 Technical Specifications

Item	Specifications
Form factor	Low-profile NIC, supporting a full-height or half-height bracket
PCIe port	PCIe x16 port, compatible with x8, x4, x2, and x1; PCIe 3.0, compatible with 2.0, 1.0
NIC chip	Huawei HiSilicon Hi1822 NIC chip
Network port	Four Ethernet service ports (SFP28, 25GE/10GE)
IPv6	Supported
PXE	Supported (default) and supported for secure boot
Mean time between failures (MTBF)	174324 hours
Mean time to repair (MTTR)	180 seconds

1.5 System Requirements

Hardware Requirements

To use the IN200, a server must have a standard PCIe x16 slot.

Software Requirements

- Table 1-4 lists the OSs supported by the IN200 NIC.

Table 1-4 Supported OSs by the IN200 NIC

OS	x86 version	ARM version
BCLinux	-	7.6
CentOS	6.8/6.9/6.10/7.0/7.1/7.2/7.3/7.4/7.5/7.6	7.4/7.5/7.6
Citrix XenServer	7.1/7.2/7.3/7.4/7.5/7.6	-
NeoKylin	6.9/7.4	V5.0U5/V7.0U5/V7.0U6
Deepin	V15.5	V15.2/V15.5
Debian	9.6	-
OpenStack	9.0/10.0/11.0/12.0/13.0	-
Oracle	6.9/6.10/7.3/7.4/7.5/7.6	-
RHEL	6.9/6.10/7.0/7.1/7.2/7.3/7.4/7.5/7.6/8.0	ALT 7.3/ALT 7.4/ALT 7.5/ALT 8.0
SLES	11.3/11.4/12.0/12.1/12.2/12.3/12.4/15	12.3/12.4/15
Ubuntu	14.04.5 LTS/16.04 LTS/16.04.1 LTS/ 16.04.2 LTS/16.04.3 LTS/16.04.4 LTS/ 16.04.5 LTS/18.04 LTS/18.04.1 LTS/ 18.04.2 LTS	16.04.3 LTS/16.04.4 LTS/ 16.04.5 LTS/18.04 LTS/ 18.04.1 LTS/18.04.2 LTS
Euler OS	V2.0 SP2/V2.0 SP3/V2.0 SP7	V2.0 SP2/V2.0 SP3/V2.0 SP8
UVP	V2R5/V3R0	V2R5/V3R0
Vmware ESXi	6.0.3/6.5/6.5.1/6.5.2/6.7/6.7.1	-
Microsoft Windows	Windows Server 2012 R2/Windows Server 2016	-

- Table 1-5 lists the OSs supported by the IN200 RoCE.

Table 1-5 Supported OSs by the IN200 RoCE

OS	x86 version	ARM version
CentOS	-	7.6
RHEL	7.0/7.1/7.2/7.3/7.5/7.6	-
Ubuntu	16.04.4 LTS/18.4.1 LTS	18.4.2 LTS

OS	x86 version	ARM version
EulerOS	V2.0 SP3	V2.0 SP3/V2.0 SP8
UVP	V2R5/V3R0	V2R5

NOTE

The preceding OSs are for reference only. For details about the OSs that can be purchased, see the **Intelligent Computing Compatibility Checker** or consult the local Huawei sales representatives.

2 Installation and Maintenance

The IN200 is a standard PCIe card. Its hardware installation method is the same as that of a common NIC. For details about how to install the IN200, see the user guide of the target server. This chapter describes only the installation and maintenance of the IN200 driver and firmware.

2.1 Obtaining Software Packages

Downloading Installation Packages

Step 1 Log in to the **Huawei Enterprise Website**.

Step 2 Choose **TECHNICAL SUPPORT > Product Support** > **Accelerator Components > IN500 Solution**.

Step 3 Click the **software Download** tab.

Step 4 Click the target version.

Step 5 Download the IN200 software package (**IN500_solution_5.1.0.zip**).

NOTE

The IN200/IN300/IN500 software package is named **IN500_solution_5.1.0.zip** or **IN500_solution_5.1.0.SPCXXX.zip**.

Step 6 Decompress the software package.**Table 2-1** lists the software packages required.

Table 2-1 Required software package

Software Package Type	OS Type	Path	Format	Installation Method
NIC Driver package	Linux	driver\linux\nic*OS name*\	*.rpm, or *.deb	**2.2 Maintaining the NIC Driver**
	Vmware	driver\vmware\nic*OS name*\ **NOTE** The driver package contains the hinicadm tool package. After the driver is installed, the hinicadm tool is automatically installed.	*.vib	
	Microsoft Windows	driver\windows\nic*OS name*\	*.msi	
RoCE Driver package	-	driver\linux\roce*OS name*\	*.rpm	**2.3 Maintaining the RoCE Driver**
Firmware upgrade package	-	firmware\update_bin\cfg_data_nic_prd_1h_4x25G\	*.bin	**2.4 Upgrading the IN200 Firmware**
Firmware log offline parsing dictionary file	-	firmware\dictionary	*.index	-
Hinicadm tool package	Linux	tools\linux\nic\	*.rpm	**6.1 Customized Management Tool - hinicadm**
		tools\linux\nic\collect_scripts **NOTE** This is a one-click information collection script, which is also applicable to the ARM platform.	*.sh	
	Linux_arm	tools\linux_arm\nic\	*.rpm or *.deb	
		tools\linux_arm\nic\collect_scripts	*.sh	

Software Package Type	OS Type	Path	Format	Installation Method
	Microsoft Windows	tools\windows\nic	*.msi	

----End

NOTE

- You are advised to use the latest driver, firmware, and management tool released on Huawei enterprise service website (**https://e.huawei.com/en/**).
- Ensure that the version of the firmware used for installation or upgrade is not earlier than the driver version. For example, if the driver version is 1.8.2.7, the firmware version must be 1.8.2.7 or a later.
- After the driver is upgraded, you must also upgrade the firmware and management tool.

Verifying Installation Package Integrity

Verify that the obtained installation packages are the same as those at the website.

On the download page, click  to obtain the digital software certificate, and click  to download the software.

Obtain the verification tool and method from **Digital Signature Verification Tool**.

(Optional) Obtain the SUSE gpg public keytaining SUSE Linux Certificate and Public Key

The driver packages of SUSE Linux are certificated. Before installing the driver, install the corresponding certificate and public key on the server.

- **Obtain the SUSE Linux PLDP certificate**
- **Obtain the SUSE gpg public key**

2.2 Maintaining the NIC Driver

2.2.1 Preparations (SUSE Linux)

The IN200 has been certificated by SUSE. Before installing, or upgrading drivers for SUSE Linux, you need to import the SUSE Linux PLDP UEFI certificate (mandatory in the Secure Boot mode) and gpg public key of the installation package.

(Optional) Importing the SUSE PLDP Certificate

Before installing the IN200 driver on the server in the BIOS UEFI safe mode, import the SUSE PLDP UEFI certificate in the BIOS to support the certificated IN200 driver.

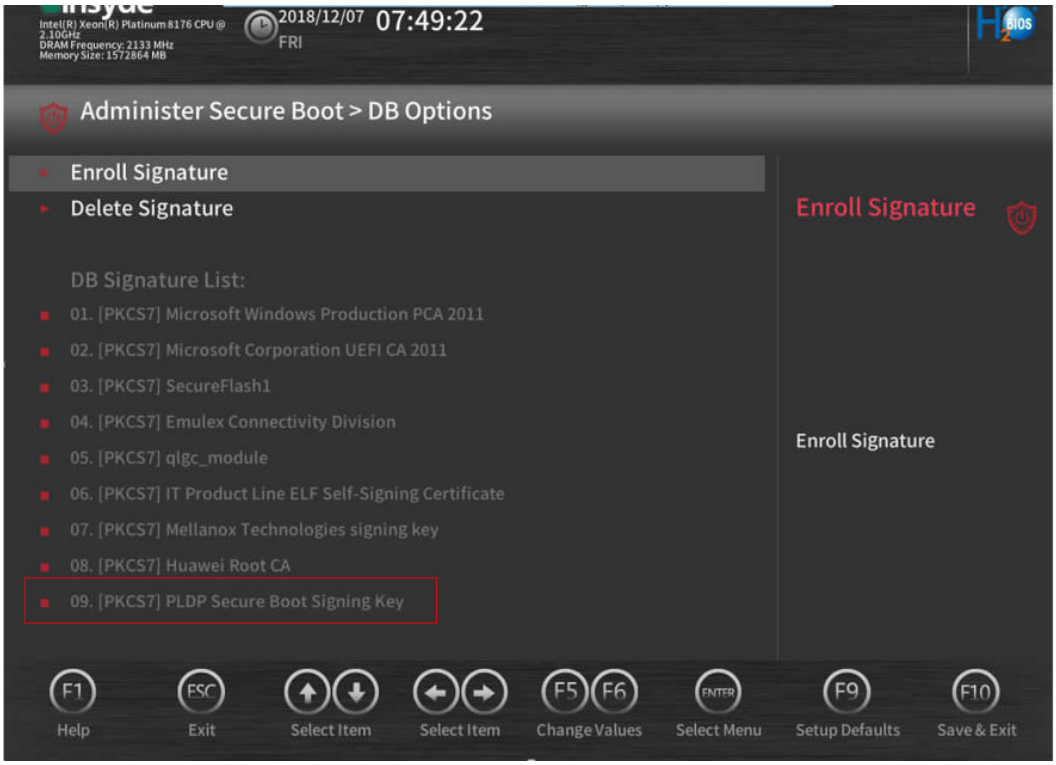
Step 1 Obtain the SUSE Linux PLDP UEFI certificate.

For details about the certificate address, see **2.1 Obtaining Software Packages**.

Step 2 Import the SUSE Linux PLDP UEFI certificate to the BIOS.

On the BIOS screen, choose **Administer Secure Boot** > **DB Options** > **Enroll Signature** and import the SUSE LDAP UEFI certificate. After the certificate is imported, **PLDP Secure Boot Signing Key** is displayed in the **DB Signature List**, as shown in **Figure 2-1**.

Figure 2-1 Importing a certificate



For details, see the server BIOS parameter reference.

----End

Importing the SUSE gpg Public Key

NOTICE

You only need to import the gpg public key of the SUSE installation package once.

Step 1 Obtain the SUSE gpg public key.

For details about the certificate address, see **2.1 Obtaining Software Packages**.

Step 2 Upload the public file such as **gpg-pubkey-c2bea7e6-4c2de264.asc** to any directory of the operating system on the server.

Step 3 Run the **rpm --import** command to import the public key to the operating system.

```
rpm --import gpg-pubkey-c2bea7e6-4c2de264.asc
```

----End

2.2.2 Installing the Driver

Prerequisites

- The driver package of the IN200 has been downloaded.

  The NIC driver package is included in the IN200 software package. For details about how to obtain the IN200 package, see **2.1 Obtaining Software Packages**.
- To upgrade the driver corresponding to SUSE Linux, ensure that the preparations are ready. For details, see **2.2.1 Preparations (SUSE Linux)**.
- The driver package has been uploaded to the server OS.

Impact

It takes about 30 seconds to install the driver. The installation process cannot be interrupted. During the installation, the OS cannot be restarted. Otherwise, the OS may be abnormal or cannot be started.

Installing the Driver on Linux

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the directory where the driver package is stored.

Step 3 Run the command to install the driver package.

- RHEL, CentOS, Oracle, and SUSE

  Run the **rpm -ivh <*driver software package name*>** command.

  Using RHEL as an example:

```
[root@localhost]# rpm -ivh kmod-hinic-1.8.2.8_3.10.0_957-1.el7.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:kmod-hinic-1.8.2.8_3.10.0_957-1.e################################# [100%]
```

- Ubuntu and Debian

  NOTE

  The NIC driver installation depends on the module-init-tools component. If the module-init-tools component is not installed on the OS, download and install the module-init-tools component based on the OS version from the official Ubuntu/Debian website.

  1. Download the module-init-tools component.
     - **Official Ubuntu website download address**
     - **Official Debian website download address**
  2. Upload the **module-init-tools** component file to the OS by referring to **7.3 Transferring a File Using the Virtual Directory**.
  3. Run the following command to install the module-init-tools component:

     dpkg -i module-init-tools.deb

  Run the **dpkg -i <*driver software package name*>** command.

  Using Ubuntu as an example:

```
root@ubuntu1804:/home/ubuntu # dpkg -i
hinic-1.6.1.1-4.15.0_20_generic.ubuntu.arm64.deb
Selecting previously unselected package hinic.
(Reading database ... 69638 files and directories currently installed.)
```

```
Preparing to unpack hinic-1.6.1.1-4.15.0_20_generic.ubuntu.arm64.deb ...
Unpacking hinic (1.6.1.1) ...
Setting up hinic (1.6.1.1) ...
Installing... Please wait for a moment.
Install hinic driver package successfully.
```

Step 4 Make the driver take effect.

You can select either of the following methods:

- Method 1

  Run the **reboot** command on the OS.
- Method 2

  Run the following commands in sequence in the OS:

  rmmod hinic

  modprobe hinic

----End

Installing the Driver on VMware ESXi

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the directory where the driver package is stored, for example, **/tmp.**

Step 3 Install the driver package.

Run the **esxcli software vib install -v** *hinic-<version>-<kernel_version>.<arch>.vib* command.

NOTE

Enter a full path after **-v**.

Example:

```
[root@localhost:~] esxcli software vib install -v /hinic-1.6.2.2-1OEM.
650.0.0.4598673.x86_64.vib
Installation Result
   Message: The update completed successfully, but the system needs to be
rebooted for the changes to be effective.
   Reboot Required: true
   VIBs Installed: Huawei_bootbank_hinic_1.6.2.2-1OEM.650.0.0.4598673
   VIBs Removed:
   VIBs Skipped:
```

Step 4 Run the **reboot** command to restart VMware ESXi for the driver to take effect.

----End

Installing the Driver on Microsoft Windows

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

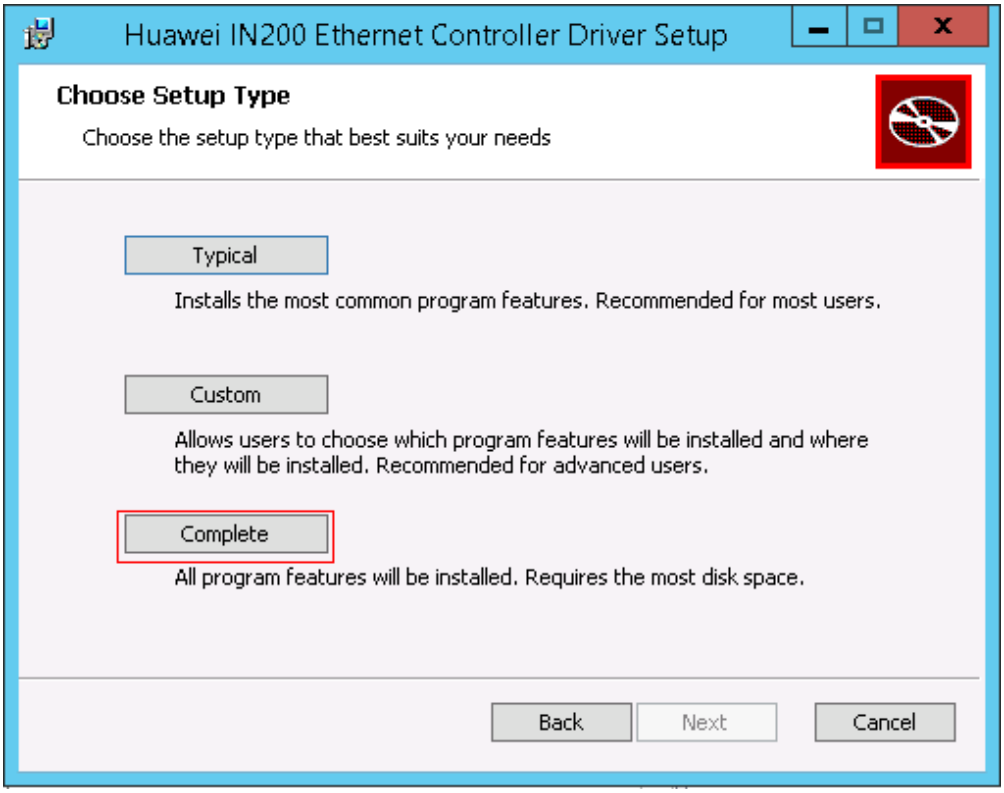
Step 2 Go to the directory where the driver package is stored, for example, "C:\".

Step 3 Install the driver package.

Double-click ***hinic_<version>_<Windows_OS>_x86_64.msi*** to install the driver package.

NOTE

Select **Complete** for **Choose Setup Type**.



----End

NOTICE

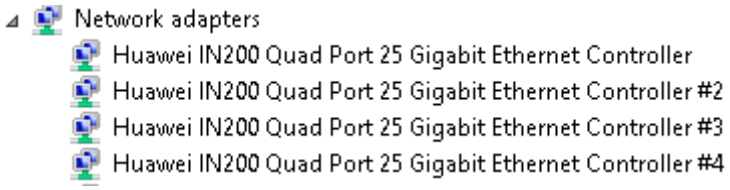
If an exception occurs during the installation (for example, the server is restarted or powered off), handle the problem by referring to **A.1 An Exception Occurs During Driver Installation or Uninstallation**.

Follow-up Procedure

After the driver is installed, you can operate the following steps.

1. Run the **lsmod | grep hinic** command on Linux, or run the **vmkload_mod -l | grep hinic** command on VMware ESXi to check whether the driver has taken effect.
   - If no command output is displayed, the driver has not taken effect, and you need to perform the operation again.
   - If the command output about the IN200 driver is displayed, the driver has taken effect.
2. Run the **version** command to query the current driver version using hinicadm tool.
3. In Microsoft Windows, you can scan for hardware detection changes in the Device Manager and check whether the driver has been installed on the IN200 and whether the driver has been identified.

For example:



2.2.3 Upgrading the Driver

Prerequisites

- The driver of the IN200 already exists in the server OS.
- The driver package of the IN200 has been downloaded.

  The NIC driver is included in the IN200 software package. For details about how to obtain the IN200 package, see **2.1 Obtaining Software Packages**.
- To upgrade the driver corresponding to SUSE Linux, ensure that the preparations are ready. For details, see **2.2.1 Preparations (SUSE Linux)**
- The driver package has been uploaded to the server OS.

Impact

The upgrade process cannot be interrupted. During the upgrade, the OS cannot be restarted. Otherwise, the OS may be abnormal or cannot be started.

Upgrading the Driver on Linux

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the directory where the driver package is stored.

Step 3 Upgrade the driver.

- RHEL, CentOS, Oracle, and SUSE

  Run the **rpm -Uvh <*driver software package name*>** command.

  Example:

```
[root@localhost]# rpm -Uvh kmod-hinic-2.3.0.0_3.10.0_957-1.el7.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:kmod-hinic-2.3.0.0_3.10.0_957-1.e################################# [ 50%]
Cleaning up / removing...
   2:kmod-hinic-1.8.2.8_3.10.0_957-1.e################################# [100%]
```

- Ubuntu and Debian

  Run the **dpkg -i <*driver software package name*>** command.

  Example:

```
root@ubuntu1804:/home/ubuntu # dpkg -i
hinic-1.6.1.2-4.15.0_20_generic.ubuntu.arm64.deb
(Reading database ... 69641 files and directories currently installed.)
Preparing to unpack hinic-1.6.1.2-4.15.0_20_generic.ubuntu.arm64.deb ...
Unpacking hinic (1.6.1.2) over (1.6.1.1) ...
Uninstalling... Please wait for a moment.
Uninstall hinic driver package successfully.
Setting up hinic (1.6.1.2) ...
```



```
Installing... Please wait for a moment.
Install hinic driver package successfully.
```

Step 4 Make the new driver take effect.

You can select either of the following methods:

- Method 1

  Run the **reboot** command on the OS.
- Method 2:

  a. Check that the current IN200 program has stopped.

  b. Run the **rmmod hinic** on the OS to install the existing driver.

  c. Run the **modprobe hinic** command on the OS to load the new driver.

----End

Upgrading the Driver on VMware ESXi

To update the driver, perform operations in **2.2.2 Installing the Driver**. The system automatically replaces the original vib driver package based on the version and restarts for the update to take effect.

Upgrading the Driver on Microsoft Windows

Step 1 Log in to the server OS.

For details, see **A.1 An Exception Occurs During Driver Installation or Uninstallation**.

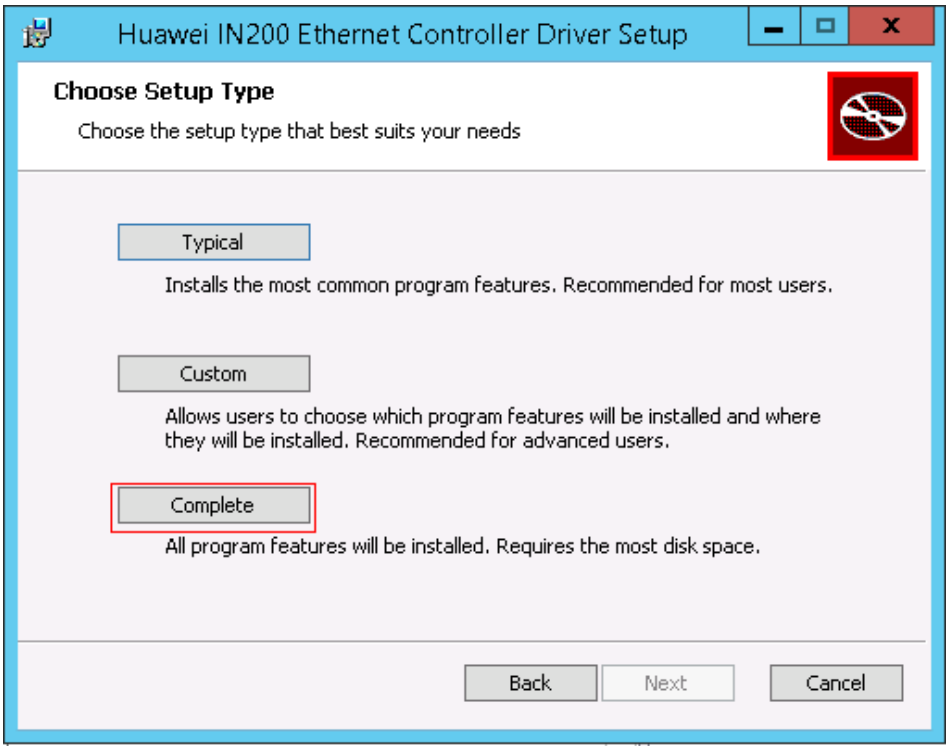
Step 2 Go to the directory where the driver package is stored, for example, "C:\".

Step 3 Upgrade the driver package.

Double-click ***hinic_<version>_<Windows_OS>_x86_64.msi*** to install the driver package.

NOTE

- Select **Complete** for **Choose Setup Type**.



- The target version must be later than the source version; otherwise, uninstall the driver package and install a later one.

----End

Follow-up Procedure

After the driver is installed, you can run the **version** command to query the current driver version using hinicadm tool.

2.2.4 Uninstalling the Driver

Prerequisites

- The driver of the IN200 already exists on the OS.
- To upgrade the driver corresponding to SUSE Linux, ensure that the preparations are ready. For details, see **2.2.1 Preparations (SUSE Linux)**.

Impact

During the uninstallation, the OS cannot be restarted. Otherwise, the OS may be abnormal or cannot be started.

Uninstalling the Driver from Linux

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the installation directory of the driver.

Step 3 Uninstall the driver.

- For RHEL, CentOS, and Oracle, run the **rpm -e kmod-hinic** command.
- For SUSE, run the **rpm -e hinic-kmp-<kernel feature>** command, for example: **rpm -e hinic-kmp-default**.

  NOTE

  You can run the **rpm -qa | grep hinic** command to query the driver software package name.
- For Ubuntu and Debian, run the **dpkg -r hiodriver** command.

Step 4 Make the uninstallation operation take effect.

You can select either of the following methods:

- Method 1

  Run the **reboot** command on the OS.
- Method 2

  a. Check that the current IN200 program has stopped.

  b. Run the **rmmod hinic** command on the OS.

----End

Uninstalling the Driver from VMware ESXi

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the directory where the driver package is installed.

Step 3 Uninstall the driver.

For example, to uninstall the driver, run the **esxcli software vib remove -n** *hifc* command.

```
[root@localhost:~] esxcli software vib remove -n hinic
Removal Result
   Message: The update completed successfully, but the system needs to be
rebooted for the changes to be effective.
   Reboot Required: true
   VIBs Installed:
   VIBs Removed:Huawei_bootbank_hinic_1.6.2.1-1OEM.650.0.0.4598673
   VIBs Skipped:
```

Step 4 Run the **reboot** command to restart VMware ESXi.

----End

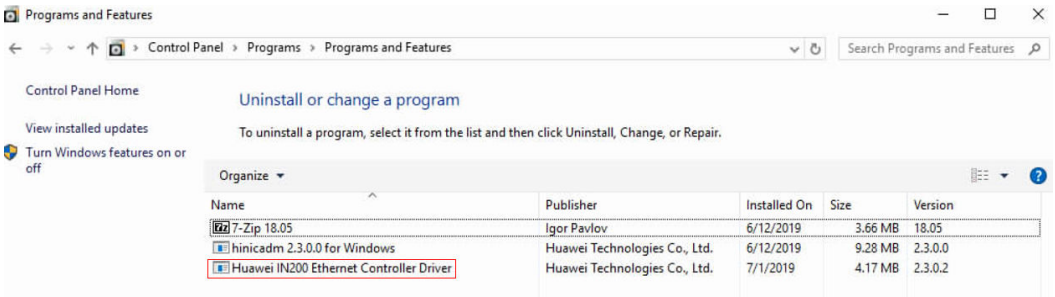
Uninstalling the Driver from Microsoft Windows

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Click , choose **Control Panel** > **Programs and Features**.

Step 3 Right-click the **HinicDriver** program name for example "Huawei IN200 Ethernet Controller Driver", choose **Uninstall/Change** from the short-cut menu.



----End

NOTE

If an exception occurs during the uninstallation (for example, the server is restarted or powered off), handle the problem by referring to **A.1 An Exception Occurs During Driver Installation or Uninstallation**.

2.3 Maintaining the RoCE Driver

2.3.1 Preparing OSs Supported by Huawei In-House OFED

Before running RDMA over Converged Ethernet (RoCE) services on the IN200, you need to install the OpenFabrics Enterprise Distribution (OFED) and RoCE drivers in advance.

OFED software packages are classified into Huawei in-house and non-Huawei OFED software packages. The Huawei in-house OFED package is included in the RoCE driver package. **Table 2-2** lists the OSs supported by the OFED.

Table 2-2 Preparing OSs supported by Huawei in-house OFED

OS	x86	ARM
CentOS	7.4, 7.5, and 7.6	7.5 and 7.6
RHEL	7.5 and 7.6	-
Ubuntu	16.04.4 and 18.04.1	18.04.2
EulerOS	V200R007C00SPC500B005	V200R008C00SPC100B220

2.3.2 Installing the Driver

Prerequisites

- The RoCE driver package has been downloaded.

  The RoCE driver package is included in the IN200 software package. For details about how to obtain the IN200 package, see **2.1 Obtaining Software Packages**.
- The IN200 driver has been installed.

  For details, see **2.2 Maintaining the NIC Driver**.

- The driver package has been uploaded to the server OS.

Impact

- If a non-OFA OFED 4.8-2 package (for example, an OFED package from a third party or an open-source OFED package of a different version) has been installed on the server, you need to uninstall the existing OFED package and then install the OFA OFED 4.8-2 package and IN200 RoCE driver before using the IN200 RoCE function (the unistallation and installation operations do not conflict with the IN200 NIC functions).
- It takes about 30 seconds to install the driver. The installation process cannot be interrupted. During the installation, the OS cannot be restarted. Otherwise, the OS may be abnormal or cannot be started.

On the OS Supported by Huawei In-House OFED

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Run either **rpm -qa | grep rdma** and **rpm -qa | grep libib** commands or **dpkg -l | grep rdma** and **dpkg -l | grep libib** commands to check whether the OFED has been installed.

- If the OFED is installed, go to **Step 3**.
- If the OFED is not installed, go to Step **Step 4**.

Step 3 Run the **rpm -e** *X* or **dpkg -r** *X* command to uninstall the original OFED package. In the command, *X* indicates the names of all OFED packages displayed in the command output in **Step 2**.

Step 4 Run the following command in the directory where the RoCE driver package is stored to decompress the OFED package:

tar -xvf OFED-*.tar.gz

Step 5 Run the following command to go to the OFED package directory:

cd OFED-*

Step 6 Run the following commands to install the OFED package:

bash ofed_install.sh

The command output is as follows:

```
2019-07-10 09:30:43
2019-07-10 09:30:43
2019-07-10 09:30:43 [INSTALL] install OFED software start.
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibverbs*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibverbs-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibumad*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest-debuginfo*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest*rpm ok
This program will install the OFED package on your machine.
Note that all other Mellanox, OEM, OFED, RDMA or Distribution IB packages will be
removed.
Those packages are removed due to conflicts with OFED, do not reinstall them.
Do you want to continue?[y/N]:y
```



```
.....................................................................................................
.....................................................................................................
Preparing...                          ################################# [100%]
Updating / installing...
   1:perftest-debuginfo-4.4-0.5.g1ceab################################# [ 50%]
   2:perftest-4.4-0.5.g1ceab48        ################################# [100%]
2019-07-10 09:33:26 [INSTALL] install_rpm perftest ok
2019-07-10 09:33:26 [INSTALL] install driver   -----------------pass
2019-07-10 09:33:26 [INSTALL] install OFED software ok.
To load the new driver, run:
/etc/init.d/openibd restart
```

Step 7 Install the RoCE driver in the directory where the RoCE driver package is stored.

- On the CentOS, SUSE, EulerOS, or RHEL:

  Run the **rpm -ivh**<*name of the driver software package* command to install the driver package.

  The following commands use CentOS 7.4 as an example:

  ```
  [root@localhost driver]# rpm -ivh
  hiroce-2.3.1.0_3.10.0_693.el7.x86_64-1.el7.centos.x86_64.rpm
  Preparing...                          ################################
  [100%]
  Updating / installing...
    1:hiroce-2.3.1.0_3.10.0_693.el7.x86################################# [100%]
  ```

- On the Ubuntu:

  NOTE

  The NIC driver installation depends on the module-init-tools component. If the module-init-tools component is not installed on the OS, download and install the module-init-tools component based on the OS version from the official Ubuntu/Debian website.

  1. Download the module-init-tools component.
     - **Official Ubuntu website download address**
     - **Official Debian website download address**
  2. Upload the **module-init-tools** component file to the OS by referring to **7.3 Transferring a File Using the Virtual Directory**.
  3. Run the following command to install the module-init-tools component:

     dpkg -i module-init-tools.deb

  Run the **dpkg -i**<*name of the driver software package* command to install the RoCE driver.

  The following commands use Ubuntu 18.04.1 as an example:

  ```
  root@ubuntu18041:/home/ubuntu # dpkg -i
  hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb
  Selecting previously unselected package hiroce.
  (Reading database ... 80445 files and directories currently installed.)
  Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...
  Unpacking hiroce (2.3.0.2) ...
  Setting up hiroce (2.3.0.2) ...
  Installing... Please wait for a moment.
  Install hiroce driver package successfully.
  ```

Step 8 Make the driver take effect.

Run the following commands on the OS.

modprobe hiroce

service network restart

NOTE

This method takes effect temporarily. If you require the OS to automatically load the driver upon startup, add the preceding two commands to the automatic startup script of the OS.

----End

On the OS Supported by Non-Huawei OFED

Step 1 Log in to the OS of the standby server.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Run either **rpm -qa | grep rdma** and **rpm -qa | grep libib** commands or **dpkg -l | grep rdma** and **dpkg -l | grep libib** commands to check whether the OFED has been installed.

- If the OFED is installed, go to **Step 3**.
- If the OFED is not installed, go to **Step 4**.

Step 3 Run the **rpm -e** *X* or **dpkg -r** *X* command to uninstall the original OFED package. In the command, *X* indicates the names of all OFED packages displayed in the command output in **Step 2**.

Step 4 Click **OFA OFED-4.8-2** to download the OFA OFED package of the 4.8-2 version.

Step 5 Compile the **rpm/deb** package (including **compat-rdma** and **rdma-core**) by referring to **README.txt** in the OFA OFED package.

Step 6 Upload the **rpm/deb** package to the OS by referring to 7.3 Transferring a File Using the Virtual Directory.

Step 7 Run the**rpm -ivh** **.rpm* or **dpkg -i** **.deb* command to install the OFA OFED V4.8-2.

Step 8 Install the RoCE driver in the directory where the RoCE driver package is stored.

- On the CentOS, SUSE, EulerOS, or RHEL:

  Run the **rpm -ivh**<*name of the driver software package* command to install the driver package.

  The following commands use RHEL 7.3 as an example:

```
[root@localhost driver]# rpm -ivh
hiroce-2.3.1.0_3.10.0_514.el7.x86_64-1.el7.x86_64.rpm
Preparing...                          ################################
[100%]
Updating / installing...
   1:hiroce-2.3.1.0_3.10.0_514.el7.x86################################ [100%]
```

- On the Ubuntu:

NOTE

The NIC driver installation depends on the module-init-tools component. If the module-init-tools component is not installed on the OS, download and install the module-init-tools component based on the OS version from the official Ubuntu/Debian website.

1. Download the module-init-tools component.
   - **Official Ubuntu website download address**
   - **Official Debian website download address**
2. Upload the **module-init-tools** component file to the OS by referring to **7.3 Transferring a File Using the Virtual Directory**.
3. Run the following command to install the module-init-tools component:

   dpkg -i module-init-tools.deb

Run the **dpkg -i**<*name of the driver software package* command to install the RoCE driver.

The following commands use Ubuntu 18.04.1 as an example:

```
root@ubuntu18041:/home/ubuntu # dpkg -i
hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb
Selecting previously unselected package hiroce.
(Reading database ... 80445 files and directories currently installed.)
Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...
Unpacking hiroce (2.3.0.2) ...
Setting up hiroce (2.3.0.2) ...
Installing... Please wait for a moment.
Install hiroce driver package successfully.
```

Step 9 Make the driver take effect.

Run the following commands on the OS.

modprobe hiroce

service network restart

NOTE

This method takes effect temporarily. If you require the OS to automatically load the driver upon startup, add the preceding two commands to the automatic startup script of the OS.

----End

Follow-up Procedure

Check whether the driver has taken effect after the installation.

1. Run the **lsmod | grep hiroce** command on the OS.
   - If no command output is displayed, the driver has not taken effect. Reinstall the driver.
   - If the command output about the IN200 driver is displayed, the driver has taken effect.
2. Run the **hiroce gids** command, view the command output, and check whether the driver takes effect.
   - If the command output is displayed, the driver has taken effect.
   - If no command output is displayed, the driver has not taken effect, and you need to perform the operation again.

2.3.3 Upgrading the Driver

Prerequisites

- The latest RoCE driver package has been downloaded.

  The RoCE driver package is included in the IN200 software package. For details about how to obtain the IN200 package, see **2.1 Obtaining Software Packages**.
- The IN200 driver has been installed.

  For details, see **2.2 Maintaining the NIC Driver**.
- The RoCE driver to be upgraded has been installed.
- The driver package has been uploaded to the server OS.

Constraints

- The target version must be later than the source version.
- If the target driver version is earlier than or the same as the source driver version, the new driver does not take effect after you run the **rpm -Uvh** command to upgrade the driver.
- To roll back to the earlier version, uninstall the driver by referring to **2.3.4 Uninstalling the Driver** and then install the driver of the earlier version by referring to **2.3.2 Installing the Driver**.

Impact

The upgrade process cannot be interrupted. During the upgrade, the OS cannot be restarted. Otherwise, the OS may be abnormal or cannot be started.

On the OS Supported by Huawei In-House OFED

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Run the following command in the directory where the RoCE driver package is stored to decompress the OFED package:

tar -xvf OFED-*.tar.gz

Step 3 Run the following command to go to the OFED package directory:

cd OFED-*

Step 4 Run the following commands to install the OFED package:

bash ofed_install.sh

The command output is as follows:

```
[root@localhost OFED]# bash ofed_install.sh
2019-07-10 09:30:43
2019-07-10 09:30:43
2019-07-10 09:30:43 [INSTALL] install OFED software start.
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/compat-rdma*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core-devel*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/rdma-core*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibverbs*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibverbs-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/libibumad*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/librdmacm-utils*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest-debuginfo*rpm ok
2019-07-10 09:30:43 [INSTALL] check RPMS/perftest*rpm ok
This program will install the OFED package on your machine.
Note that all other Mellanox, OEM, OFED, RDMA or Distribution IB packages will be
removed.
Those packages are removed due to conflicts with OFED, do not reinstall them.
Do you want to continue?[y/N]:y
.....................................................................................................................
.....................................................................................................................
Preparing...                          ################################# [100%]
Updating / installing...
   1:perftest-debuginfo-4.4-0.5.g1ceab################################# [ 50%]
   2:perftest-4.4-0.5.g1ceab48        ################################# [100%]
2019-07-10 09:33:26 [INSTALL] install_rpm perftest ok
```



```
2019-07-10 09:33:26 [INSTALL] install driver   -----------------pass
2019-07-10 09:33:26 [INSTALL] install OFED software ok.
To load the new driver, run:
/etc/init.d/openibd restart
```

Step 5 Install the RoCE driver in the directory where the RoCE driver package is stored.

- On the CentOS, SUSE, EulerOS, or RHEL:

  Run the **rpm -Uvh**<*name of the driver software package* command to upgrade the driver package.

  The following commands use CentOS 7.3 as an example:

```
[root@localhost driver]# rpm -Uvh
hiroce-2.3.1.0_3.10.0_514.el7.x86_64-1.el7.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:hiroce-2.3.1.0_3.10.0_514.el7.x86################################# [100%]
```

  - On the Ubuntu:

    Run the **dpkg -i**<*name of the driver software package* command to upgrade the RoCE driver.

    The following commands use Ubuntu 18.04.1 as an example:

```
root@ubuntu18041:/home/ubuntu # dpkg -i
hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb
Selecting previously unselected package hiroce.
(Reading database ... 80445 files and directories currently installed.)
Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...
Unpacking hiroce (2.3.0.2) ...
Setting up hiroce (2.3.0.2) ...
Installing... Please wait for a moment.
Install hiroce driver package successfully.
```

Step 6 Make the driver take effect.

Run the following commands on the OS.

rmmod hiroce

modprobe hiroce

NOTE

This method takes effect temporarily. If you require the OS to automatically load the driver upon startup, add the preceding two commands to the automatic startup script of the OS.

----End

On the OS Supported by Non-Huawei OFED

Step 1 Log in to the OS of the standby server.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Click **OFA OFED-4.8-2** to download the OFA OFED package of the 4.8-2 version.

Step 3 Compile the **rpm/deb** package (including **compat-rdma** and **rdma-core**) by referring to **README.txt** in the OFA OFED package.

Step 4 Upload the **rpm/deb** package to the OS by referring to 7.3 Transferring a File Using the Virtual Directory.

Step 5 Run the**rpm -ivh** **.rpm* or **dpkg -i** **.deb* command to install the OFA OFED V4.8-2.

Step 6 Install the RoCE driver in the directory where the RoCE driver package is stored.

- On the CentOS, SUSE, EulerOS, or RHEL:

  Run the **rpm -Uvh**<*name of the driver software package* command to upgrade the driver package.

  The following commands use RHEL 7.3 as an example:

```
[root@localhost driver]# rpm -Uvh
hiroce-2.3.1.0_3.10.0_514.el7.x86_64-1.el7.x86_64.rpm
Preparing...                          ################################
[100%]
Updating / installing...
1:hiroce-2.3.1.0_3.10.0_514.el7.x86################################ [100%]
```

- On the Ubuntu:

  Run the **dpkg -i**<*name of the driver software package* command to upgrade the RoCE driver.

  The following commands use Ubuntu 18.04.1 as an example:

```
root@ubuntu18041:/home/ubuntu # dpkg -i
hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb
Selecting previously unselected package hiroce.
(Reading database ... 80445 files and directories currently installed.)
Preparing to unpack hiroce-2.3.0.2-4.15.0_29_generic.ubuntu.amd64.deb ...
Unpacking hiroce (2.3.0.2) ...
Setting up hiroce (2.3.0.2) ...
Installing... Please wait for a moment.
Install hiroce driver package successfully.
```

Step 7 Make the driver take effect.

Run the following commands on the OS.

rmmod hiroce

modprobe hiroce

NOTE

This method takes effect temporarily. If you require the OS to automatically load the driver upon startup, add the preceding two commands to the automatic startup script of the OS.

----End

2.3.4 Uninstalling the Driver

Prerequisites

The RoCE driver already exists on the OS.

Impact

During the uninstallation, the OS cannot be restarted. Otherwise, the OS may be abnormal or cannot be started.

Procedure

NOTICE

Before uninstallation, all applications that use the RoCE must be stopped. Otherwise, the driver will be occupied and the uninstallation will fail.

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the installation directory of the driver.

Step 3 Uninstall the driver.

- On the CentOS, SUSE, EulerOS, or RHEL, run the **rpm -e hiroce** command to uninstall the driver.

NOTE

You can run the **rpm -qa | grep hiroce** command to query the name of the driver software package.

- On the Ubuntu or Debian, run the **dpkg -r hiroce** command to uninstall the driver.

NOTE

You can run the **dpkg -l | grep hiroce** command to query the name of the driver software package.

Step 4 (Optional) Uninstall the OFED driver.

- On the OS supported by Huawei in-house OFED, run the **bash uninstall.sh** command to uninstall the OFED driver.
- On the OS supported by non-Huawei OFED, run the **rpm -e** *X* or **dpkg -r** *X* command to uninstall the OFED driver.

Step 5 Make the uninstallation operation take effect.

You can select either of the following methods:

- Method 1

  Run the **reboot** command on the OS.
- Method 2

  a. Check that RoCE services of the IN200 have been stopped.

  b. Run the **rmmod hinic** command on the OS.

----End

2.4 Upgrading the IN200 Firmware

Prerequisites

- The IN200 driver has been installed.
- The firmware upgrade package of the IN200 driver has been downloaded.

  The firmware upgrade package is included in the IN200 software package. For details about how to obtain the IN200 package, see **2.1 Obtaining Software Packages**.

- The upgrade package has been uploaded to the server OS.
- The hinicadm tool has been installed.

Impact

During the upgrade, the Linux, VMware ESXi, or Microsoft Windows cannot be restarted. Otherwise, the Linux, VMware ESXi, or Microsoft Windows may be abnormal or cannot be started.

Procedure

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the directory where the firmware upgrade package is stored.

Step 3 Upgrade the firmware and make the upgrade take effect.

1. Run the **hinicadm updatefw -i <*NIC device name*> -f <*Firmware file path*>** command.

   In the preceding command, ***NIC device name*** indicates the name of the NIC in the system. For example, **hinic0** indicates the first NIC, and **hinic1** indicates the second NIC.

   Example:

   ```
   # hinicadm updatefw -i hinic0 -f Hi1822_nic_prd_1h_4x25G.bin
   Please do not remove driver or network device
   Loading...
   [>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>]  [100%] [\]
   Loading firmware image succeed.
   Please reboot OS to take firmware effect.
   ```

2. Run the **reboot** command to restart the OS.

----End

Follow-up Procedure

After the firmware upgrade is complete, you can run the **hinicadm version -i hinic***X* command to query the IN200 firmware version to confirm that the upgrade is successful.

3 Configuring SR-IOV

When the IN200 works in an SR-IOV environment, enable SR-IOV in the kernel to ensure good performance.

3.1 x86 Version

3.1.1 Configuring the Server BIOSs

The following uses the BIOS of the 2288H V5 as an example to describe how to enable the SR-IOV function in the BIOS.

NOTE

Set the server boot mode to the **UEFI** mode before performing this operation.

Step 1 Log in to the real-time server desktop using the Remote Virtual Console.

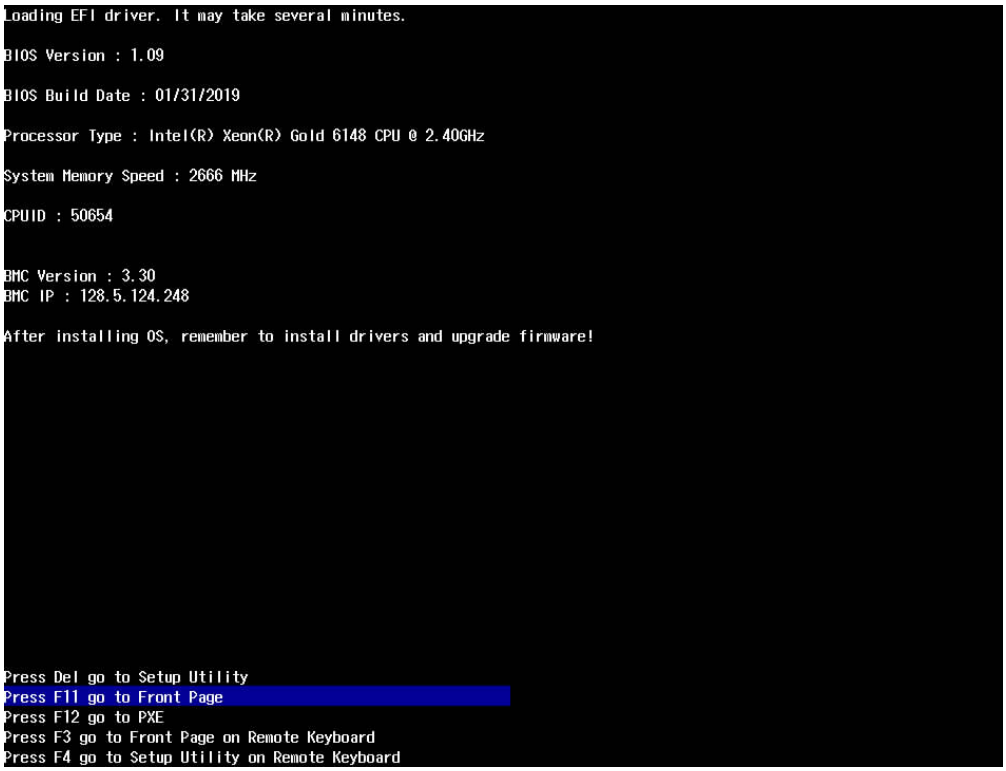
For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Restart the server.

Step 3 The BIOS configuration screen is displayed.

- If the BIOS version is V363 or earlier, the message shown in **Figure 3-1** is displayed.

Figure 3-1 Startup information (1)

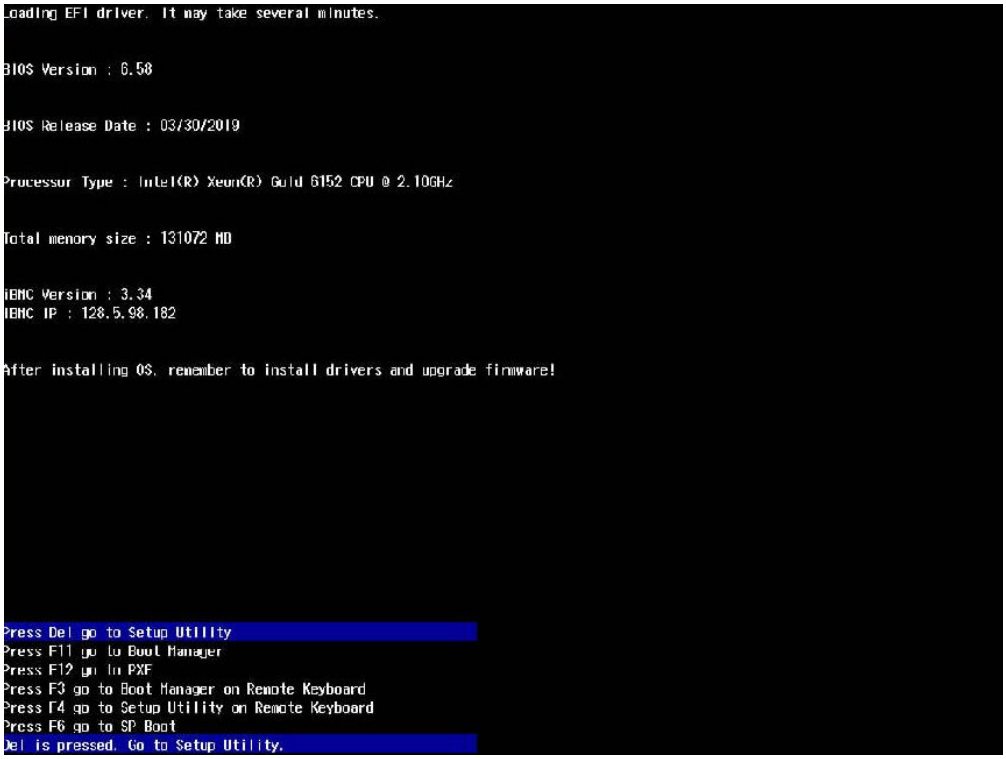


Press **F11** to access the BIOS screen, as shown in **Figure 3-2**. Go to **Step 5**.

Figure 3-2 BIOS screen (1)

- If the BIOS version is V658 or later, the message shown in **Figure 3-3** is displayed.

Figure 3-3 Startup information (2)



Press **Delete** to access the BIOS screen, as shown in **Figure 3-4**.

Figure 3-4 BIOS screen (2)

Step 4 Use arrow keys to select **BIOS Configuration** in the lower right corner and press **Enter**. The **Setup Utility** configuration screen is displayed. Go to **Step 6**.

Step 5 Use arrow keys to select **Setup Utility** in the lower right corner and press **Enter**. The **Setup Utility** configuration screen is displayed.

Step 6 In the navigation area on the left, choose **Advanced**, as shown in **Figure 3-5**.

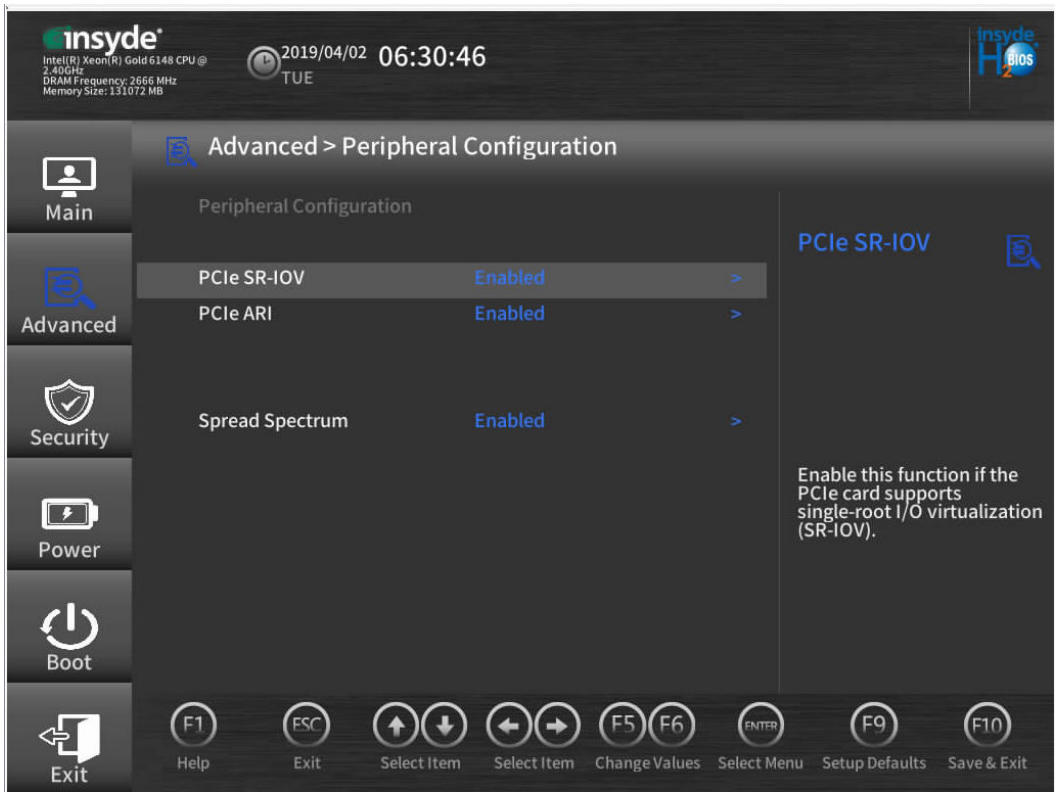
Figure 3-5 Advanced screen

Step 7 Use arrow keys to select **Peripheral Configuration** and press **Enter**. The **Peripheral Configuration** configuration screen is displayed, as shown in **Figure 3-6**.

Set **PCIe SR-IOV** to **Enbled**.

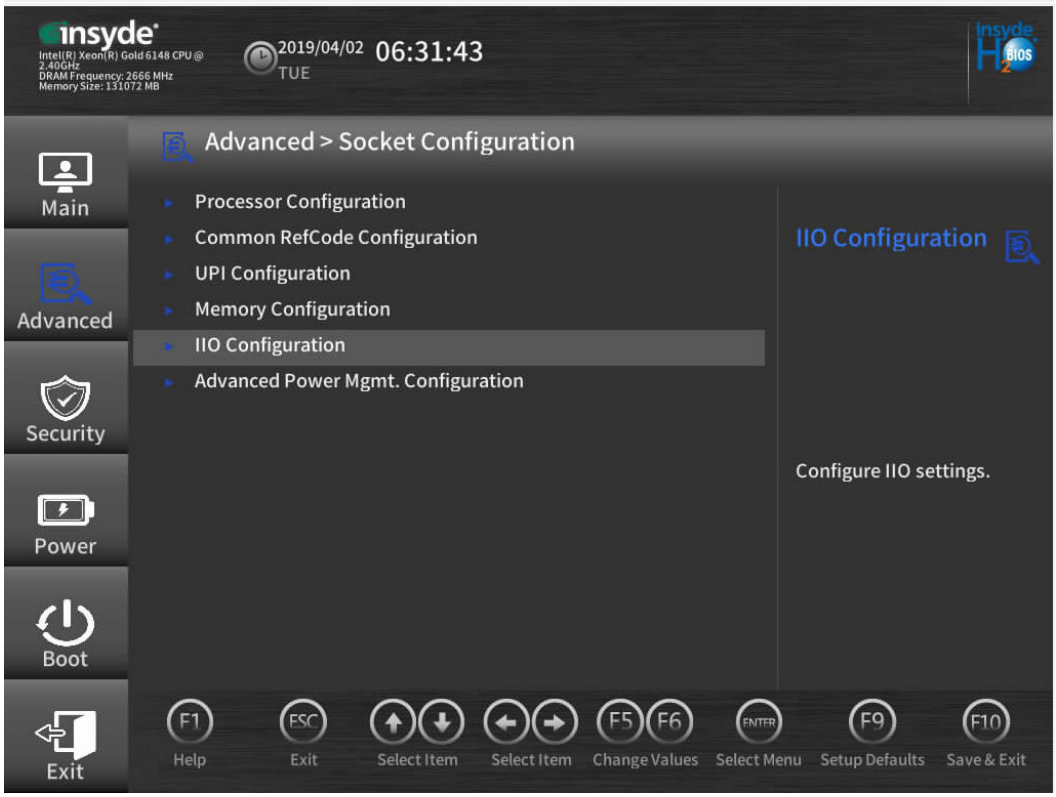
Press **Esc** to return to the **Advanced** screen.

Figure 3-6 Peripheral Configuration screen



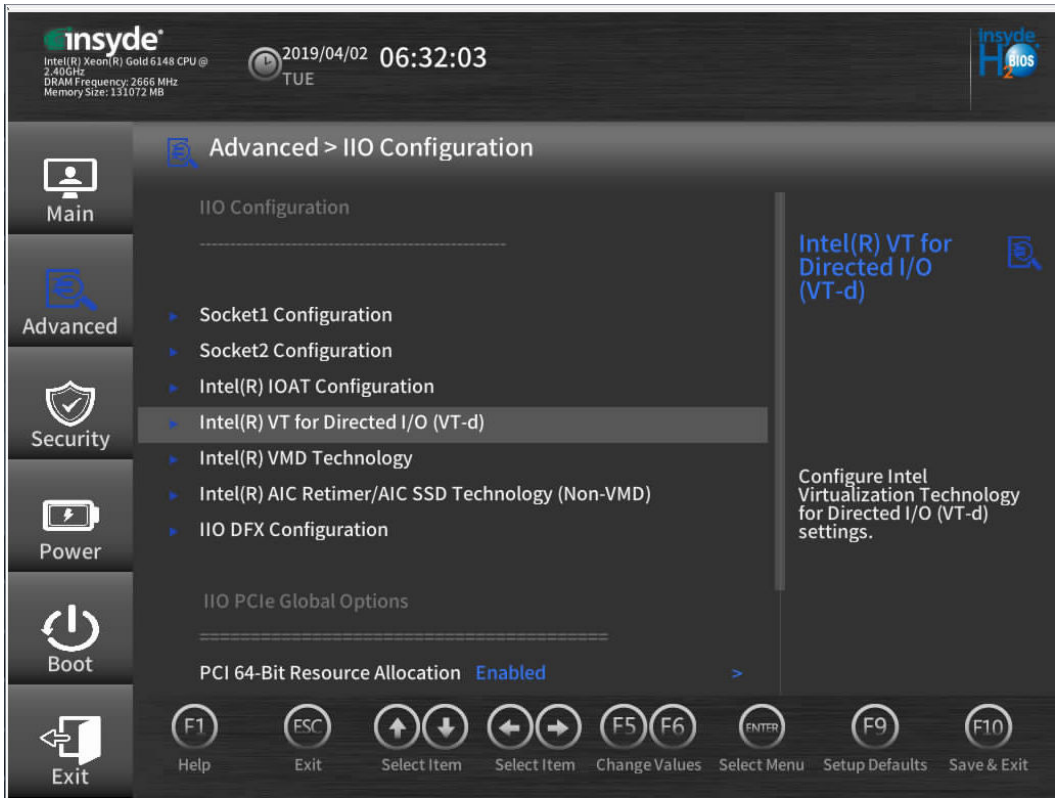
Step 8 Use the arrow keys to select **Socket Configuration** and press **Enter** to access the **Socket Configuration** screen, as shown in **Figure 3-7**.

Figure 3-7 Socket Configuration screen



Step 9 Use arrow keys to select **IIO Configuration** and press **Enter** to access the **IIO Configuration** screen, as shown in **Figure 3-8**.

Figure 3-8 IIO Configuration screen



Step 10 Use arrow keys to select **Intel(R) TV for Directed I/O (TV-d)** and press **Enter** to access the **Intel(R) TV for Directed I/O (TV-d)** screen, as shown in **Figure 3-9**.

Set **Intel(R) TV for Directed I/O (TV-d)** to **Enabled**. Press **Esc** to exit the screen.

Figure 3-9 Intel(R) TV for Directed I/O (TV-d) screen

----End

3.1.2 Modifying the GRUB Configuration File

Depending on the Linux distributions, the system Grand Unified Boot Loader (GRUB) configuration file may be **grub.conf** or **grub.cfg**.

This section describes how to configure the **iommu** and **intel_iommu** parameters in the SR-IOV pass-through mode.

- To enable SR-IOV in the kernel, add **intel_iommu=on** to the GRUB file.
- To avoid memory mapping and performance problems on the host, add **iommu=pt** to the GRUB file when SR-IOV is enabled.

Procedure

This section uses RHEL 7.4 as an example to describe how to modify the system GRUB configuration file.

Step 1 Run the following command to check whether SR-IOV is enabled:

```
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG=en_US
.UTF-8
```

- If yes, no further action is required.
- If no, go to **Step 2**.

Step 2 In the system, run the **find** command to search for the GRUB file.

```
[root@localhost ~]# find /boot/ -name grub.cfg
/boot/efi/EFI/redhat/grub.cfg
```

Step 3 Edit the **grub.cfg** file

vi /boot/efi/EFI/redhat/grub.cfg

Press **I** to edit the file and add the **intel_iommu=on iommu=pt** system startup command at the end of the file.

```
### BEGIN /etc/grub.d/10_linux ###
menuentry 'Red Hat Enterprise Linux Server (3.10.0-693.el7.x86_64) 7.4 (Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted $m
enuentry_id_option 'gnulinux-3.10.0-693.el7.x86_64-advanced-299e0ab2-22c9-42d4-9ff2-dd898bf1d541' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_gpt
        insmod xfs
        set root='hd0,gpt2'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2  91720993-f811-4d2e-8bcc-5dc0ff
e8e953
        else
          search --no-floppy --fs-uuid --set=root 91720993-f811-4d2e-8bcc-5dc0ffe8e953
        fi
        linuxefi /vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG
=en_US.UTF-8 intel_iommu=on iommu=pt
        initrdefi /initramfs-3.10.0-693.el7.x86_64.img
}
menuentry 'Red Hat Enterprise Linux Server (0-rescue-0b0eb4fa67204a6db3a36e22f4084149) 7.4 (Maipo)' --class red --class gnu-linux --class gnu --class
os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-0b0eb4fa67204a6db3a36e22f4084149-advanced-299e0ab2-22c9-42d4-9ff2-dd898bf1d541' {
```

After the modification is complete, press Esc to exit the editing mode and enter :wq! to save the settings and exit.

Step 4 Restart the system.

reboot

Step 5 Check whether **intel_iommu=on iommu=pt** is added to **/proc/cmdline**.

```
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG=en_US.UTF
intel_iommu=on iommu=pt
```

- If yes, no further action is required.
- If no, repeat **Step 2** to **Step 5**.

----End

3.2 ARM Version

3.2.1 Configuring the Server BIOSs

The following uses the BIOS of the TaiShan 2280 V2 as an example to describe how to enable the SR-IOV function in the BIOS.

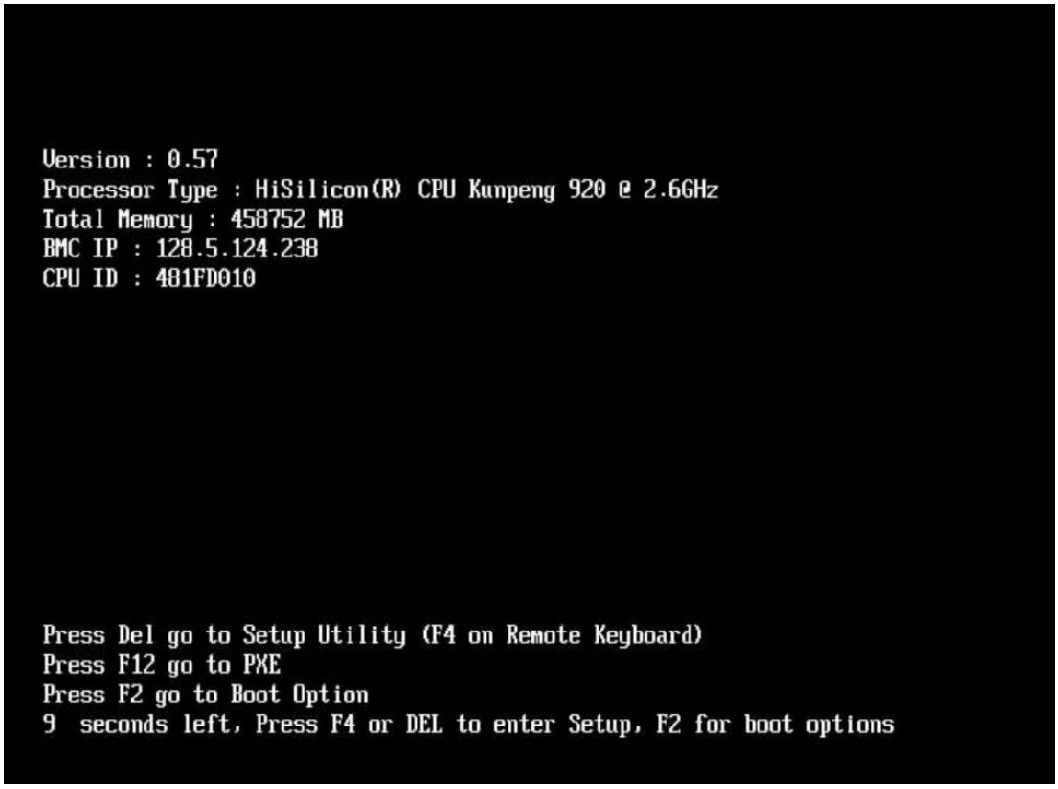
Step 1 Log in to the real-time server desktop using the Remote Virtual Console.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Restart the server.

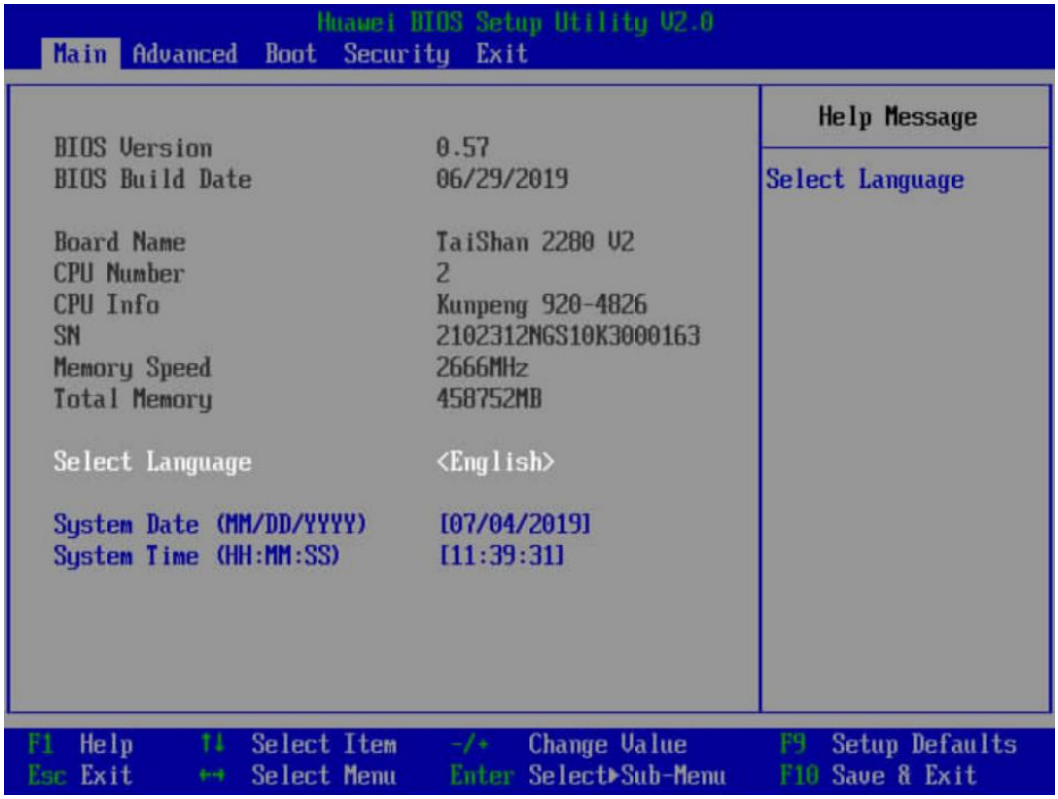
Step 3 Press **Delete** when the information shown in **Figure 3-10** is displayed.

Figure 3-10 Startup information



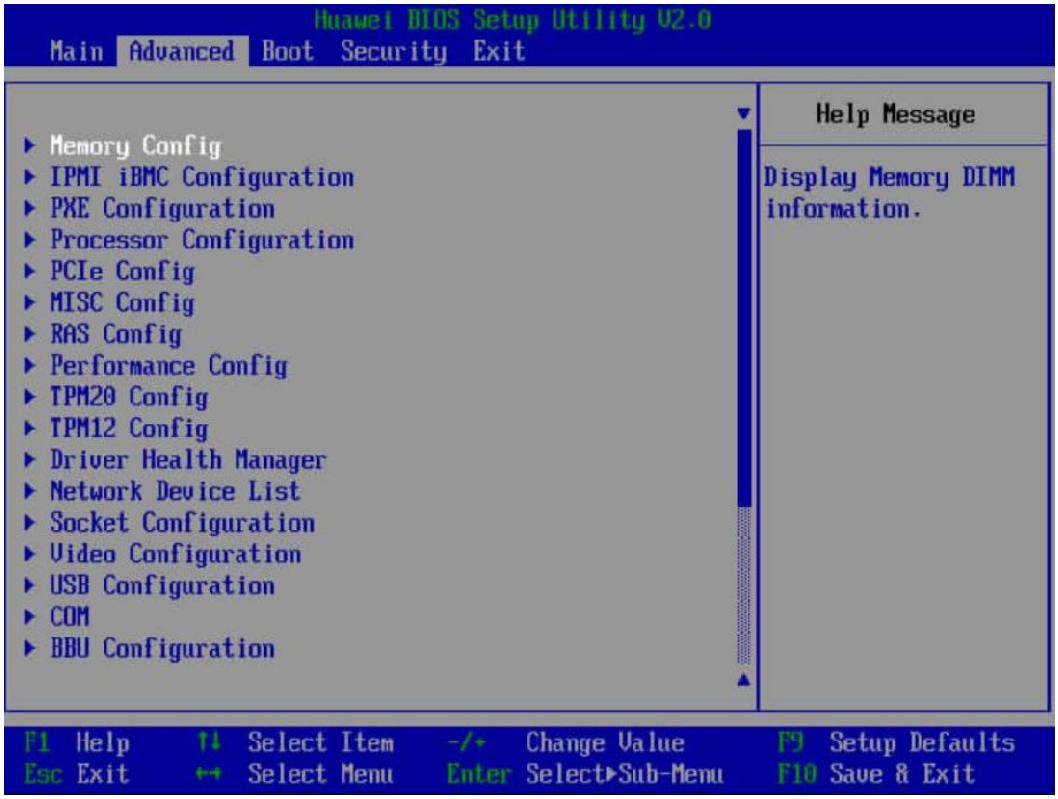
The **Setup Utility** screen is displayed, as shown in **Figure 3-11**.

Figure 3-11 Setup Utility screen



Step 4 Press ← or → to select **Advanced**. The **Advanced** screen is displayed, as shown in **Figure 3-12**.

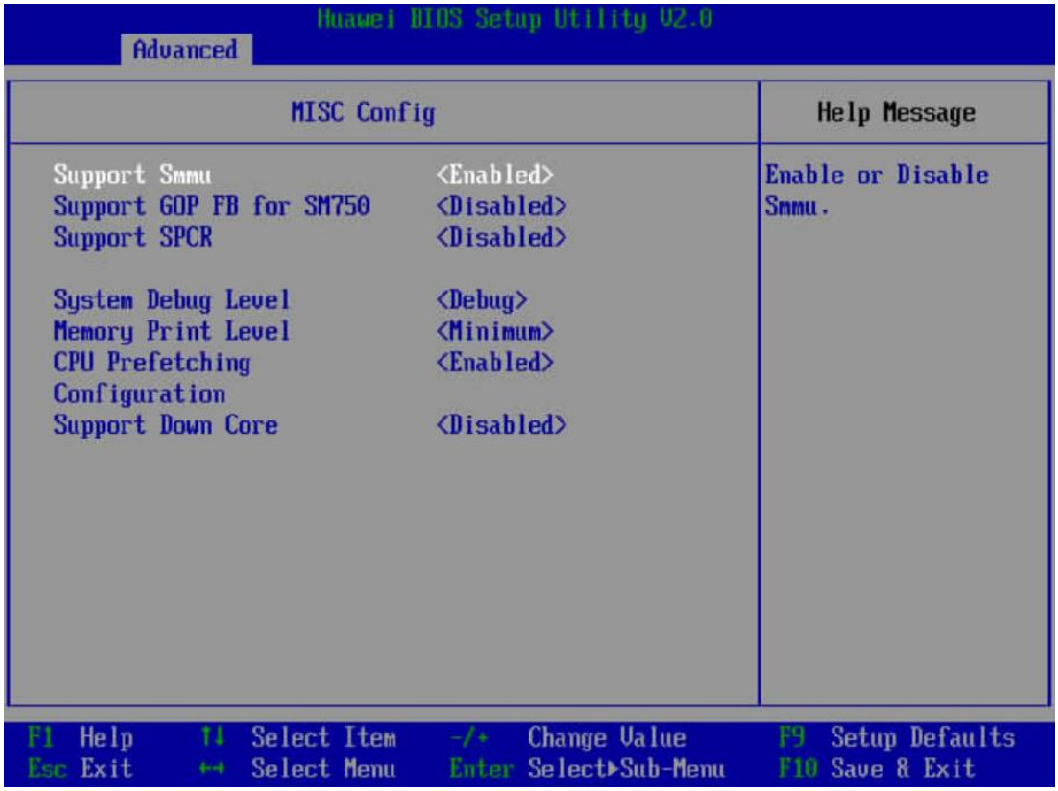
Figure 3-12 Advanced screen



Step 5 Press ↓ or ↑ to select **MISI Config** and press **Enter**. The **MISI Config** screen is displayed.

Step 6 Set **Support Smmu** to **Enabled** and keep default values of other options, as shown in **Figure 3-13**. Press **F10** to exit BIOS configuration.

Figure 3-13 Advanced screen



----End

3.2.2 Modifying the GRUB Configuration File

Depending on the Linux distributions, the system Grand Unified Boot Loader (GRUB) configuration file may be **grub.conf** or **grub.cfg**.

To avoid memory mapping and performance problems on the host, add **iommu.passthrough=1** to the GRUB file when SR-IOV is enabled.

This section describes how to configure the **iommu.passthrough=1** parameters in the SR-IOV pass-through mode.

Procedure

This section uses RHEL 7.6 as an example to describe how to modify the system GRUB configuration file.

Step 1 Run the following command to check whether SR-IOV is enabled:

```
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-4.14.0-115.el7a.aarch64 root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG
=zh_CN.UTF-8 selinux=0 pci=realloc highres=on
```

- If yes, no further action is required.
- If no, go to **Step 2**.

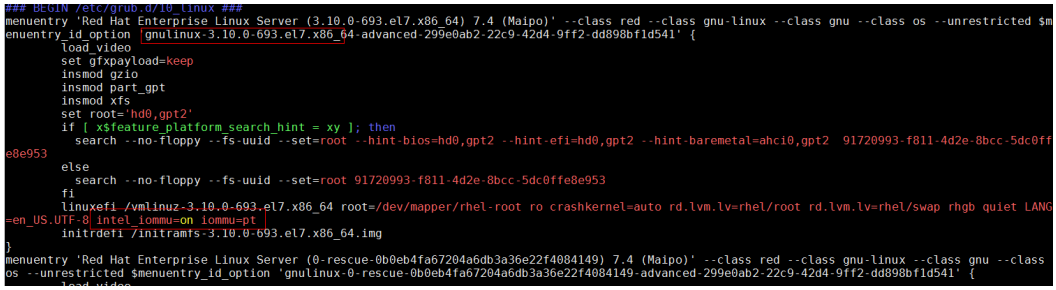
Step 2 In the system, run the **find** command to search for the GRUB file.

```
[root@localhost ~]# find /boot/ -name grub.cfg
/boot/efi/EFI/redhat/grub.cfg
```


Step 3 Edit the **grub.cfg** file

vi /boot/efi/EFI/redhat/grub.cfg

Press **I** to edit the file and add the **iommu.passthrough=1** system startup command at the end of the file.



After the modification is complete, press Esc to exit the editing mode and enter :wq! to save the settings and exit.

Step 4 Restart the system.

reboot

Step 5 Check whether **intel_iommu=on iommu=pt** is added to **/proc/cmdline**.

- If yes, no further action is required.
- If no, repeat **Step 2** to **Step 5**.

----End

4 Configuring QoS

4.1 Overview of Flow Control

IN200 flow control is enabled by default. You can run the **ethtool -a DEV_NAME** command to query the status of flow control. Enabling PFC disables flow control.

The QoS can be configured only when the RoCE driver is used.

4.2 Configuring Flow Control

Prerequisites

- The RoCE driver package of the IN200 NIC has been installed.
- The VLAN has been configured.

Procedure

To configure QoS, perform the following procedure:

1. Configure the Data Center Bridging (DCB). Use the hinicadm management tool. For command details, see **6.1.4.45 Querying and Setting the DCB Function (dcb)**.
2. Configure PFC. Use the hinicadm management tool. For command details, see **6.1.4.46 Querying and Setting the PFC Function (pfc)**.
3. Configure ETS. Use the hinicadm management tool. For command details, see **6.1.4.47 Setting the ETS Function (ets)**.

5 Configuring RoCE Bonding

You can implement RoCE bonding by switching physical ports based on the bonding function provided by the OS and using the RoCE driver. Multiple physical ports can be bound as a logical port to implement fault tolerance, bandwidth expansion, and load balancing.

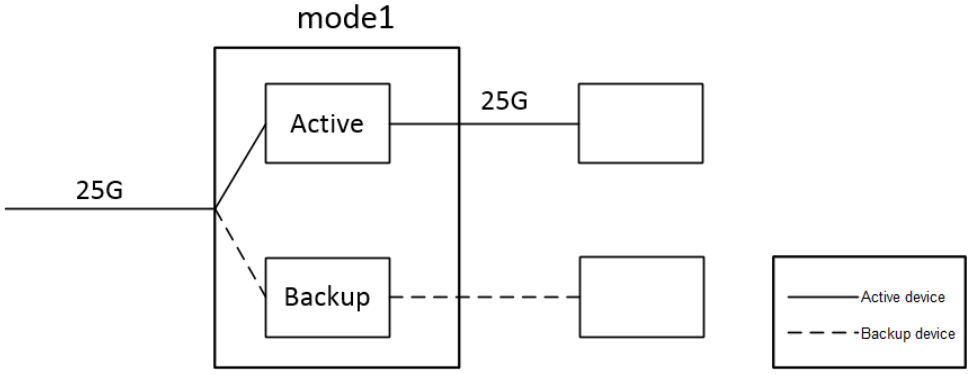
5.1 Overview

The RoCE bonding supports three modes: Active-backup mode1, Balance-XOR mode2, and 802.3ad mode4.

- Active-backup mode1: Only one device is active. If the active device fails, the other device becomes active. The MAC address is visible externally. The MAC address of the bonding in this mode is unique, preventing switch access disorders. This mode provides only the fault tolerance capability and high availability of network connections, but the resource utilization is low.

  Figure 5-1 shows Active-backup mode1.

  Figure 5-1 Active-backup mode1

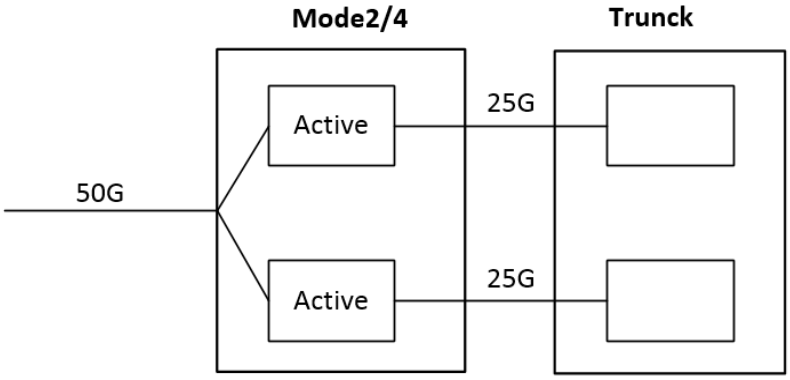
  

- Balance-XOR mode2: Data is transmitted based on the selected hash policy. This mode provides load balancing and fault tolerance capabilities.

- 802.3ad mode4: IEEE 802.3ad dynamic link aggregation mode. You can create an aggregation group that shares the same rate and duplex settings. Multiple devices can work in the same aggregation group that has been activated based on 802.3ad.

  Figure 5-2 shows Balance-XOR mode2 and 802.3ad mode4.

  Figure 5-2 Balance-XOR mode2/802.3ad mode4



5.2 Preparing OSs Supported by the RoCE Bonding

Table 5-1 OSs supported by Huawei in-house OFED

OS	x86	ARM
CentOS	7.4, 7.5, and 7.6	7.5 and 7.6
RHEL	7.5 and 7.6	-
UVP	2.5.RC10.SPC110B055d and 2.5.RC9.B057	3.0.RC1.SPC800B050, 3.0.RC2.B033, and 2.5.RC8.SPC800.B010
Ubuntu	18.04.1	18.04.2
EulerOS	V200R007C00SPC500B005 and V200R005C00SPC310B056	2.2.RC3, V200R008C00SPC100B220, V200R008C00B180, and V200R005C00SPC310B056

5.3 Configuring RoCE Bonding

You can implement bonding by creating, destroying, and configuring bonding devices using commands or a configuration file. NetworkManager is not recommended for bonding configuration.

Prerequisites

The RoCE driver package of the IN200 NIC has been installed.

The VLAN has been configured.

5.3.1 Using Commands

This section uses CentOS 7.4 as an example to describe how to configure bonding using commands.

NOTICE

This method takes effect temporarily. After the network service is reset in the OS or the OS is restarted, the configuration is lost.

Step 1 Log in to the server OS as the **root** user, right-click the OS desktop, and choose **Open Terminal** from the shortcut menu.

Step 2 Run the following command to find the port in the **up** state:

ibdev2netdev

Step 3 Run the following commands in sequence to create **bond0** (ports **enp6s0** and **enp7s0** are examples):

modprobe -first-time bonding

echo *X* **> /sys/class/net/bond0/bonding/mode** #The **X** value can be **1**, **2**, or **4**.

NOTE

- In the preceding command, *X* specifies the bonding mode.
- If **mode** is set to **1**, the switch is configured properly. If **mode** is set to **2**, the switch must be configured with a trunk. If **mode** is set to **4**, the trunk of the switch must be configured to the static LACP mode. It is recommended that the rate be set to the same value as that of bonding.
- If **mode** is set to **4**, you need to configure **lacp_rate**.

  echo 1 > /sys/class/net/bond0/bonding/lacp_rate

ifconfig bond0 up

ifenslave bond0 enp6s0 enp7s0

echo 100 > /sys/class/net/bond0/bonding/miimon #miimon: interval for checking the network, in milliseconds

Step 4 Run the following command to create an IP address of **bond0**:

ifconfig bond0 *IP* #*IP* specifies the IP address of **bond0**.

----End

5.3.2 Using a Configuration File

This section uses CentOS 7.4 as an example to describe how to configure bonding using a configuration file.

Step 1 Log in to the server OS as the **root** user, right-click the OS desktop, and choose **Open Terminal** from the shortcut menu.

Step 2 Run the following command to find the port in the **up** state:

ibdev2netdev

Step 3 Run the following command to access the **/etc/sysconfig/network-scripts** directory:

cd /etc/sysconfig/network-scripts

Step 4 Run the following command to create and open the **ifcfg-fsb_bond** configuration file:

vi ifcfg-fsb_bond

Press **i** to add the following information to the file:

```
DEVICE=fsb_bond
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
TYPE=Bonding
MTU=1500
BONDING_OPTS='mode=1 miimon=100 updelay=0 downdelay=0 num_grat_arp=50'
NM_CONTROLLED=no
IPADDR=
NETMASK=
NETWORK=
```

NOTE

- Set the bonding mode and network check time in the **BONDING_OPTS** configuration item.
- If **mode** is set to **1**, the switch is configured properly. If **mode** is set to **2**, the switch must be configured with a trunk. If **mode** is set to **4**, the trunk of the switch must be configured to the static LACP mode. It is recommended that the rate be set to the same value as that of bonding.
- If the bonding mode is set to **4** in the **BONDING_OPTS** item, you are advised to set lacp_rate.

After the modification is complete, press **Esc** to exit editing mode and enter **:wq!** to save the modification and exit.

Step 5 This step uses ports **enp8s0** and **enp9s0** as an example to describe how to bind fsb_bond. In the **/etc/sysconfig/network-scripts** directory, modify **ifcfg-enp8s0** and **ifcfg-enp9s0** and add the **MASTER=fsb_bond** option.

For example, to edit the **ifcfg-enp8s0** file, run the following command:

vi ifcfg-enp8s0

Press **i** to add the following information to the file:

```
DEVICE=enp8s0
BOOTPROTO=none
ONBOOT=yes
MASTER=fsb_bond
SLAVE=yes
USERCTL=no
NM_CONTROLLED=no
```

After the modification is complete, press **Esc** to exit editing mode and enter **:wq!** to save the modification and exit.

Step 6 Run the following command to restart the network:

service network restart

----End

6 Management Tools

6.1 Customized Management Tool - hinicadm

The hinicadm is a management tool customized for the IN200. You can use this tool to manage the IN200.

6.1.1 Installing hinicadm

Prerequisites

- The hinicadm installation package has been downloaded.
  For the package name and download method, see **2.1 Obtaining Software Packages**.
- The installation package has been uploaded to the server OS.

Installing hinicadm on Linux

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Install the hinicadm tool.

- RPM package installation
  Run the **rpm -ivh hinicadm-xxxx-xxxx.xxx.rpm** command.
  Example:

```
linux-uueJIR:/Hi1822/tools/linux # rpm -ivh hinicadm-1.2.8.0-1.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:hinicadm-1.2.8.0-1               ################################# [100%]
```

- DEB package installation
  Run the **dpkg -i hinicadm-xxxx-xxxx.xxx.deb** command.
  Example:

```
root@ubuntu1804:/home/ubuntu # dpkg -i
hinicadm-1.6.1.1-4.15.0_20_generic.arm64.deb
```



```
Selecting previously unselected package hinic.
(Reading database ... 72646 files and directories currently installed.)
Preparing to unpack hinicadm-1.6.1.1-4.15.0_20_generic.arm64.deb ...
Unpacking hinic (1.6.1.1) ...
Setting up hinic (1.6.1.1) ...
Installing... Please wait for a moment.
Install hinicadm tool successfully.
```

----End

Installing hinicadm on Microsoft Windows

Step 1 Log in to Microsoft Windows.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the directory where the msi package is stored, for example, **C:**.

Step 3 Double-click **hinicadm_<version>x86_64.msi** to install the hinicadm tool. During the installation, click  on the task bar and click **Install** in the displayed window to install the 7z-Zip tool. You can use 7z-Zip to compress and parse logs. The default installation path is **C:\Program Files (x86)\hinicadm**.

NOTE

The default installation path includes the hinicadm tool, log collection script **collect_log.exe**, and log packing tool 7z.

----End

6.1.2 Using hinicadm

Command Format

The hinicadm supports the following command format:

hinicadm <***major cmd***> <***minor cmd***> <*parameter*> …<***minor cmd***> <*parameter*>

The command function is determined by the major command (***major cmd***) and minor commands (***minor cmd***).

NOTE

On Microsoft Windows, you cannot double-click **hinicadm.exe** to use to the tool. The following describes how to use the commands on Microsoft Windows:

1. Run **cmd** to open the command line interface (CLI), right-click the title bar and choose **Properties** from the menu, click the **Layout** tab, and set **Width** under **Screen Buffer Size** to **999** to ensure that the command output format is correct.
2. Go to the tool installation directory, for example, **C:\Program Files (x86)\hinicadm**.
3. Enter **hinicadm.exe<major cmd> <minor cmd> <*parameter*> ···<minor cmd> <*parameter*>**.

Obtaining Help Information

When using the hinicadm tool, you can use the following commands to obtain help information:

- Obtaining the tool version information

  Short command: **hinicadm -v**

Long command: **hinicadm --v**

- Obtaining the list of major commands supported by the tool

  Short command: **hinicadm -h**

  Long command: **hinicadm --help**

- Obtaining the list of minor commands supported by the tool

  Short command: **hinicadm <major cmd> -h**

  Long command: **hinicadm <major cmd> --help**

  In addition, you can add **-h** to any major command to obtain the value ranges of all minor commands and the parameters in the commands.

6.1.3 Command List

Table 6-1 lists the major commands and their functions supported by hinicadm.

Table 6-1 List of major commands

Major Command	Function	Remarks
version	Queries version information.	-
info	Queries the system device list or the basic information about a specified device.	-
log	Obtains online logs and parses offline logs.	-
updatefw	Upgrades and activates the firmware.	-
counter	Collects statistics.	-
table	Prints linear table information.	-
mac	Queries the fixed MAC address.	-
fec	Queries and sets the Forward Error Correction (FEC) mode of a specified port.	-
rate	Queries and sets the port rate limit.	-
temperature	Queries the chip temperature and optical module temperature.	-
event	Queries asynchronous event statistics.	-
clear	Clears statistics.	-
qos	Queries and Sets the User Priority and CGE Pause Time	For Linux only
reg	Queries the device register information.	-

Major Command	Function	Remarks
autoneg	Queries and sets the auto negotiation mode.	-
nic_queue	Queries the rx/tx queue information of a device.	-
mode	Queries and sets the device work mode.	-
hilink_param	Queries the **ctle**, **ffe**, and **dfe** parameters.	-
fe_epc	Obtains Fusion Engine (FE) error information.	-
hilink_dump	Queries information about the hilink register.	-
hilink_port	Queries port information.	-
hilink_count	Queries hilink statistics information.	-
hilink_speed	Queries and sets the rate of a port.	-
bp	Queries chip back pressure information.	The ARM versions of the following OSs are not supported: ● SLES 15/12.3/12.4 ● RHEL 7.4/7.5/7.6 ● CentOS 7.4/7.5/7.6 ● EulerOS V2.0 SP8 ARM ● NeoKylin Server V5.0U5/V7.0U5/ V7.0U6 **NOTE** ● These commands are used only by developers for troubleshooting and do not affect your use of the hinicadm tool. ● These commands can be used only after the iommu (for an x86 version) or Smmu (for an ARM version) parameter is enabled in the SR-IOV pass-through mode. For details, see **3 Configuring SR-IOV**.
dp	Queries statistics on chip data path modules.	
cpb	Queries the usage of chip CPB cells.	
tile_io	Queries the status of microcode threads.	
csr_rd	Reads the chip register.	
csr_wr	Writes the chip register.	
csr_dump	Reads and saves chip register values in batches.	
pdm	Queries the CPB CELL resource allocation and current resource usage.	

Major Command	Function	Remarks
fm_show	Queries the interrupt information reported by a hardware module	-
sriov	Queries and sets the PF virtualization control status.	-
self_adaption	Queries and sets the port auto-adaptation mode.	-
serdes	Queries port serdes information.	-
sfp	Queries optical module information.	-
reset	Restores factory settings.	-
lro	Queries and Sets the LRO Coalesce Time	-
xstats	Queries the statistics of received and sent packets of a NIC port	for Windows only
inter_coal	Queries and sets the interrupt coalescence parameters	
sdi_mode	Queries and Sets the SDI Card Mode	for Linux only
sdi_cfg	Queries and Sets the SDI Configuration	
roce_port_traffic	Queries the real-time traffic sent and received by each port of a NIC where a RoCE device is located.	-
reboot_notice	Sets NIC Packet Data to Be Sent and Triggering NIC Packet Sending When AC Power Is Lost	for ARM versions of Linux only
dcb	Queries and sets the DCB function.	Currently, QoS can be configured only in Linux.
pfc	Queries and sets the PFC function.	
ets	Queries and sets the ETS function.	

6.1.4 Command Reference

This section describes the meanings, parameters, and usage of the common commands of the hinicadm tool. For more information about the commands, you can run the **-h** command on the hinicadm.

NOTE

On Microsoft Windows, you cannot double-click **hinicadm.exe** to use to the tool. The following describes how to use the commands on Microsoft Windows:

1. Run **cmd** to open the command line interface (CLI), right-click the title bar and choose **Properties** from the menu, click the **Layout** tab, and set **Width** under **Screen Buffer Size** to **999** to ensure that the command output format is correct.
2. Go to the tool installation directory, for example, **C:\Program Files (x86)\hinicadm**.
3. Enter **hinicadm.exe<major cmd> <minor cmd> <*parameter*> ···<minor cmd> <*parameter*>**.

6.1.4.1 Querying the Version Information of a Device (version)

Function

The **version** command is used to query the version information about the IN200 driver, firmware, and tools.

Format

hinicadm version -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the version information of a specified IN200.

```
[root@localhost tool]# hinicadm version -i hinic0
boot: 1.2.0.0 2018-01-09_10:47:41
up: 1.2.0.0 2018-01-09_10:47:41
ucode: 1.2.0.0 2018-01-09_10:47:41
hinicadm tool: 1.2.0.0
hinic driver: 1.2.0.0
```

6.1.4.2 Querying Basic Information (info)

Function

The **info** command is used to query basic information about all IN200s or a specified IN200 on a server.

Format

hinicadm info

hinicadm info -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query information about all the IN200 cards in the server.

```
[root@localhost tool]# hinicadm info
Hi1822 device Information
      Card          PCIe Function
|---- hinic0(ETH)
|------------ 08:00.0 (NIC:eth5)
|------------ 08:00.1 (NIC:eth6)
|------------ 08:00.2 (NIC:eth7)
|------------ 08:00.3 (NIC:eth8)
|---- hinic1(ETH)
|------------ 09:00.0 (NIC:eth9)
|------------ 09:00.1 (NIC:eth10)
|------------ 09:00.2 (NIC:eth11)
|------------ 09:00.3 (NIC:eth12)
```

Query the basic information of a specified IN200.

```
[root@localhost tool]# hinicadm info -i hinic0
       Card information:
       card type:      ETH
       port num:       4
       port speed:     25GE
       pcie width:     16
       host num:       1
       pf num:         4
       vf total num:   240
       tile num:       2
       qcm num:        6
       core num:       4
       work mode:      0
       service mode:   2
       pcie mode:      X16_MODE
       cfg addr:       0x20000
       boot sel:       0
```

6.1.4.3 Collecting Logs (log)

6.1.4.3.1 Collecting All Firmware Logs of a Specified Device (-a)

Function

The **log -a** command is used to collect all firmware logs of a specified device, including the firmware and microcode logs of the RAM and flash memory as well as the last words of the firmware and microcode.

The collected logs are automatically saved to **/opt/hinic/fwlog/** in the tool installation directory.

Format

hinicadm log -i <*devicename*> **-a**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query all logs on a specified IN200.

```
[root@localhost tool]# hinicadm log -i hinic0 -a
/opt/hinic/fwlog/ucode_flash_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/ucode_ram_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/ucode_lastword_flash_hinic0_2018_01_15_17_05_41.log create
succeed.
/opt/hinic/fwlog/up_lastword_flash_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/up_flash_hinic0_2018_01_15_17_05_41.log create succeed.
/opt/hinic/fwlog/up_ram_hinic0_2018_01_15_17_05_40.log create succeed.
[root@localhost tool]# /opt/hinic/fwlog # ls
ucode_flash_hinic0_2018_01_15_17_05_41.log
ucode_ram_hinic0_2018_01_15_17_05_41.log
up_lastword_flash_hinic0_2018_01_15_17_05_41.log
ucode_lastword_flash_hinic0_2018_01_15_17_05_41.log
up_flash_hinic0_2018_01_15_17_05_41.log
up_ram_hinic0_2018_01_15_17_05_40.log
```

6.1.4.3.2 Collecting Logs of a Specified Type (-t)

Function

The **log -t** command is used to collect logs of a specified type and save the collected logs to **/opt/hinic/fwlog**.

Format

hinicadm log -i <*devicename*> **-t** <*logtype*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**
logtype	Indicates the type of logs to be collected.	● **0**: indicates the firmware logs in the RAM. ● **1**: indicates the firmware logs in the flash memory. ● **2**: indicates the microcode logs in the RAM. ● **3**: indicates the microcode logs in the flash memory. ● **4**: indicates the last words of the firmware and microcode.

Usage Instruction

None

Example

Query the firmware logs in the flash memory on a specified IN200.

```
[root@localhost tool]# hinicadm log -i hinic0 -t 1
/opt/hinic/fwlog/ up_flash_hinic0_2018_01_15_17_05_41.log create succeed.
```

6.1.4.3.3 Parsing Run Logs Offline (-o1)

Function

The **log -o1** command is used to parse firmware and microcode run logs offline and save the parsing results to **/opt/hinic/fwlog** in the tool installation directory.

Format

hinicadm log -o1 <*logfile*> **-m** <*mgmt_index*> **-u** <*ucode_index*>

Parameters

Parameter	Description	Value
logfile	Indicates the name of the log file to be parsed. **NOTE** This file is the out-of-band run log file of the IN200 obtained using the one-click information collection function of the iBMC. For details about the file obtaining method and save path, see the iBMC User Guide.	Example: **err_log.bin**
mgmt_index	Indicates the firmware index file corresponding to the log file to be parsed. **NOTE** The archive address and download method of this file are the same as those of the IN200 driver and firmware. For details, see **2.1 Obtaining Software Packages**.	Example: **up_index**
ucode_index	Indicates the microcode index file corresponding to the log file to be parsed. **NOTE** The archive address and download method of this file are the same as those of the IN200 driver and firmware. For details, see **2.1 Obtaining Software Packages**.	Example: **ucode_index**

Usage Instruction

Before running this command, transfer the log files and index files to be parsed to the installation directory of the tool.

When running this command, you can specify either **-m** or **-u** or both.

Example

Parse the logs of the IN200 offline.

```
[root@localhost tool]# hinicadm log -o1 error_log.bin -m up_index -u ucode_index
/opt/hinic/fwlog/ up_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ up_flash_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_flash_offline_2018_01_15_18_52_32.log create succeed.
[root@localhost tool]# /opt/hinic/fwlog # ls
/opt/hinic/fwlog/ up_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ up_flash_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_ram_offline_2018_01_15_18_52_32.log create succeed.
/opt/hinic/fwlog/ ucode_flash_offline_2018_01_15_18_52_32.log create succeed.
```

6.1.4.3.4 Parsing Last Words Offline (-o2)

Function

The **log -o2** command is used to parse firmware and microcode last words offline and save the parsing results to **/opt/hinic/fwlog** in the tool installation directory.

Format

hinicadm log -o2 <*logfile*>

Parameters

Parameter	Description	Value
logfile	Indicates the name of the last-word file to be parsed. **NOTE** This file is the out-of-band last-word file of the IN200 obtained using the one-click information collection function of the iBMC. For details about the file obtaining method and save path, see the iBMC User Guide.	Example: **last_word.bin**

Usage Instruction

Before running this command, transfer the last-word files and index files to be parsed to the installation directory of the tool.

Example

Parse the last words of the IN200 offline.

```
[root@localhost tool]# hinicadm log -o2 last_word.bin
/opt/hinic/fwlog/ ucode_lastword_flash_offline_2018_01_15_18_58_25.log create
succeed.
/opt/hinic/fwlog/ up_lastword_flash_offline_2018_01_15_18_58_25.log create
succeed.
[root@localhost tool]# /opt/hinic/fwlog # ls
ucode_lastword_flash_offline_2018_01_15_18_58_25.log
up_lastword_flash_offline_2018_01_15_18_58_25.log
```

6.1.4.3.5 Exporting Windows System Event Logs (-e)

Function

The **log -e** command is used to export windows system event logs.

Format

hinicadm.exe log -e

Parameters

None

Usage Instruction

The command is used only for the Windows.

Example

Export Windows system event logs.

```
C:\Users\Administrator\Desktop> .\hinicadm.exe log -e
C:\Program Files (x86)\hinic\fwlog\hinicdriver_20190329_185903.evtx create
succeed.
```

6.1.4.4 Upgrading Firmware (updatefw)

6.1.4.4.1 Cold-Upgrading All Firmware (-f)

Function

The **updatefw -f** command is used to cold-upgrade all firmware. You need to restart the OS for the upgrade to take effect.

Format

hinicadm updatefw -i <*devicename*> **-f** <*fwfile*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be upgraded in the system.	Example: **hinic0** and **hinic1**
fwfile	Indicates the path to the firmware upgrade package.	Format: *Path*/*Upgrade package name*

Usage Instruction

None

Example

Cold-upgrade the IN200 firmware.

```
[root@localhost tool]#hinicadm updatefw -i hinic0 -f Hi1822_nic_prd_1h_4x25G.bin
Please do not remove driver or network device.
Loading...
Firmware update start: 2018-07-14 22:22:48
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] [100%][\]
Firmware update finish: 2018-07-14 22:23:10
Firmware update time uesd: 22s
Loading firmware image succeed.
Please reboot OS to take firmware effect
```

6.1.4.4.2 Cold-Upgrading the Firmware and Configuration File (-f, -c)

Function

The **updatefw -f -c** command is used to cold-upgrade the firmware and configuration file.

NOTE

Currently, the IN200 firmware is upgraded as a whole. Therefore, do not use this command for the upgrade. To perform the upgrade, see **Cold-Upgrading All Firmware (-f)**.

Format

hinicadm updatefw -i <*devicename*> **-f** <*fwfile*> **-c** <*configfile*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be upgraded in the system.	Example: **hinic0** and **hinic1**
fwfile	Indicates the path to the firmware upgrade package.	Format: *Path*/*Upgrade package name*
configfile	Indicates the path to the configuration file upgrade package.	Format: *Path*/*Upgrade package name*

Usage Instruction

None

Example

Cold-upgrade the firmware and configuration file of the IN200.

```
[root@localhost tool]#hinicadm updatefw -i hinic0 -f /OSM/modules/port/
Hi1822_fw.bin -c /OSM/modules/Hi1822/update_cfg/Hi1822_cfg_nic_smart_4x25G.bin
Please do not remove driver or network device.
Loading...
Firmware update start: 2018-07-14 22:22:48
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] [100%][\]
Firmware update finish: 2018-07-14 22:23:10
Firmware update time uesd: 22s
Loading firmware image succeed.
Please reboot OS to take firmware effect.
```

6.1.4.4.3 Hot-Upgrading Firmware (-f, -a)

Function

The **updatefw -f -a** command is used to hot-upgrade firmware and specify the activation mode.

Format

hinicadm updatefw -i <*devicename*> **-f** <*fwfile*> **-a** <*activemode*> [**-t**<*activetype*>]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be upgraded in the system.	Example: **hinic0** and **hinic1**
fwfile	Indicates the path to the firmware upgrade package.	Format: *Path/Upgrade package name*
activemode	Indicates the activation mode.	● **now**: activates the firmware immediately. ● **later**: activates the firmware later. NOTE ● If the command does not contain **-a**, it implements **6.1.4.4.1 Cold-Upgrading All Firmware (-f)**. ● If the frimware is not activated immediately, you need to set **activemode** to **now** and run the command again to activate the firmware later.
activetype	Indicates the activation type.	● **up**: activates the uP firmware. ● **ucode**: activates the microcode firmware. NOTE If the command does not contain **-t**, all firmware is activated.

Usage Instruction

None

Example

Upgrade abd activate the IN200 firmware.

```
[root@localhost tool]#hinicadm updatefw -i hinic0 -f ./Hi1822_nic_std_4_25G.bin -
a now
Please do not remove driver or network device.
Loading...
Firmware update start: 2018-07-14 22:54:47
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] [100%][\]
Firmware update finish: 2018-07-14 22:55:07
Firmware update time uesd: 20s
Do not operate the device during the hot upgrade.
Firmware is activing. Please waiting…...
```

6.1.4.4.4 Activating Firmware (-a)

Function

The **updatefw -a** command is used to activate hot-upgraded firmware that has not been activated.

Format

hinicadm updatefw -i <*devicename*> **-a** now

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be upgraded in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

Before running this command, ensure that hot-upgraded firmware that has not been activated exists in the system.

Example

Activate firmware.

```
[root@localhost tool]#hinicadm updatefw -i hinic0 -a now
Do not operate the device during the hot upgrade.
Firmware is activing. Please waiting…
Active the Loading firmware succeed.
```

6.1.4.5 Querying Statistics (counter)

6.1.4.5.1 Querying the Statistics of a Specified Device (-t, -x)

Function

The **counter** command is used to query all statistics of a specified device, including the MIB, microcode, firmware, driver, and IPSU statistics.

The **counter -t** command is used to query the statistics of a specified device type.

The **counter -t -x** command is used to query the statistics of a specified sequence of a specified device type.

Format

hinicadm counter -i <*devicename*>

hinicadm counter -i <*devicename*> **-t** <*countertype*>

hinicadm counter -i <*devicename*> **-t** <*countertype*> **-x** <*counterindex*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1** **NOTE** ● If **countertype** is set to 3, the value can be **eth0** or **eth1**. ● If the value of this parameter is **hinic0**, this command queries the statistics of all PFs on the card. If the value of this parameter is a port name, this command queries the statistics of the specific port.
countertype	Indicates the statistics type.	● **0**: firmware statistics ● **1**: microcode statistics ● **2**: IPSURX statistics ● **3**: driver statistics
counterindex	Indicates the statistics sequence.	You can run the **counter -h** command to obtain the value range.

Usage Instruction

If the command contains **-x**, the value of **countertype** can be **0** and **1**.

Example

Query all statistics of the IN200.

```
[root@localhost tool]# hinicadm counter -i hinic0
uP Statistics:
uP Common Counter:
mag    module except: 117
general level except: 246
suggest level except: 127
mag:err_mag_rf_lf: 252
mag:err_mag_linkdown: 250
mag:err_mag_linkup: 127
……
```

Query the microcode statistics of the IN200.

```
[root@localhost tool]# hinicadm counter -i hinic0 -t 1
uCode Statistics:
NIC MIB TX UC Counter:
func1:tx_uc: 0x00000193b7694888 00000000197c70de
func76:tx_uc: 0x0000000002e9c940 00000000000b4b76
func77:tx_uc: 0x0000000ba3609e9c 00000000000bbd36
NIC MIB TX BC Counter:
func1:tx_bc: 0x0000000000000790 000000000000002e
func77:tx_bc: 0x000000000000002a 0000000000000001
NIC MIB TX MC Counter:
func0:tx_mc: 0x0000000000001930 0000000000000048
func1:tx_mc: 0x000000000000ae3a 00000000000001e3
func2:tx_mc: 0x0000000000001944 0000000000000048
func3:tx_mc: 0x0000000000001930 0000000000000048
```

```
func76:tx_mc: 0x00000000000001b6 0000000000000005
……
```

Query the statistics of a specified sequence of the microcode type of the IN200.

```
[root@localhost tool]# hinicadm counter -i hinic0 -t 1 -x 1
uCode Statistics:
NIC MIB TX BC Counter:
func1:tx_bc: 0x0000000000000790 000000000000002e
func77:tx_bc: 0x000000000000002a 0000000000000001
……
```

6.1.4.5.2 Parsing Inspection Information Offline (-o)

Function

The **counter -o** command is used to parse inspection information offline and save the parsing results to the installation directory of the tool.

Format

hinicadm counter -o <*countfile*>

Parameters

Parameter	Description	Value
countfile	Indicates the name of the inspection file to be parsed. **NOTE** This file is the out-of-band inspection information file of the IN200 obtained using the one-click information collection function of the iBMC. For details about the file obtaining method and save path, see the iBMC User Guide.	Example: **running_log.bin**

Usage Instruction

Before running this command, transfer the inspection information file to be parsed to the installation directory of the tool.

Example

Parse the inspection information of the IN200 offline.

```
[root@localhost tool]# hinicadm counter -o running_log_20180206095647.bin
/opt/hinic/fwlog/inspection_info_bmc_offline_2018_02_07_10_22_38.log create
succeed.
```

6.1.4.6 Querying the Linear Table of a Specified Device (table)

Function

The table command is used to query the linear table a specific device.

Format

hinicadm table -i <*devicename*> **-t** <*tabletype*>

hinicadm table -i <*devicename*> **-t** <*tabletype*> **-x** <*tableindex*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**
tabletype	Indicates the type of the linear table to be queried.	● **0**: MAC table ● **1**: global table ● **2**: function configuration table ● **3**: port configuration table ● **4**: VLAN table
tableindex	Indicates the index number of a linear table.	● If the value of **tabletype** is **1**, the value of **tableindex** ranges from **0** to **63**. ● If the value of **tabletype** is **2**, the value of **tableindex** ranges from **0** to **511**. ● If the value of **tabletype** is **3**, the value of **tableindex** ranges from **0** to **7**. ● If the value of **tabletype** is **4**, the value of **tableindex** ranges from **0** to **4095**.

Usage Instruction

None

Example

Query information about the linear table of the IN200.

```
[root@localhost tool]# hinicadm table -i hinic0 -t 0
static  er_id  mac               vlan_id  forward_type  forward_id
1     1     01:00:5e:00:00:01     0       6             9
1     1     33:33:00:00:00:01     0       6             8
1     3     00:02:03:04:05:09     0       0             3
1     2     00:02:03:04:05:08     0       0             2
1     1     00:02:03:04:05:07     0       0             1
1     0     00:02:03:04:05:06     0       0             0
```

#Query information about a linear table of a specified type.

```
[root@localhost sbin]# hinicadm table -i hinic0 -t 4 -x 12
node: 10 instance : 3 entry_size : 16 index : 12.
```



```
tag_sml_vlan_tbl_elem[0]_learn_en : 0
tag_sml_vlan_tbl_elem[0]_elb_index : 0
tag_sml_vlan_tbl_elem[1]_learn_en : 0
tag_sml_vlan_tbl_elem[1]_elb_index : 0
tag_sml_vlan_tbl_elem[2]_learn_en : 0
tag_sml_vlan_tbl_elem[2]_elb_index : 0
tag_sml_vlan_tbl_elem[3]_learn_en : 0
tag_sml_vlan_tbl_elem[3]_elb_index : 0
tag_sml_vlan_tbl_elem[4]_learn_en : 0
tag_sml_vlan_tbl_elem[4]_elb_index : 0
tag_sml_vlan_tbl_elem[5]_learn_en : 0
tag_sml_vlan_tbl_elem[5]_elb_index : 0
tag_sml_vlan_tbl_elem[6]_learn_en : 0
tag_sml_vlan_tbl_elem[6]_elb_index : 0
tag_sml_vlan_tbl_elem[7]_learn_en : 0
tag_sml_vlan_tbl_elem[7]_elb_index : 0
```

6.1.4.7 Querying MAC Addresses (mac)

Function

The **mac** command is used to query the MAC address of a specified device.

Format

hinicadm mac -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the MAC address of the IN200.

```
[root@localhost tool]# hinicadm mac -i hinic0
mac[0]: 00:02:03:04:05:06
mac[1]: 00:02:03:04:05:07
mac[2]: 00:02:03:04:05:08
mac[3]: 00:02:03:04:05:09
mac[4]: 00:02:03:04:05:10
mac[5]: 00:02:03:04:05:11
mac[6]: 00:02:03:04:05:12
mac[7]: 00:02:03:04:05:13
mac[8]: 00:02:03:04:05:14
mac[9]: 00:02:03:04:05:15
mac[10]: 00:02:03:04:05:16
mac[11]: 00:02:03:04:05:17
mac[12]: 00:02:03:04:05:18
mac[13]: 00:02:03:04:05:19
```



```
mac[14]: 00:02:03:04:05:20
mac[15]: 00:02:03:04:05:21
```

6.1.4.8 Querying and Setting the Port FEC Mode (fec)

6.1.4.8.1 Querying and Setting the Port FEC Mode (-p)

Function

The **fec -p** command is used to query set the forward error correction (FEC) mode for a port of a specified device.

Format

hinicadm fec -i <*devicename*> **-p** <*portid*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set or queried in the system.	Example: **hinic0** and **hinic1**
portid	Indicates the number of the port to be set or queried.	-

Usage Instruction

None

Example

Query the FEC mode of port 0 of the IN200.

```
[root@localhost tool]# hinicadm fec -i hinic0 -p 0
Port0 FEC mode: nofec
```

6.1.4.8.2 Setting the Port FEC Mode (-p -m)

Function

The **fec -p -m** command is used to set the forward error correction (FEC) mode for a port of a specified device.

Format

hinicadm fec -i <*devicename*> **-p** <*portid*> **-m** <*fecmode*> [**-d**]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set or queried in the system.	Example: **hinic0** and **hinic1**
portid	Indicates the number of the port to be set or queried.	-
fecmode	Indicates the FEC mode.	● **rsfec**: Reed Solomon Forward Error Correction (RSFEC) ● **basefe**: basic FEC mode ● **nofec**: non-FEC

Usage Instruction

- If the command contains **-d**, the setting is permanent and takes effect after a restart.
- If the command does not contain **-d**, the setting is not permanent and will be restored to the factory setting after a restart.
- The FEC mode takes effect only when the current optical module supports the FEC mode or an optical module that supports the FEC mode is install; otherwise, the original mode is used.

Example

Set the FEC mode of port 0 of the IN200 to **rsfec** and use persistent configurations.

```
[root@localhost tool]# hinicadm nic_fec -i hinic0 -p 0 -m rsfec -d
Set default FEC mode succeed.
Active default configuration succeed.
```

6.1.4.8.3 Clearing Existing Configurations (-p -c)

Function

The **fec -p -c** command is used to clear the existing configurations and restore the default configurations of the FEC, auto-negotiation, and forced rate.

Format

hinicadm fec -i <*devicename*> **-p** <*portid*> **-c**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set or queried in the system.	Example: **hinic0** and **hinic1**
portid	Indicates the number of the port to be set or queried.	-

Usage Instruction

None

Example

Clear the existing configurations and restore the default configurations of the FEC, auto-negotiation, and forcible rate.

```
[root@localhost tool]# hinicadm fec -i hinic0 -p 0 -c
Clear default configuration succeed.
Active default configuration succeed.
```

6.1.4.9 Querying and Setting a Port Rate Limit (rate)

Function

The **rate** command is used to query and set the rate limit parameters for the sending direction of a specified device port.

Format

hinicadm rate -i <*devicename*>

hinicadm rate -i <*devicename*> **-l** <*limitvalue*> [**-d**]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 port to be set or queried in the system.	Example: **eth0** and **eth1**
limitvalue	Indicates the percentage of the actual transmit rate to the maximum rate of the port.	An integer ranging from 1 to 100

Usage Instruction

- If the command contains **-d**, the setting is permanent and takes effect after a restart.
- If the command does not contain **-d**, the setting is not permanent and will be restored to the factory setting after a restart.

Example

Set the rate limit of port eth2 to 50%.

```
[root@localhost tool]# hinicadm rate -i eth2 -l 50
Set eth2 tx limit rate to 50% succeed
```

Query the rate limit of port eth2.

```
[root@localhost tool]# hinicadm rate -i eth2
eth2 tx limit rate: 50%.
eth2 default tx limit rate: 100%.
```

6.1.4.10 Querying the Chip and Optical Module Temperatures (temperature)

Function

The **temperature** command is used to query and set the chip and optical module temperatures of a specified IN200.

Format

hinicadm temperature -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the chip and optical module temperatures of the IN200.

```
[root@localhost tool]# hinicadm temperature -i hinic0
current temperature (unit: degree centigrade)
controller: 55
sfp1: 59
sfp2: absent
sfp3: NA
sfp4: 56
```


6.1.4.11 Querying Asynchronous Event Statistics (event)

Function

The **event** command is used to query the asynchronous event statistics of a specified device.

Format

hinicadm event -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query asynchronous event statistics of the IN200.

```
[root@localhost tool]# hinicadm event -i hinic0
SDK Statistics:

Heartbeat Lost Counter:
heartbeat lost: 0

CQM Counter:
cqm cmd alloc: 0
cqm cmd free: 0
cqm send cmd box: 0
cqm send cmd imm: 0
cqm db addr alloc: 0
cqm db addr free: 0
cqm fc srq create: 0
cqm srq create: 0
cqm rq create: 0
cqm qpc mpt create: 0
cqm nonrdma queue create: 0
cqm rdma queue create: 0
cqm rdma table create: 0
cqm qpc mpt delete: 0
cqm nonrdma queue delete: 0
cqm rdma queue delete: 0
cqm rdma table delete: 0
cqm func timer clear: 0
cqm func hash buf clear: 0
cqm scq callback: 0
cqm ecq callback: 0
cqm nocq callback: 0

Link Event Counter:
link down: 0
link up: 0
```



```
pcie fault: 0

Chip Faults info:
module_id err_level er_type stats
0x02      0x01      0x6b    0x04
0x05      0x01      0x8c    0x04
```

6.1.4.12 Clearing Statistics of a Specified Device (clear)

Function

The **clear** command is used to clear the statistics of a specified type of a specified device type.

Format

hinicadm clear -i <*nicdevicename*> **-t** <*type*>

Parameters

Parameter	Description	Value
nicdevicename	Indicates the name of the IN200 whose statistics are to be cleared in the system.	Example: **hinic0** and **hinic1** **NOTE** ● If **type** is set to 1 or 2, the value can be **eth0**. ● If the object name used to clear the driver statistics is a device name (for example, hinic0), the statistics of all network devices on hinic0 are cleared. If the object name used to clear the driver statistics is a network device name (for example, eth0), the statistics on the specified network device (eth0) are cleared.
type	Indicates the statistics type to be cleared.	● **0**: asynchronous information statistics ● **1**: DFX information statistics ● 2: Basic I/O statistics

Usage Instruction

When type=2, the device name must be the network port name.

Example

Clear the asynchronous event statistics of the IN200.

```
[root@localhost]# hinicadm clear -i hinic0 -t 0
Clear event stats succeed.
```

Clear the I/O statistics of the eth0 port on the IN200.

```
[root@localhost]# hinicadm clear -i eth0 -t 2
Clear driver stats succeed.
Clear vport stats succeed.
Clear port stats succeed.
```


6.1.4.13 Querying and Setting the User Priority and CGE Pause Time (qos)

Function

The **qos** command is used to query and set the user priority and CGE pause time of a specified device.

Format

hinicadm qos -i <*devicename*>

hinicadm qos -i <*devicename*> **-c** <qoscos>

hinicadm qos -i <device*name*> **-t** <cge pause time> **-p** <port id>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose user priority is to be queried and set in the system.	Example: **hinic0** and **hinic1**
qoscos	Indicates the user priority sequence to be configured.	● The value range of each element in the sequence is 0 to 7. ● The sequence must be in descending mode. ● The number of elements contained in the sequence must be the same as the number of supported classes of service (CoSs) (the number of CoSs can be obtained by running the query command).
cge pause time	Indicates the pause frame flow control time of the CGE module on the current chip.	-
port id	Indicates the physical port number.	Example: **0**

Usage Instruction

None

Example

Set the user priority of the IN200 to **6 5 3 0**.

```
[root@localhost tool]# hinicadm qos -i hinic0 -c 6530
Set cos2up succeed.
```

Query the user priority of the IN200.


```
[root@localhost tool]# hinicadm qos -i hinic0
cos number: 4
user priority:       6  5  3  0
```

Set the CGE pause time for port 0.

```
[root@localhost tool]# hinicadm qos -i hinic0 -t 65535 -p 0

qos command error(-6): Only cge card could set pause time.
```

6.1.4.14 Querying the RegisterInformation of a Specified Device (reg)

Function

The **reg** command is used to query the current register value of a specified device.

Format

hinicadm reg -i <*devicename*> **-t** <*registertype*> **-a** <*address*> [**-n** <num>] [**-c** <*channel*>]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** and **hinic1**
registertype	Indicates the type of the register to be queried	● **0**: mag register ● **1**: read register
address	Indicates the address of the register to be queried	The value must be a hexadecimal address.
num	Indicates the number of registers to be queried.	The value must range from 1 to 32.
channel	Indicates the channel type of registers to be queried.	The value must range from 0 to 1.

Usage Instruction

If the command does not contain **-n**, it queries only the value of a register whose address has been specified.

Example

Query the current values of the three consecutive registers starting from 0x1022C in the IN200.

```
[root@localhost tool]# hinicadm reg -i hinic0 -t 0 -a 0x1022c -n 3
addr:       0x1022c
val[0] = 0x00
val[1] = 0x00
val[2] = 0x00
```


Query the value of the read register whose address is **0x1d00000**.

```
[root@localhost tool]# hinicadm reg -i hinic0 -t 1 -a 0x1d00000
addr:       0x1d00000
data:       0x182219e5
```

6.1.4.15 Querying and Setting the Auto-Negotiation Mode of a Specified Device (autoneg)

Function

The **autoneg** command is used to query and set the port auto-negotiation mode of a specified device.

Format

hinicadm autoneg -i <*devicename*> **-p** <*portid*>

hinicadm autoneg -i <*devicename*> **-p** <*portid*> **-m** <*mode*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose port auto-negotiation mode is to be queried and set in the system.	Example: **hinic0** and **hinic1**
portid	Indicates the number of the port to be queried or set.	-
mode	Indicates the auto-negotiation mode to be set.	● **25g**: auto-negotiation mode defined by the 25G Ethernet Consortium ● **ieee**: auto-negotiation mode defined by IEEE For details about the differences between the two modes, see the protocol details of the IEEE 802.3 and 25G Ethernet Consortium.

Usage Instruction

None

Example

Set the auto-negotiation mode of port 1 on the IN200 to **ieee**.

```
[root@localhost tool]# hinicadm autoneg -i hinic0 -p 1 -m ieee
Set port1 auto-negotiation mode to ieee mode succeed.
```

Query the auto-negotiation mode of port 1 on the IN200.


```
[root@localhost tool]# hinicadm autoneg -i hinic0 -p 1
Port1 auto-negotiation mode: ieee
```

6.1.4.16 Querying the Queue Information of a Specified Device (nic_queue)

Function

The **nic_queue** command is used to query the queue information of a specified network device.

Format

hinicadm nic_queue -i <*ethdevicename*> **-d** <*direction*> **-t** <*type*> **-q** <queueid> [**-w** <*wqeid*>]

Parameters

Parameter	Description	Value
ethdevicename	Indicates the name of the network device to be queried in the system.	Example: **eth0** and **eth1**
direction	Indicates the direction of the queue to be queried.	● **0**: sending direction (tx) ● **1**: receiving direction (rx)
type	Indicates the information type to be queried.	● **0**: queue information ● **1**: work queue entry (WQE) information. ● **2**: CI table information (only tx) ● **3**: completion queue entry (CQE) information (only rx)
queueid	Indicates the ID of the queue to be queried.	-
wqeid	Indicates the ID of the WQE to be queried.	-

Usage Instruction

None

Example

Query the queue information of eth0 on the IN200.

```
[root@localhost tool]# hinicadm nic_queue -i eth0 -d 1 -t 0 -q 0
Receive queue0 information:
```



```
queue id:0
hw_pi:1023
ci:5
sw_pi:1023
rq_depth:1024
rq_wqebb_size:32
……
```

6.1.4.17 Querying and Setting the Work Mode of a Specified Device (mode)

Function

The **mode** command is used to query and set the work mode of a specified device.

Format

hinicadm mode -i <*devicename*>

hinicadm mode -i <*devicename*> **-m** <*workmode*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried and configured in the system.	Example: **hinic0** or **hinic1**
workmode	Indicates the work mode to be configured.	● **nic**: indicates the standard NIC work mode. ● **ovs**: indicates the open virtual switch work mode.

Usage Instruction

Restart the operating system for the setting to take effect.

Example

Set the work mode to **ovs**.

```
[root@localhost tool]# hinicadm mode -i hinic0 -m ovs
Set work mode to ovs succeed.
Please reboot OS for the new mode to take effect.
```

Query the IN200 work mode.

```
[root@localhost tool]# hinicadm mode -i hinic0
Current work mode: NIC
Work mode in permanent configuration: OVS
```

6.1.4.18 Querying FE Error Information (fe_epc)

6.1.4.18.1 Obtaining FE Error Information of a Specified Device (show)

Function

The **fe_epc show** command is used to obtain FE error information of a specified device.

Format

hinicadm fe_epc -i <*devicename*> **show -c** <*coreid*> **-t** <*tid*> **-n** <*num*>

hinicadm fe_epc -i <*devicename*> **show -b**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
coreid	Indicates the core ID to be queried.	-
tid	Indicates the thread ID to be queried.	-
num	Indicates the number of FE error records.	-

Usage Instruction

The **-b** is used to display all abnormal records.

Example

Query all the error information about history FE threads of the IN200.

```
[root@localhost tool]# hinicadm fe_epc -i hinic0 show -b
index: 192
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
vaild:0x0
excp_code: 0x0
core_id:0x0
t_id:0x0
I07_status:0x0
I06_status:0x0
I05_status:0x0
I04_status:0x0
I03_status:0x0
I02_status:0x0
I01_status:0x0
I00_status:0x0
;;
```


6.1.4.18.2 Triggering a FE Thread and Recording Error Information (-c -t)

Function

The **fe_epc** command is used to trigger a FE error information report for a specified device.

Format

hinicadm fe_epc -i <*devicename*> **-c** <*coreid*> **-t** <*tid*>

hinicadm fe_epc -i <*devicename*> **-a**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
coreid	Indicates the core ID to be queried.	-
tid	Indicates the thread ID to be queried.	-

Usage Instruction

If the command contains **-a**, all the threads are triggered.

Example

Query all the error information about all FE threads of the IN200.

```
[root@localhost tool]# hinicadm fe_epc -i hinic0 -a
index: 192
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
vaild:0x0
excp_code: 0x0
core_id:0x0
t_id:0x0
I07_status:0x0
I06_status:0x0
I05_status:0x0
I04_status:0x0
I03_status:0x0
I02_status:0x0
I01_status:0x0
I00_status:0x0
;;
```


6.1.4.19 Querying Basic Information of a Specified Port (hilink_port)

Function

The **hilink_port** command is used to query basic information of a specified port.

Format

hinicadm hilink_port -i <*devicename*> **-p** <*portid*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of a port to be queried.	-

Usage Instruction

None

Example

Queries basic information of port 0.

```
[root@localhost tool]# hinicadm hilink_port -i hinic0 -p 0
gpio_insert = present
link_state = no link
an_state = on
alos = 1
rx_los = 0
speed = 25
fec = rsfec
rf_lf = 0x0
pma_fifo_reg = 0x00000029
pma_signal_ok_reg = 0x00000001
pcs_64_66b_reg = 0x00001001
pcs_err_cnt = 0x00000000
pcs_link = no link
vendorName = Amphenol
port_type = copper
port_sub_type = unknow
cable_length = 3(m)
cable_temperature = 0
max_speed = 25Gbps
sfp_type = sfp
tx_power = 0uW
rx_power = 0uW
```


6.1.4.20 Querying Statistics of a Specified Port (hilink_count)

Function

The **hilink_count** command is used to query statistics of a specified port.

Format

hinicadm hilink_count -i <*devicename*> **-p** <*portid*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of a port to be queried.	-

Usage Instruction

None

Example

Query statistics of port 0.

```
[root@localhost tool]# hinicadm hilink_count -i hinic0 -p 0
err_bloack = 0x00000000
base_fec = 0x00000000 0x00000000
rs_fec = 0x00000000 0x00000000 0x00000000
base_fec_idle = 0x00000000
MAG RXTP TXDP:
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x00000000 0x00000000
;;
```

6.1.4.21 Quering Information About the an_train Register (hilink_dump)

Function

The **hilink_dump** command is used to query configuration of the an_train register in hi30 mode.

Format

hinicadm hilink_dump -i <*devicename*> **-p** <*portid*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the port number to be queried.	-

Usage Instruction

None

Example

Queries information about the an_train register numbered 0.

```
[root@localhost tool]# hinicadm hilink_dump -i hinic0 -p 0
an train:
lane 0:
00000000 00000007 00000000 00000000 00000000 0020042b 00000021 00460023 ffffffff
0004c4b4 01e08c01 00000404 00000001 00000000 00000000 00000000
```

6.1.4.22 Querying Physical Parameters of a Device in Specified Mode (hilink_param)

Function

The **hilink_param** command is used to query the **ctle**, **dfe**, and **ffe** parameters of a device in specified mode.

Format

hinicadm hilink_param -i <*devicename*> **-t**<*type*> [**-p** <*portid*>]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
type	Indicates the mode for the query.	0: indicates the PCIe mode. 1: indicates the NIC mode.
portid	Indicates the number of a port to be queried.	This parameter takes effect when *type* is set to **1**.

Usage Instruction

None

Example

Queries parameters of the PCIe mode.

```
[root@localhost tool]# hinicadm hilink_param -i hinic0 -t 0
HI16 parameter info:

 lane 0 :
     TX_FFE: pre= 8;main= d;post= a
     RX_CTLE: PASSGN=-2dB; ACTGN= 6 6 7; BST= 8 4 4; ZA= 1 1 1; SQH= 1 1 1;
RMBAND= 1 1 1; CMBAND= 1 1 1;
     RX_DFE1: Tap1=0; Tap2=2; Tap3=2; Tap4=2; Tap5=2;
     RX_DFE2: Tap1=0; Tap2=0; Tap3=0; Tap4=0; Tap5=0; Tap6=0;
;;
```

6.1.4.23 Querying and Setting Port Rate Parameters (hilink_speed)

6.1.4.23.1 Querying the Transmission Rate of a Specified Port

Function

The **hilink_speed** command is used to query the transmission rate of a specified port.

Format

hinicadm hilink_speed -i <*devicename*> **-p** <*portid*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be configured in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of the port to be set.	-

Usage Instruction

None

Example

Query the transmission rate of port 0.

```
[root@localhost tool]# hinicadm hilink_speed -i hinic0 -p 0
Port0 auto-negotiation: on
Port0 force speed: 25GE
Port0 default GE mode: disable
```



```
Port0 default auto-negotiation: on
Port0 default force speed: 25GE
```

6.1.4.23.2 Setting the Transmission Rate of a Specified Port (-s)

Function

The **hilink_speed -s** command is used to set the transmission rate of a specified port.

Format

hinicadm hilink_speed -i <*devicename*> **-p** <*portid*> **-s** <*speed*> [**-d**]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be configured in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of the port to be set.	-
speed	Indicates the port rate when the auto-negotiation mode is disabled. The unit is GB.	10 or 25

Usage Instruction

- If the command contains **-d**, the setting is permanent and takes effect after a restart.
- If the command does not contain **-d**, the setting is not permanent and will be restored to the factory setting after a restart.

Example

Set the transmission rate of port 0 to 25 GB and make the setting take effect permanently.

```
[root@localhost tool]# hinicadm hilink_speed -i hinic0 -p 0 -s 25 -d
Set forced speed succeed.
Set default configuration succeed.
Active default configuration succeed.
```

6.1.4.23.3 Setting the Auto-Negotiation Mode of a Specified Port (-an)

Function

The **hilink_speed -an** command is used to enable or disable the auto-negotiation mode of a specified port.

Format

hinicadm hilink_speed -i <*devicename*> **-p** <*portid*> **-an** <*state*> [**-d**]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be configured in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of the port to be set.	-
state	Indicates whether the auto-negotiation mode is enabled.	● on ● off

Usage Instruction

- If the command contains **-d**, the setting is permanent and takes effect after a restart.
- If the command does not contain **-d**, the setting is not permanent and will be restored to the factory setting after a restart.

Example

Enable the auto-negotiation mode of port 0 and make the setting take effect permanently.

```
[root@localhost tool]# hinicadm hilink_speed -i hinic0 -p 0 -an on -d
Set auto-negotiation succeed.
Set default configuration succeed.
Active default configuration succeed.
```

6.1.4.23.4 Setting the Network Port Connection Mode (-m)

Function

The **hilink_speed -m** command is used to set the network port connection mode.

Format

hinicadm hilink_speed -i <*devicename*> **-p** <*portid*> **-m** <*mode*> [**-d**]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be configured in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of the port to be set.	-

Parameter	Description	Value
mode	Indicates the network port connection mode.	● enabled: GE transmission is used. ● disabled: 10GE/25GE transmission is used.

Usage Instruction

- If the command contains **-d**, the setting is permanent and takes effect after a restart.
- If the command does not contain **-d**, the setting is not permanent and will be restored to the factory setting after a restart.
- The mode setting of port 0 correlates with port 1, and the setting of port 2 correlates with port 3. For example, if mode is changed from **enabled** to **disabled** for port 0 and takes effect, mode will be automatically changed to **disabled** for port 1.
- Setting the GE mode takes effect after restart. After the GE mode is adopted, the FEC, forced rate, and auto-negotiation cannot be configured.

Example

#Enable the GE mode for port 0.

```
[root@localhost ~]# hinicadm hilink_speed -i hinic0 -p 0 -m enable
Set default configuration succeed.
Active default configuration succeed.
```

6.1.4.23.5 Clearing Rate Permanence Configuration of a Specified Port (-c)

Function

The **hilink_speed -c** command is used to clear rate permanence configuration of a specified port, namely, **-d** property in **6.1.4.23.2 Setting the Transmission Rate of a Specified Port (-s)**.

Format

hinicadm hilink_speed -i <*devicename*> **-p** <*portid*> **-c**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be configured in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of the port to be set.	-

Usage Instruction

Running this command does not affect server running. After the system restarts, the rate configuration of the port is restored to the factory setting.

Example

Clear the rate permanence configuration of port 0.

```
[root@localhost tool]# hinicadm hilink_speed -i hinic0 -p 0 -c
Clear default configuration succeed.
Active default configuration succeed.
```

6.1.4.24 Querying the serdes Information of a Specified Configuration Type (serdes)

Function

The **serdes** command is used to query the serdes information of a specified configuration type.

Format

hinicadm serdes -i <*devicename*>

hinicadm serdes -i <*devicename*> **-t** <hilink_type> **-m** [<macro>]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 or port to be queried in the system.	Example: **hinic0**, **eth0**, and **eth1**
hilink_type	Indicates the configuration type.	**0**: hi16 and hi30 **1**: hi16 or hi30
macro	Indicates the macro digit.	-

Usage Instruction

If the value of **ethdevicename** is an IN200 name, this command queies serdes information of all ports of the NIC.

Example

Queries serdes information of the hinic0 port.

```
[root@localhost tool]# hinicadm serdes -i hinic0 -t 0
--------------------show hi16 info--------------------
macro is 0  dsnum is 0
cs0_csr2 = 0 (success)
cs0_csr_51 = 0 (success)
```

```
cs0_csr54 = 11882
cs0_csr63 = 0
cs1_csr2 = 0 (success)
cs1_csr_51 = 0 (success)
cs1_csr54 = 6254
cs1_csr63 = 529
dsclk_csr0 = 0 (success)
dsclk_csr24 = 0 (success)
dsclk_csr27 = 38352
tx_csr2 = 180
tx_csr26 = 51100
rx_csr10 = 180
rx_csr26 = 24703
rx_csr61 = 6164
tx_csr_rw_result = 0 (success)
rx_csr_rw_result = 0 (success)
tx_csr48 = 49163
rx_csr10 = 1
eye_top = 15
......
```

6.1.4.25 Querying Optical Module Information of a Specified Port (sfp)

Function

The **sfp** command is used to query optical module information of a port.

Format

hinicadm sfp -i <*devicename*> **-p** <*portid*> [**-a**]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
portid	Indicates the number of a port to be queried.	This parameter takes effect when *type* is set to **1**.

Usage Instruction

If the command contains **-a**, all information about the optical module is displayed in hexadecimal format.

Example

Queries optical module information of port 0.

```
[root@localhost tool]#hinicadm sfp -i hinic0 -p 0
--------------------show sfp info--------------------
ucId = 0x03
ucIdExt = 0x04
ucConnector = 0x21
ucEncoding = 6
ucBrNominal = 255 (100MBd)
```



```
ucRateIdentifier = 0
ucLengthSmfKm = 0(km)
ucLengthSmf = 0(100m)
ucLengthSmfOm2 = 0(10m)
ucLengthSmfOm1 = 0(10m)
ucLengthCable = 3(m)
ucLengthOm3 = 0(m)
aucVendorName = Amphenol
aucVendorSN = 0215761612073874
ucTransceiver = 0
aucVendorOui = 78 a7 14
aucVendorPn = NDCCGF-H203
aucVendorRev = A
aucWaveLength = 256
ucUnAllocated = 0
ucCcBase = 35
aucTemperature = NA (DAC)
--------------------------------------------------
```

6.1.4.26 Querying and Setting the Virtualization Control Status of a Port (sriov)

Function

The **sriov** command is used to query and set the virtualization control status of a port.

Format

hinicadm sriov -i <*devicename*> **-p** <*portid*>

hinicadm sriov -i <*devicename*> **-p** <*portid*> **-s** <*sriov*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried or set in the system.	Example: **hinic0** and **hinic1**
portid	Indicates the number of the port to be queried or set in the system.	-
sriov	Indicates the status of the virtualization control mode.	● enable ● disable

Usage Instruction

None

Example

Query the virtualization control status of port 0.

```
[root@localhost tool]# hinicadm sriov -i hinic0 -p 0
Pf0 sriov control status: disable.
```


Enable virtualization control for port 0.

```
[root@localhost tool]# hinicadm sriov -i hinic0 -p 0 -s enable
Set pf0 sriov control to enable succeed.
```

6.1.4.27 Restoring Factory Settings (reset)

Function

The **reset** command is used to restore factory settings of an IN200 port.

Format

hinicadm reset -i <*devicename*> [**-p** <*portid*>]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**
portid	Indicates the number of the target port.	-

Usage Instruction

Restart the OS for the operation to take effect after the command is executed.

Example

Restore factory settings of all IN200 ports.

```
[root@localhost tool]#hinicadm reset -i hinic0

WARNING! reset operation will cause all data be cleared.
WARNING! You have selected to do reset operation.
Proceed with the reset? (Y|N): Y
Pf0 restore factory settings succeed.
Pf1 restore factory settings succeed.
Pf2 restore factory settings succeed.
Pf3 restore factory settings succeed.
Please reboot OS to take effect.
```

6.1.4.28 Querying Chip Back Pressure Information (bp)

Function

The **bp** command is used to query chip back pressure information of the IN200.

Format

hinicadm bp -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query chip back pressure information of the IN200.

```
[root@localhost tool]#hinicadm bp -i hinic0
===[ Hi1822 backpressure info]===
********** Hi1822 TX bp history counter **********
*** TX link level ***


*** TX sub level ***

********** Hi1822 RX bp history counter **********
*** RX link level bp ***


********** Hi1822 TX bp status **********
*** TX link level bp ***


*** TX sub_level bp  ***


********** Hi1822 RX bp status **********
*** RX link level bp ***


*** RX sub_level bp  ***
```

6.1.4.29 Querying Statistics on Chip Data Path Modules (dp)

Function

The **dp** command is used to query statistics on chip data path modules.

Format

hinicadm dp -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query statistics on chip data path modules.

```
[root@localhost tool]#hinicadm dp -i hinic0
******** CPI GLOBAL STATIC ********
dwqe_api2sm    dwqe_dbe2mqm   nm_dbe2mqm     nm_dbe_drop
0              0              60             0

******** MQM statistics ********
rx_cpi_pkt     rx_sm_pkt      rx_tile_pkt    deq_iqm          deq_eqm
dis_cpi        filterd_db
60             0              0              60             0
0              0

age_drop       age2eqm        iqm2qu         iqm2sm
0              0              0              60
```

6.1.4.30 Querying the Status of a Microcode Threads (tile_io)

Function

The **tile_io** command is used to query the status of microcode threads.

Format

hinicadm tile_io -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the status of microcode threads.

```
[root@localhost tool]#hinicadm tile_io -i hinic0
*************** tile IO status ***************

core0_tc0_IO_0 : WAIT  core0_tc1_IO_0 : WAIT  core0_tc2_IO_0 : WAIT
core0_tc3_IO_0 : WAIT
core1_tc0_IO_0 : WAIT  core1_tc1_IO_0 : WAIT  core1_tc2_IO_0 : WAIT
core1_tc3_IO_0 : WAIT
core2_tc0_IO_0 : WAIT  core2_tc1_IO_0 : WAIT  core2_tc2_IO_0 : WAIT
core2_tc3_IO_0 : WAIT
core3_tc0_IO_0 : WAIT  core3_tc1_IO_0 : WAIT  core3_tc2_IO_0 : WAIT
```



```
core3_tc3_IO_0 : WAIT
core4_tc0_IO_0 : WAIT  core4_tc1_IO_0 : WAIT  core4_tc2_IO_0 : WAIT
core4_tc3_IO_0 : WAIT
core5_tc0_IO_0 : WAIT  core5_tc1_IO_0 : WAIT  core5_tc2_IO_0 : WAIT
core5_tc3_IO_0 : WAIT
core6_tc0_IO_0 : WAIT  core6_tc1_IO_0 : WAIT  core6_tc2_IO_0 : WAIT
core6_tc3_IO_0 : WAIT
core7_tc0_IO_0 : WAIT  core7_tc1_IO_0 : WAIT  core7_tc2_IO_0 : WAIT
core7_tc3_IO_0 : WAIT
core8_tc0_IO_0 : WAIT  core8_tc1_IO_0 : WAIT  core8_tc2_IO_0 : WAIT
core8_tc3_IO_0 : WAIT
```

6.1.4.31 Querying the Usage of Chip CPB Cells (cpb)

Function

The **cpb** command is used to query the usage of chip CPB cells.

Format

hinicadm cpb -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the usage of chip CPB cells.

```
[root@localhost tool]#hinicadm cpb -i hinic0
****** CPB CELL NUM STATIC ******
total_cell_num    :  30720
free_cell_num     :  30720
pdm_glb_num       :  178
cpi_octl_cell_num :  178
leak_cell_num     :  0
fq_free_oeid_num  :  511
```

6.1.4.32 Reading Chip Register (csr_rd)

Function

The **csr_rd** command is used to read the value of a chip register.

Format

hinicadm csr_rd -i <*devicename*> **-m** <*module name*> **-a** <*address*> [**-x** <*index*>]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**
module name	Module type.	**cpb**, **ipsurx**, **ipsutx**, **qu**, **pe**, **esch**, **mqm**, **lcam**, **sml0**, **sml1**, **smf**, **tile0**, **ppe0**, **tile1**, or **ppe1**
address	Relative address of the register module	-
index	Indirect register index	-

Usage Instruction

None

Example

Read the register value of the hinic0 chip.

```
[root@localhost tool]# hinicadm csr_rd -i hinic0 -m cpb -a 0x64
```

6.1.4.33 Writing Chip Register (csr_wr)

Function

The **csr_wr** command is used to write the value of a chip register.

Format

hinicadm csr_wr -i <*devicename*> **-m** <*module name*> **-a** <*address*> [**-x** <*index*>] **-d** <*data*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**
module name	Module type.	**cpb**, **ipsurx**, **ipsutx**, **qu**, **pe**, **esch**, **mqm**, **lcam**, **sml0**, **sml1**, **smf**, **tile0**, **ppe0**, **tile1**, or **ppe1**
address	Relative address of the register module	**cpb**, **ipsurx**, **ipsutx**, **qu**, **pe**, **esch**, **mqm**, **lcam**, **sml0**, **sml1**, **smf**, **tile0**, **ppe0**, **tile1**, or **ppe1**
index	Indirect register index	-

Parameter	Description	Value
data	Register value	-

Usage Instruction

None

Example

Write the register value of the hinic0 chip.

```
[root@localhost tool]# hinicadm csr_wr -i hinic0 -m cpb -a 0x64 -d 0x3
```

6.1.4.34 Reading Chip Register Values in Batches (csr_dump)

Function

The **csr_dump** command is used to read and save the chip register values in batches.

Format

hinicadm csr_dump -i <*devicename*> **-m** <*module name*> **-t** <*type*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**
module name	Module type.	**cpb**, **ipsurx**, **ipsutx**, **qu**, **pe**, **esch**, **mqm**, **lcam**, **sml0**, **sml1**, **smf**, **tile0**, **ppe0**, **tile1**, **ppe1**, or **all**
type	Register type	**cfg**, **err**, **merr**, **int**, **cnt**, **ctp**, **cap**, **hst**, **mem**, **dummy**, or **all**

Usage Instruction

- Stop the services before running this command.
- Do not run this command twice on the same IN200.
- When this command is run, the registers of the chip cannot be read or written.

Example

Read and save the register values of the hinic0 chip in batches.

```
[root@localhost tool]# hinicadm csr_dump -i hinic0 -m cpb -t err
```


6.1.4.35 Queryingthe CPB CELL Resource Allocation and Current Resource Usage (pdm)

6.1.4.35.1 Querying the CPB CELL Resource Allocation and Current Resource Usage of an Index (-m -x)

Function

The **pdm -m -x** command is used to query the CPB CELL resource allocation and current resource usage of an index.

Forma

hinicadm pdm -i <*devicename*> **-m** <*module name*> **-x** <*index*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**
module name	Indicates the resource type.	**all**, **glb**, **pg**, **tx**, **rx**, **yzone**, **rzone**, **nport**, **ncos**, **host**, **hsrv**, **hport**, **hpri**, **hep**, or **hque**
index	Indicates the entry index of a resource type.	-

Usage Instruction

None

Example

Query the host resource information.

```
[root@localhost ~]# hinicadm pdm -i hinic0 -m host -x 0
****** HOST RESOURCE ******
tbl_idx        nm_th          mqm_th         gap_th         nm_cnt
host0          10292          10036          150            0
```

6.1.4.35.2 Querying the CPB CELL Resource Allocation and Current Resource Usage in a Specified Range (-m -s -e)

Function

The **pdm -m -s -e** command is used to query the CPB CELL resource allocation and current resource usage in a specified range.

Format

hinicadm pdm -i <*devicename*> **-m** <*module name*> **-s** <*start index*> **-e** <*end index*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**
module name	Indicates the resource type.	**all**, **glb**, **pg**, **tx**, **rx**, **yzone**, **rzone**, **nport**, **ncos**, **host**, **hsrv**, **hport**, **hpri**, **hep**, or **hque**
start index	Indicates the start index of a resource type. Each resource type has a unique index range.	-
end index	Indicates the end index of a resource type. Each resource type has a unique index range, which must be greater than or equal to the start index.	-

Usage Instruction

None

Example

Query the host resource information in a specified range.

```
[root@localhost ~]# hinicadm pdm -i hinic0 -m host -s 0 -e 3
****** HOST RESOURCE ******
tbl_idx         nm_th           mqm_th          gap_th          nm_cnt
host0           10292           10036           150             0
host1           10292           10036           150             0
host2           10292           10036           150             0
host3           10292           10036           150             0
```

6.1.4.36 Querying the Interrupt Information Reported by a Hardware Module (fm_show)

Function

The **fm_show** command is used to query the interrupt information reported by a hardware module.

Format

hinicadm fm_show -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Display the interrupt information reported by the hinic0 hardware module.

```
[root@localhost ~]# hinicadm fm_show -i hinic0
********no fm info!********
```

6.1.4.37 Querying and Setting the Status of the Port Auto-adaptation Mode (self_adaption)

Function

The **self_adaption** command is used to query and set the status of the port auto-adaptation mode.

Format

hinicadm self_adaption -i <*devicename*> **-p** <*portid*>

hinicadm self_adaption -i <*devicename*> **-p** <*portid*> **-s** <*status*> [**-d**]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the target IN200 in the system.	Example: **hinic0** and **hinic1**
portid	Indicates the target port number.	-
status	Indicates the status of the port auto-adaptation mode.	● enable ● disable (default value)

Usage Instruction

- If the command contains **-d**, the setting is permanent and takes effect after a restart.
- If the command does not contain **-d**, the setting is not permanent and will be restored to the factory setting after a restart.

Example

Query the auto-adaptation status of port 0.

```
[root@localhost tool]# hinicadm self_adaption -i hinic0 -p 0
Port 0 link mode adaptation : on
```

Enable auto-adaptation for port 0.

```
[root@localhost tool]# hinicadm self_adaption -i hinic0 -p 0 -s on -d
Set default link mode adaptation succeed.
Active default configuration succeed.
Set default link cfg succeed.
```

6.1.4.38 Querying and Setting the LRO Coalesce Time (lro)

6.1.4.38.1 Querying the LRO Coalesce Time

Function

The **lro** command is used to set the LRO coalesce time.

Format

hinicadm lro -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the LRO coalesce time.

```
[root@localhost tool]# hinicadm lro -i hinic0
LRO coalesce time: 10us.
```

6.1.4.38.2 Setting the LRO Coalesce Time (-t)

Function

The **lro -t** command is used to set the LRO coalesce time.

Format

hinicadm lro -i<*devicename*> **-t** <*lro coalesce time*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**
lro coalesce time	Indicates the coalesce time. Unit: us	The value is an integer ranging from 1 to 1024.

Usage Instruction

None

Example

Set the LRO coalesce time.

```
[root@localhost tool]# hinicadm lro -i hinic0 -t 100
Set LRO timer to 100us succeed.
```

6.1.4.39 Querying the Statistics of Received and Sent Packets of a NIC Port (xstats)

Function

The **xstats** command is used to query the statistics of received and sent packets of a NIC port.

Format

./hinicadm.exe xstats -i<*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: hinic0 or hinic1

Usage Instruction

None

Example

Query the statistics of received and sent packets of Ethernet 10.

```
[root@localhost tool]#./hinicadm.exe xstats -i "Ethernet 10"
NIC statistics:
```



```
    tx_unicast_pkts_vport: 0
    tx_unicast_bytes_vport: 0
    tx_broadcast_pkts_vport: 0
    tx_broadcast_bytes_vport: 0
    rx_multicast_pkts_vport: 0
    rx_multicast_bytes_vport: 0
    rx_broadcast_pkts_vport: 0
      ……
```

6.1.4.40 Querying and Setting the Interrupt Coalescence Parameters (inter_coal)

Function

The **inter_coal** command is used to query or set the interrupt coalescence parameters.

Format

hinicadm inter_coal-i<*devicename*>

hinicadm inter_coal-i<*devicename*> [-c <coal_timer>] [-p <pend_limit>] [-rl <rate_low>] [-ul <usecs_low>] [-pl <pend_low>] [-rh <rate_high>] [-uh <usecs_high>] [-ph <pend_high>]

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried in the system.	Example: **hinic0** or **hinic1**
coal_timer	Indicates the coalescence time.	-
pend_limit	Indicates the number of coalescence packets.	-
rate_low	Indicates the lower threshold of the packet rate.	-
usecs_low	Indicates the lower threshold of the receive time.	-
pend_low	Indicates the lower coalescence threshold.	-
rate_high	Indicates the upper threshold of the packet rate.	-
usecs_high	Indicates the upper threshold of the receive time.	-
pend_high	Indicates the upper coalesce threshold.	-

Usage Instruction

None

Example

Query the interrupt coalesce parameters of hinic0.

```
[root@localhost tool]# hinicadm inter_coal -i hinic0
coalesce_timer_cfg: 32
pending_limt: 24
pkt_rate_low: 40000
rx_usecs_low: 16
rx_pending_limt_low: 2
pkt_rate_high: 100000
rx_usecs_high: 64
rx_pending_limt_high: 8
```

Set the interrupt coalesce parameters of hinic0.

```
[root@localhost tool]# hinicadm inter_coal -i hinic0 -c 64 -p 30 -rl 50000 -ul 8 -
pl 4 -rh 90000 -uh 10 -ph 10
Set interrupt coalesce parameters succeed.
```

6.1.4.41 Querying and Setting the SDI Card Mode (sdi_mode)

6.1.4.41.1 Querying the SDI Card Mode

Function

The **sdi_mode** command is used to query the SDI card mode.

Format

hinicadm sdi_mode -i <*devicename*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the SDI card mode.

```
linux-EXvCVE:~ # hinicadm sdi_mode -i hinic0
Current sdi mode: vm
SDI mode in permanent configuration: vm
```


6.1.4.41.2 Setting the SDI Card Mode (-m)

Function

The **sdi_mode -m** command is used to query and set the SDI card mode.

Format

hinicadm sdi_mode -i <*devicename*> **-m** <*mode*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**
mode	Indicates the SDI card mode.	● **vm**: virtual machine mode ● **bm**: bare-metal mode

Usage Instruction

None

Example

Set the SDI card mode.

```
linux-EXvCVE:~ # hinicadm sdi_mode -i hinic0 -m bm
Set sdi mode to bm succeed.
Please reboot OS for the new mode to take effect.
```

6.1.4.42 Querying and Setting the SDI Configuration (sdi_cfg)

The **sdi_cfg** command is used to query and set the SDI configuration, including the management VLAN and PF rate limit parameters.

6.1.4.42.1 Querying the Configured Management VLAN (-t cpath_vlan)

Function

The **-t cpath_vlan** command is used to query the configured management VLAN.

Format

hinicadm sdi_cfg -i <*devicename*> **-t cpath_vlan**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the configured management VLAN.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i hinic0 -t cpath_vlan
None control path vlan configured.
```

6.1.4.42.2 Configuring a Management VLAN (-t cpath_vlan -v -s)

Function

The **-t cpath_vlan-v -s** command is used to configure a management VLAN.

Format

hinicadm sdi_cfg -i <*devicename*> **-t cpath_vlan -v** <*vlan*> **-s**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**
vlan	Indicates the VLAN ID.	It is an integer ranging from 0 to 4095.

Usage Instruction

None

Example

Configure a management VLAN.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i hinic0 -t cpath_vlan -v 4095 -s
Set control path vlan 4095 succeed
```


6.1.4.42.3 Deleting the Configured Management VLAN (-t cpath_vlan -v -c)

Function

The **-t cpath_vlan -v -c** command is used to delete the configured management VLAN.

Format

hinicadm sdi_cfg -i <*devicename*> **-t cpath_vlan -v** <*vlan*> **-c**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**
vlan	Indicates the VLAN ID.	It is an integer ranging from 0 to 4095.

Usage Instruction

None

Example

Delete the configured management VLAN.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i hinic0 -t cpath_vlan -v 4095 -c
Clear control path vlan 4095 succeed.
```

6.1.4.42.4 Querying the TX Rate Limit of a Specified PF (-t pf_tx_rate)

Function

The **-t pf_tx_rate** command is used to query the TX rate limit of a specified PF.

Format

hinicadm sdi_cfg -i <*devicename*> **-t pf_tx_rate**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the TX rate limit of a specified eth4 device.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_tx_rate
eth4 pf_tx_rate 100000000kbps,2560000kbits,100000000kbps,2560000kbits.
```

6.1.4.42.5 Setting the TX Rate Limit of a Specified PF (-t pf_tx_rate -l)

Function

The **-t pf_tx_rate -l** command is used to set the TX rate limit of a specified PF.

Format

hinicadm sdi_cfg -i <*devicename*> **-t pf_tx_rate -l** <*cir,cbs,pir,pbs*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**
cir,cbs,pir,pbs	Indicates the TX rate limit.	● The unit of cir and pir is kbit/s. ● The unit of cbs and pbs is kbit.

Usage Instruction

None

Example

Set the TX rate limit of a specified eth4 device.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_tx_rate -l
1000000,800000,1000000,800000
Set eth4 pf rate succeed.
```

6.1.4.42.6 Deleting the TX Rate Limit of a Specified PF (-t pf_tx_rate -c)

Function

The **-t pf_tx_rate -c** command is used to delete the TX rate limit of a specified PF.

Format

hinicadm sdi_cfg -i <*devicename*> **-t pf_rx_rate -c**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Delete the TX rate limit of a specified eth4 device.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_rx_rate -c
Clear eth4 pf rate limit succeed.
```

6.1.4.42.7 Querying the RX Rate Limit of a Specified PF (-t pf_rx_rate)

Function

The **-t pf_rx_rate** command is used to query the RX rate limit of a specified PF.

Format

hinicadm sdi_cfg -i < *devicename*> **-t pf_rx_rate**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Query the RX rate limit of a specified eth4 device.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_rx_rate
eth4 pf_rx_rate 100000000kbps,2560000kbits.
```


6.1.4.42.8 Setting the RX Rate Limit of a Specified PF (-t pf_rx_rate -l)

Function

The **-t pf_rx_rate -l** command is used to set the RX rate limit of a specified PF.

Format

hinicadm sdi_cfg -i <*devicename*> **-t pf_rx_rate -l** <*cir,cbs*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**
cir,cbs	Indicates the TX rate limit.	● The unit of cir is kbit/s. ● The unit of cbs is kbit.

Usage Instruction

None

Example

Set the RX rate limit of a specified eth4 device.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_rx_rate -l 1000000,800000
Set eth4 pf rate succeed.
```

6.1.4.42.9 Deleting the RX Rate Limit of a Specified PF (-t pf_rx_rate -c)

Function

The **-t pf_rx_rate -c** command is used to delete the RX rate limit of a specified PF.

Format

hinicadm sdi_cfg -i <*devicename*> **-t pf_rx_rate -c**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be set in the system.	Example: **hinic0** and **hinic1**

Usage Instruction

None

Example

Delete the RX rate limit of a specified eth4 device.

```
linux-xvUVkf:~ # hinicadm sdi_cfg -i eth4 -t pf_rx_rate -c
Clear eth4 pf rate limit succeed.
```

6.1.4.43 Querying the Real-Time Traffic Sent and Received by Each Port of a NIC Where a RoCE Device Is Located (roce_port_traffic)

Function

The **roce_port_traffic** command is used to query the real-time traffic sent and received by each port of a NIC where a RoCE device is located.

Format

hinicadm roce_port_traffic -i <*device*> -t <*exec_time*>

Parameters

Parameter	Description	Value
device	Indicates the valid RoCE device name.	Example: **hrn0_0** or **hrn0_bond_0**
exec_time	Indicates the execution time. Unit: second	-

Usage Instruction

None

Example

Run the perftest test command **ib_send_lat** to enable the RoCE device **hrn0_bond_0** to receive and send data.

```
[root@localhost ~]# service: ib_send_lat -d hrn0_bond_0 -F -a -x 1
[root@localhost ~]# client: ib_send_lat -d hrn0_bond_0 -F -a -x 1 127.0.0.1
```

Use hrn_bond_0 to show the real-time traffic information of all RoCE ports, view the traffic data of hrn0_bond_0.

```
[root@localhost ~]# hinicadm roce_port_traffic -i hrn0_bond_0 -t 2
| port tx(Mpps) tx(Gbps)  rx(Mpps) rx(Gbps)  | port tx(Mpps) tx(Gbps)  rx(Mpps)
rx(Gbps)  |
| 0    0.000     0.000000 0.000     0.000000 | 1    0.000     0.000000 0.000
0.000000 |
| 2    0.004     0.000768 0.004     0.001024 | 3    0.000     0.000000 0.000
0.000000 |

| port tx(Mpps) tx(Gbps)  rx(Mpps) rx(Gbps)  | port tx(Mpps) tx(Gbps)  rx(Mpps)
```



```
rx(Gbps)   |
| 0    0.000     0.000000 0.000     0.000000 | 1    0.000     0.000000 0.000
0.000000 |
| 2    0.004     0.003072 0.004     0.001024 | 3    0.000     0.000000 0.000
0.000000 |
```

6.1.4.44 Setting NIC Packet Data to Be Sent and Triggering NIC Packet Sending When AC Power Is Lost (reboot_notice)

Function

The **reboot_notice** command is used to set NIC packet data to be sent and trigger NIC packet sending when AC power is lost

Format

hinicadm reboot_notice -i <*devicename*> **-m** <*mode*> **-f** <*filename*>

hinicadm reboot_notice -i <*devicename*> **-s**

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 whose FEC mode is to be queried in the system.	Example: **hinic0** and **hinic1**
mode	Reads packet data to be sent from the NIC, or writes data to the NIC.	● **read**: obtains packet data to be sent from the NIC and saves the data to the file specified by *filename*. ● **write**: writes data to be sent to the NIC. The data is from the file specified by *filename*.
filename	Name of the file that stores the packet data.	-

Usage Instruction

- **-s**: triggers all ports of the NIC to send packet data.
- This command supports only NICs of the TaiShan 2280 V2 and supports only Linux OSs.

Example

Read the packet data to be sent on hinic0.

```
[root@localhost ~]# hinicadm reboot_notice -i hinic0 -m read -f ./read.file
File:read.file is created succeed
reboot_notice command succeed.
```

Write the packet data to be sent to hinic0.

```
[root@localhost ~]# hinicadm reboot_notice -i hinic0 -m write -f ./write.file
reboot_notice command succeed.
```


Trigger all ports of the network adapter to send packets.

```
[root@localhost ~]# hinicadm reboot_notice -i hinic0 -s
reboot_notice command succeed.
```

6.1.4.45 Querying and Setting the DCB Function (dcb)

Function

The **dcb** command is used to query or set the DCB function of a device.

Format

hinicadm dcb -i <*devicename*> -e <*mode*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried and set in the system.	For example, the value can be **enp6s0** or **enp7s0**.
mode	Enables or disables the DCB function.	● **1**: Enables the DCB function. ● **0**: Disables the DCB function.

Usage Instruction

None

Example

Check whether DCB is enabled for enp6s0.

```
[root@localhost ~]# hinicadm dcb -i enp6s0
enp6s0      dcb state: disable
```

Enable DCB for enp6s0.

```
[root@localhost ~]# hinicadm dcb -i enp6s0 -e 1
Set enp6s0 dcb to enable succeed.
```

6.1.4.46 Querying and Setting the PFC Function (pfc)

Function

The **pfc** command is used to query or set the PFC function of a device.

Format

hinicadm pfc -i <*devicename*> -e <*mode*> [-p <*priority*>]

hinicadm pfc -i <*devicename*> -p <*priority*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried and set in the system.	For example, the value can be **enp6s0** or **enp7s0**.
mode	Enables or disables the PFC function.	● **1**: Enables PFC. ● **0**: Disables PFC.
priority	Enables or disables PFC pointer for the 0 – 7 priority.	xxxxxxxx ● **1**: enables the pointer. ● **0**: disables the pointer.

Usage Instruction

After PFC is enabled for the first time after you run the **-e** command, you can run the **-p** command to set the PFC pointer priority.

Example

Check whether PFC is enabled for enp6s0.

```
[root@localhost ~]# hinicadm pfc -i enp6s0
enp6s0      pfc state: disable
pfcup:      1       1       1       1       1       1       1       1
num TC's:   4
```

Enable PFC for enp6s0.

```
[root@localhost ~]# hinicadm pfc -i enp6s0 -e 1
Set enp6s0 pfc to enable succeed
```

Enable the PFC for priority 2 and priority 3 of enp6s0.

```
[root@localhost ~]# hinicadm pfc -i enp6s0 -p 00110000
Set enp6s0 pfc to enable succeed.
Set enp6s0 pfc succeed
```

6.1.4.47 Setting the ETS Function (ets)

6.1.4.47.1 Querying and Setting the ETS Function (-e)

Function

The **ets -e** command is used to query or set the ETS function of a device.

Format

hinicadm ets -i <*devicename*> **-e** <*mode*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried and set in the system.	For example, the value can be **enp6s0** or **enp7s0**.
mode	Enables or disables the ETS function.	● **1**: enables ETS. ● 0: disables ETS.

Usage Instruction

None

Example

Check whether ETS is enabled for enp6s0.

```
[root@localhost ~]# hinicadm ets -i enp6s0
enp6s0      ets state: disable
up2tc:      3       3       3       3       3       2       1       0
pcnt:       100     0       0       0       0       0       0       0
strict:     0       0       0       0       0       0       0       0
```

Enable ETS for enp6s0.

```
[root@localhost ~]# hinicadm ets -i enp6s0 -e 1
Set enp6s0 ets succeed
```

6.1.4.47.2 Setting SP Scheduling (-t)

Function

The **ets -t** command is used to set SP scheduling of a device.

Format

hinicadm ets -i <*devicename*> -e <*mode*> -t <*tc*>

hinicadm ets -i <*devicename*> -t <*tc*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried and set in the system.	For example, the value can be **enp6s0** or **enp7s0**.
mode	Enables or disables the ETS function.	● **1**: enables ETS. ● 0: disables ETS.

Parameter	Description	Value
tc	Controls the scheduling relationship before the traffic reaches the Traffic Classifier (TC).	xxxxxxxx The **x** value ranges from 0 to 3.

Usage Instruction

- After ETS is enabled for the first time after you run the **-e** command, you can run the **-t** command to control the scheduling relationship before traffic reaches the TC.
- After the ETS function is disabled, the ETS function cannot be configured and the scheduling relationship before the traffic reaches the TC does not change. ETS needs to be manually reconfigured.

Example

Set the priority-2 and priority-3 ETS of enp6s0 to SP scheduling.

```
[root@localhost ~]# hinicadm ets -i enp6s0 -e 1 -t 00120000
Set enp6s0 ets succeed
```

6.1.4.47.3 Setting DWRR Scheduling (-t -p)

Function

The **ets -t -p -s** command is used to set Deficit Weighted Round Robin (DWRR) scheduling of a device.

Format

hinicadm ets -i <*devicename*> -e <*mode*> [-t <*tc*>] -p <*percent*>

hinicadm ets -i <*devicename*> [-t <*tc*>] -p <*percent*>

Parameters

Parameter	Description	Value
devicename	Indicates the name of the IN200 to be queried and set in the system.	For example, the value can be **enp6s0** or **enp7s0**.
mode	Enables or disables the ETS function.	- **1**: enables ETS. - 0: disables ETS.
tc	Controls the scheduling relationship before the traffic reaches the TC.	xxxxxxxx The **x** value ranges from 0 to 3.

Parameter	Description	Value
percent	Controls the scheduling usage percentage in different COSs.	x,x,x,x,x,x,x,x The **x** value ranges from 0 to 100.

Usage Instruction

- After ETS is enabled for the first time after you run the **-e** command, you can run the **-t** command to control the scheduling relationship before traffic reaches the TC.
- After ETS is enabled for the first time after you run the **-e** command, you can run the **-p** command to control the scheduling usage percentage in different COSs.
- After the ETS function is disabled, the ETS function cannot be configured and the scheduling relationship before the traffic reaches the TC does not change. ETS needs to be manually reconfigured.
- After the ETS function is disabled, the ETS function cannot be configured and the scheduling usage percentage in different COSs does not change. ETS needs to be manually reconfigured.

Example

Set the priority-2 and priority-3 ETS of enp6s0 to DWRR scheduling. The bandwidth of priority 2 accounts for 20%, and the bandwidth of priority 3 accounts for 80%.

```
[root@localhost ~]# hinicadm ets -i <device> -e 1 -t 00120000 -p 0,0,20,80,0,0,0,0
Set enp6s0 ets succeed
```

6.1.5 Upgrading hinicadm

Prerequisites

- The hinicadm installation package has been downloaded.

  For the package name and download method, see **2.1 Obtaining Software Packages**.
- The hinicadm installation package has been uploaded to the server OS.

Upgrading hinicadm on Linux

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Upgrade hinicadm.

- RPM Package:

  Run the **rpm -Uvh** *hinicadm-<version>-<release>.<arch>.rpm* command.

  Example:

```
linux-uueJIR:/Hi1822/tools/linux # rpm -Uvh hinicadm-1.2.8.0-1.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:hinicadm-1.2.8.0-1               ################################# [ 50%]
Cleaning up / removing...
   2:hinicadm-1.2.7.0-1               ################################# [100%]
```


- DEB Package:

  Run the **dpkg -i hinicadm-xxxx-xxxx.xxx.deb** commond.

  Example:

```
root@ubuntu1804:/home/ubuntu # dpkg -i hinicadm-1.6.1.2-1.aarch64.deb
(Reading database ... 69641 files and directories currently installed.)
Preparing to unpack hinicadm-1.6.1.2-1.aarch64.deb ...
Unpacking hinicadm (1.6.1.2) over (1.6.1.1) ...
Uninstall hinicadm tool  successfully.
NOTE: There are some user files under the directory /opt/hinic/.
      If you don't need them, you can delete them manually.
Setting up hinicadm (1.6.1.2) ...
Installing... Please wait for a moment.
Install hinicadm tool successfully.
```

----End

Upgrading hinicadm on Microsoft Windows

Perform operations in **6.1.1 Installing hinicadm** and the new driver package automatically replaces the existing one.

Follow-up Procedure

After the upgrade is complete, you can run the **hinicadm -v** command to query the current version of the tool to ensure that the upgrade is successful.

6.1.6 Uninstalling hinicadm

Uninstalling hinicadm from Linux

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Go to the hinicadm installation directory.

Step 3 Uninstall hinicadm.

- RPM Package:

  Run the **rpm -e hinicadm** command.

```
linux-uueJIR:/Hi1822/tools/linux # rpm -e hinicadm
NOTE: There are some user files under the directory /opt/hinic/.
      If you don't need them, you can delete them manually.
```

- DEB Package:

  Run the **dpkg -r hinicadm** commond.

```
root@ubuntu1804:/home/ubuntu# dpkg -r hinicadm
(Reading database ... 69641 files and directories currently installed.)
Removing hinicadm (1.6.1.2) ...
Uninstall hinicadm tool  successfully.
NOTE: There are some user files under the directory /opt/hinic/.
      If you don't need them, you can delete them manually.
```

----End

Uninstalling hinicadm from Microsoft Windows

Step 1 Log in to the server OS.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Click  , choose **Control Panel** > **Programs and Features**.

Step 3 Right-click the **HinicadmTool** program name and choose **Uninstall/Change** from the shortcut menu.

----End

6.2 Standard Management Tools

The ethtool, ifconfig, ip link, vconfig, and dcbtool tools are standard NIC management tools delivered with the Linux system. They can be used to configure and manage the IN200 because the IN200 is a standard PCIe card.

Table 6-2 lists the common ethtool commands supported by the IN200 and their functions. For details about how to use the commands, see the help information provided on the command-line interface (CLI).

Table 6-2 Common ethtool commands

Command	Function
ethtool ethx	Displays the basic settings of a specified Ethernet port.
ethtool -s ethx	Sets the auto-negotiation mode and network port rate of a specified Ethernet port. **NOTE** If DAC cables are used to connect IN200 ports and auto-negotiation is disabled on the peer device, run the **ethtool -s ethx autoneg off** command to disable auto-negotiation on the corresponding IN200 port.
ethtool -a ethx	Displays the pause status of a specified Ethernet port.
ethtool -A ethx	Sets the pause status of a specified Ethernet port.
ethtool -c ethx	Displays the interrupt aggregation parameter of a specified Ethernet port.
ethtool -C ethx	Configures the interrupt aggregation parameter of a specified Ethernet port.
ethtool -g ethx	Displays queue depth information of a specified Ethernet port.
ethtool -G ethx	Configures queue depth information of a specified Ethernet port.
ethtool -k ethx	Displays the offload and acceleration features supported by the NIC.
ethtool -K ethx	Configures the offload and acceleration features supported by the NIC.
ethtool -i ethx	Displays related information of a specified Ethernet port.
ethtool -t ethx	Runs a loopback test.
ethtool -S ethx	Displays hardware and drive statistics.
ethtool -n ethx	Displays the Receive Side Scaling (RSS) rule currently supported by a specified network port.

Command	Function
ethtool -N ethx	Configures the RSS rule.
ethtool -x ethx	Query the indirect table.
ethtool -X ethx	Configures the indirect table.
ethtool -P ethx	Displays the fixed MAC address.
ethtool -l ethx	Queries the number of queues.
ethtool -L ethx	Configures the number of queues.
Some ethtool standard commands are not applicable to an IN200, and their functions are substituted by hinicadm. For example: ● ethtool -e: provides the eeprom burning function, which is used in NIC firmware update. For an IN200, ethtool -e is substituted by the **updatefw** command of hinicadm. ● ethtool -d: provides the chip register dump function, which is used to collect chip location information. For an IN200, ethtool -d is substituted by the **hilink_dump** command of hinicadm.	

Table 6-3 lists the common ifconfig commands supported by the IN200 and their functions. For details about how to use the commands, see the help information provided on the CLI.

Table 6-3 Common ifconfig commands

Command	Function
ifconfig ethx hw	Configures the NIC address.
ifconfig ethx mtu	Configures the network port MTU.
ifconfig ethx [-]allmulti	Enables or disables the full multicast feature.
ifconfig ethx [-]promisc	Enables or disables the promiscuous mode.

Table 6-4 lists the common ip link commands supported by the IN200 and their functions. For details about how to use the commands, see the help information provided on the CLI.

Table 6-4 Common ip link commands

Command	Function
ip link set dev ethx vf <num> <mac addr>	Sets the MAC address of the virtual network port.
ip link set devethx vf <num> vlan <vlan_id> qos <vlan-qos>	Sets the VLAN ID and QoS priority of the virtual network port.

Command	Function
ip link set dev ethx vf <num> <rate txrate>	Sets the rate limit parameter of the virtual network port.
ip link set dev ethx vf <num> state <staus>	Sets the link status of the virtual network port.

Table 6-5 lists the common vconfig commands supported by the IN200 and their functions. For details about how to use the commands, see the help information provided on the CLI.

Table 6-5 Common vconfig commands

Command	Function
vconfig add ethx <vlan_id>	Adds a VLAN sub-device.
vconfig rem <vlan-name>	Deletes a VLAN sub-device.

Table 6-6 lists the common dcbtool commands supported by the IN200 and their functions. For details about how to use the commands, see the help information provided on the CLI.

Table 6-6 Common dcbtool commands

Command	Function
dcbtool sc ethx dcb on/off	Enables or disables Data Center Bridgin (DCB).
dcbtool gc ethx dcb	Queries the DCB status.
dcbtool gc ethx pg	Queries the priority group settings.
dcbtool sc ethx pg pgid:xxx	Specifies the priority bandwidth group to which a specified priority is mapped.
dcbtool sc ethx pg pgpct:x,x,x,x,x,x,x,x	Specifies the bandwidth ratio of each priority bandwidth group (the sum is 100%).
dcbtool sc ethx pg strict:xxxxxxxx	Sets a specified priority to **strict**.
dcbtool sc ethx pg up2tc:xxxxxxxx	Specifies the TC group to which each priority is mapped.
dcbtool gc ethx pfc	Queries the PFC status of each priority.
dcbtool sc ethx pfc pfcup:xxxxxxxx	Specifies the PFC enablement status of each priority.

7 Appendix

7.1 Logging In to the Real-Time Server Desktop

To log in to the server OS, you can use the server multi-port connector to connect to the KVM as the login terminal or use the remote console provided by the server iBMC for remote login.

The following is the procedure for using the remote console provided by the server iBMC to log in to the real-time server desktop.

Step 1 Configure the login environment.

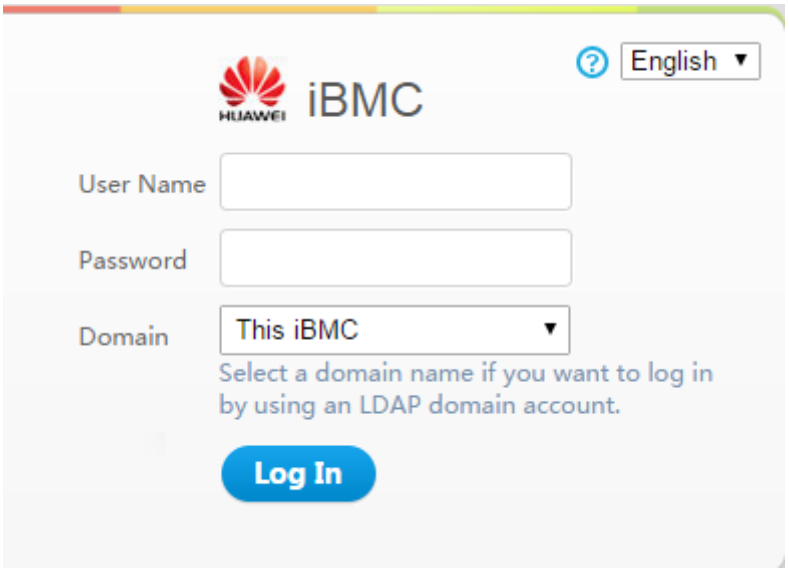
1. Connect the PC to the iBMC management network port using a network cable.
2. Set an IP address for the PC, and ensure that the IP address is on the same network segment as that of the iBMC management network port.

   For example, set the IP address to **192.168.2.10** and the subnet mask to **255.255.255.0**.

Step 2 Log in to the iBMC WebUI.

1. Open a browser, enter **https://*IP address of the iBMC management network port*** in the address box, and press **Enter**.

Figure 7-1 iBMC login page



2. On the iBMC login page, perform the following operations:
   - Select the language to be used.
   - Enter the user name.
   - Enter the password.
   - Set **Domain** to **This iBMC**.
   - Click **Log In**.

     The iBMC WebUI home page is displayed.

Step 3 Go to the Remote Virtual Console.

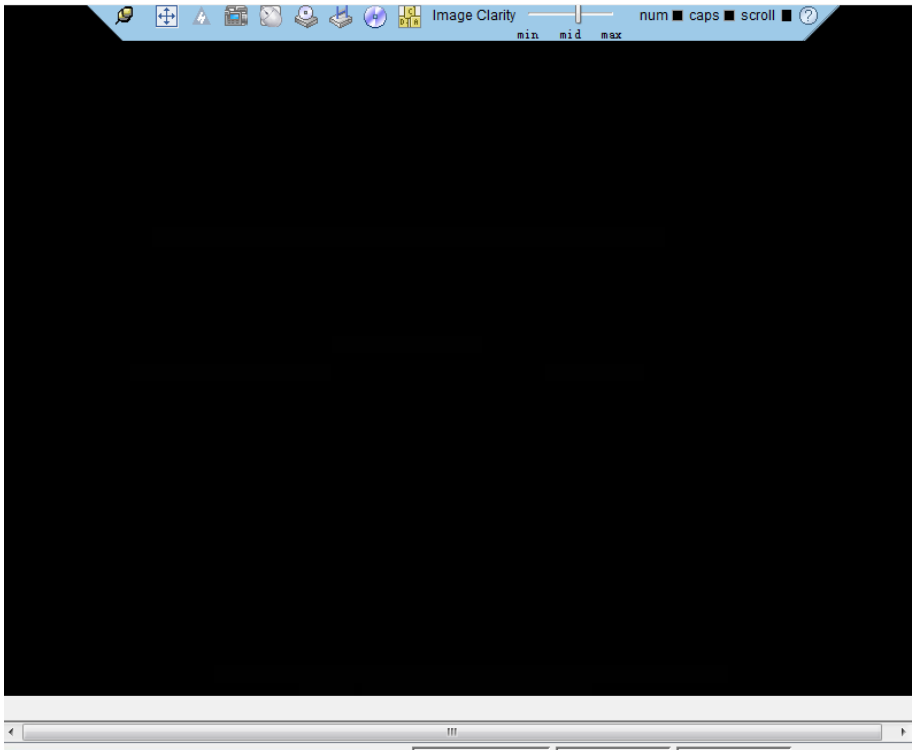
1. In the navigation tree, choose **Remote Control**.

   The **Remote Control** page is displayed.
2. Click the Remote Virtual Console button.
   - If Java is integrated, click **Java Integrated Remote Console**.
   - If HTML5 is integrated, click **HTML5 Integrated Remote Console**.

   The following uses the Java integrated Remote Virtual Console as an example.

   The **Remote Virtual Console** screen is displayed, as shown in **Figure 7-2**.

Figure 7-2 Remote Virtual Console screen



----End

7.2 Restarting the Server

You can use the iBMC to power off and then power on the server for the firmware upgrade to take effect.

Restarting the Server on the iBMC WebUI

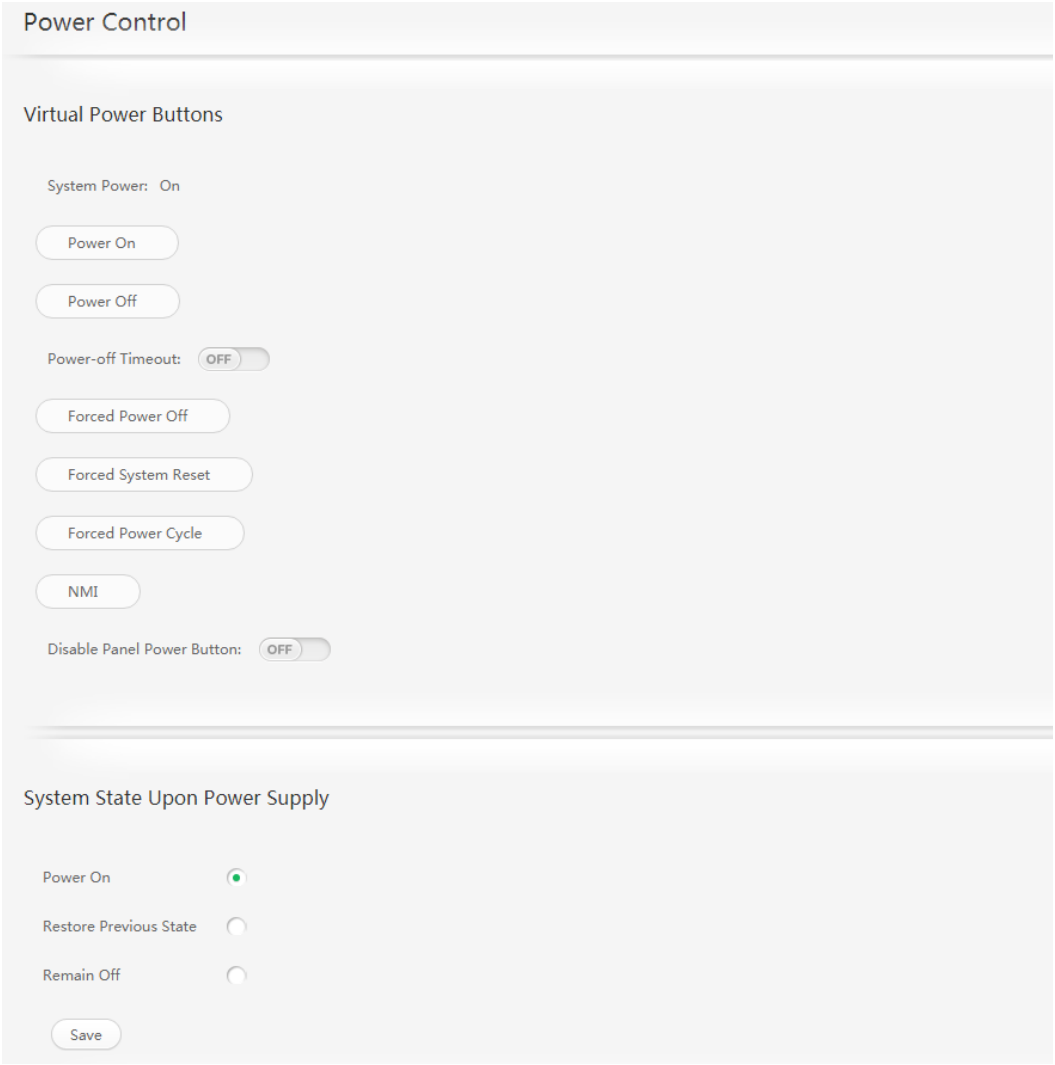
Step 1 Log in to the iBMC WebUI.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 In the navigation tree, choose **Power** > **Power Control**.

The **Power Control** page is displayed, as shown in **Figure 7-3**.

Figure 7-3 Power Control page



Step 3 Select **Forced Power Cycle**.

Step 4 Confirm the operation in the operation confirmation dialog box displayed.

----End

Restarting the Server Through the Remote Virtual Console

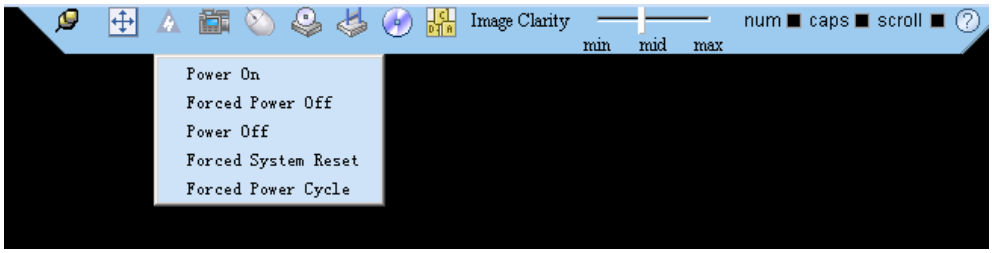
Step 1 Log in to the real-time server desktop using the Remote Virtual Console.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 On the toolbar of the Remote Virtual Console, click .

The power control menu is displayed, as shown in **Figure 7-4**.

Figure 7-4 Power control menu



Step 3 Choose **Forced Power Cycle**.

Step 4 Confirm the operation in the operation confirmation dialog box displayed.

----End

7.3 Transferring a File Using the Virtual Directory

Before installing and upgrading the driver, upgrading the firmware, and installing management tool, you need to transfer the corresponding file to the OS of the server.

The following describes how to transfer a file using the virtual directory function of the Virtual Remote Console.

Step 1 Log in to the real-time server desktop using the Remote Virtual Console.

For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 On the toolbar of the Remote Virtual Console, click  .

The virtual device list is displayed, as shown in **Figure 7-5**.

Figure 7-5 Virtual device menu

Step 3 Select **Directory** and click **Browse**.

Step 4 Select the local path where the file is.

Step 5 Click **Connect**.

Step 6 Copy the file to the server OS.

The device type and location of a virtual directory varies depending on the operating system type. For details, see the official website of each respective OS vendor.

----End

7.4 Common BIOS Configuration

The following uses the BIOS of the 2288H V5 as an example.

NOTE

Set the server boot mode to the **UEFI** mode before performing this operation.

Step 1 Log in to the real-time server desktop using the Remote Virtual Console.

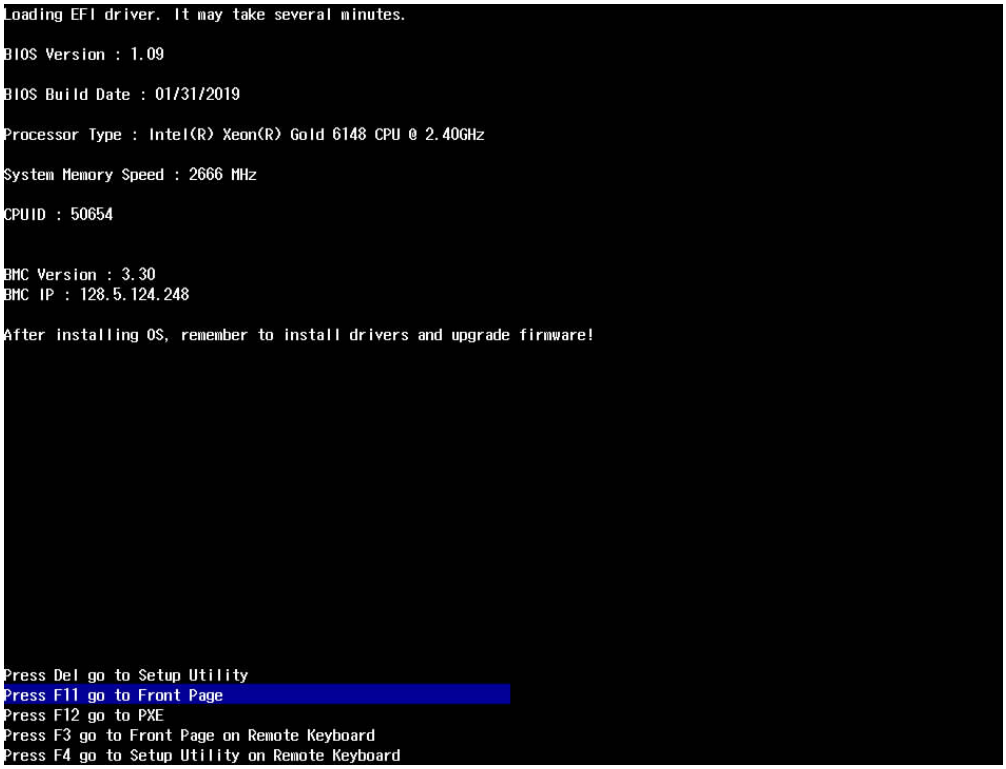
For details, see **7.1 Logging In to the Real-Time Server Desktop**.

Step 2 Restart the server.

Step 3 The BIOS configuration screen is displayed.

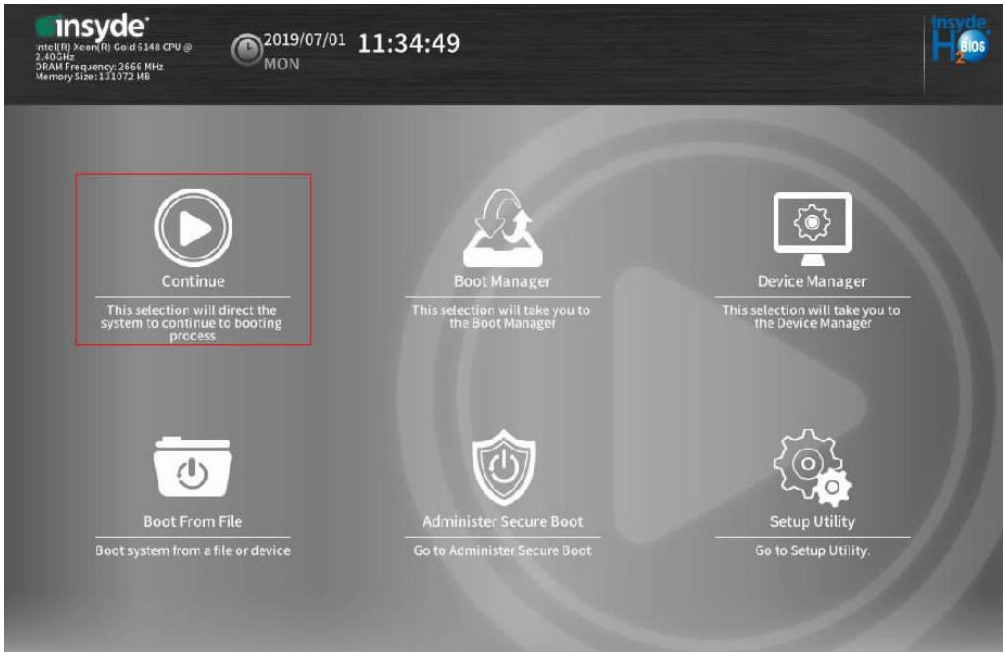
- If the BIOS version is V363 or earlier, the message shown in **Figure 7-6** is displayed.

Figure 7-6 Startup information (1)



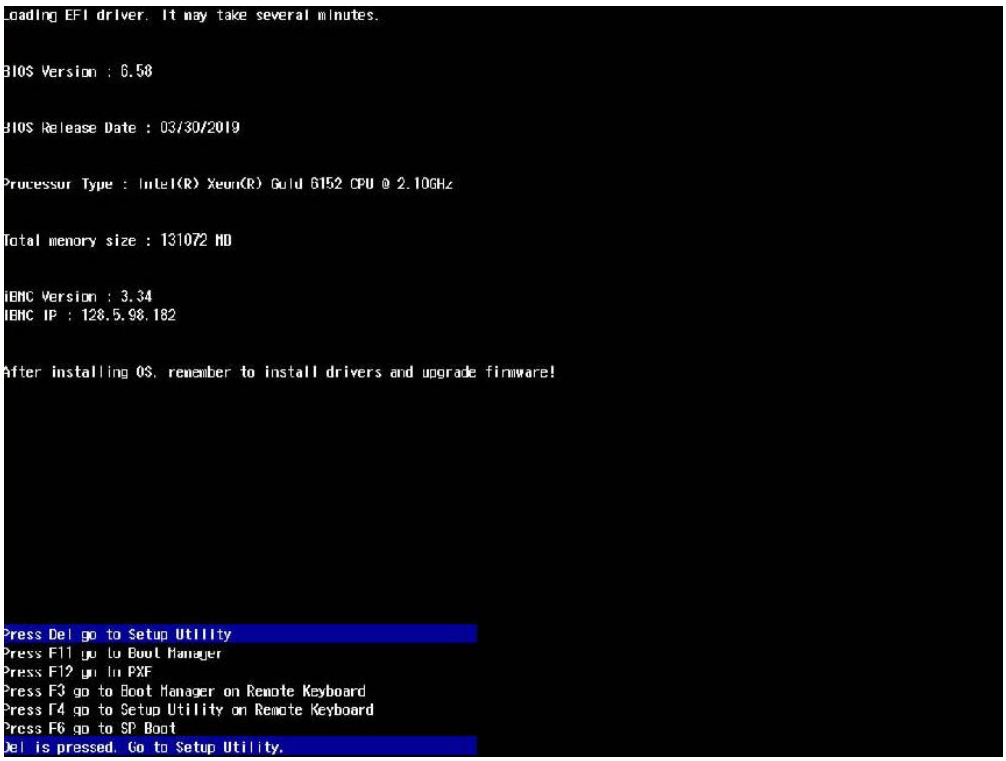
Press **F11** to access the BIOS screen, as shown in **Figure 7-7**. Go to **Step 4**.

Figure 7-7 Front Page screen (1)



- If the BIOS version is V658 or later, the message shown in **Figure 7-8** is displayed.

Figure 7-8 Startup information (2)



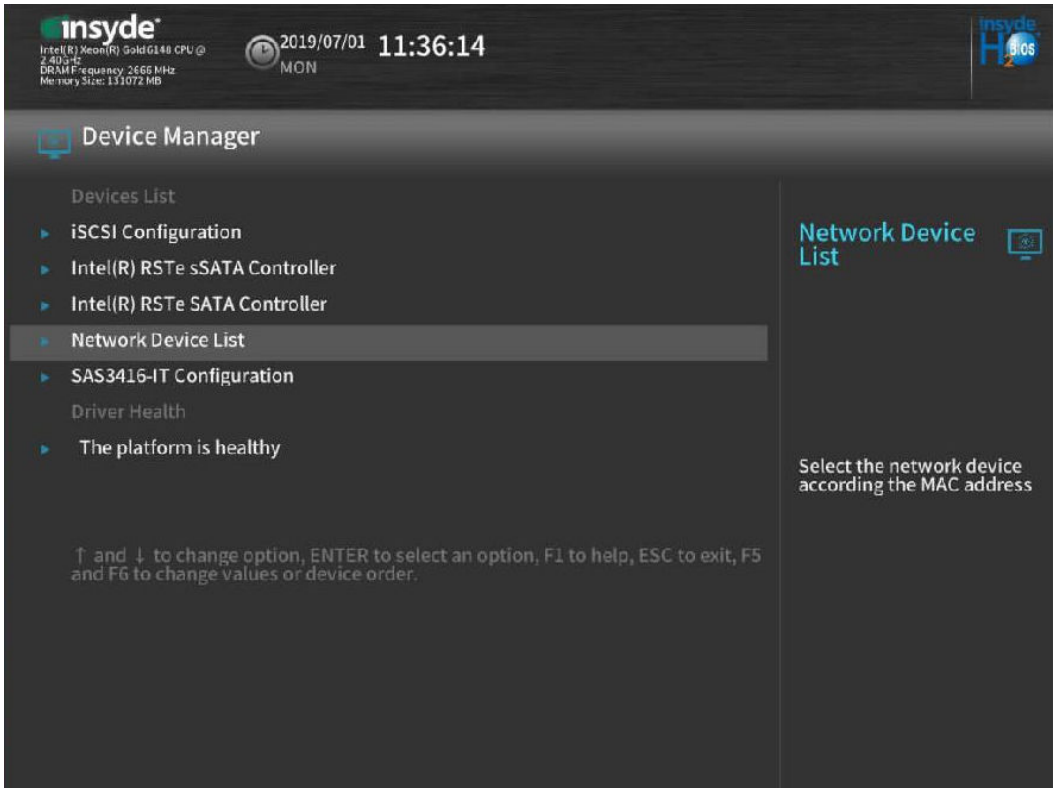
Press **Delete** to access the BIOS screen, as shown in **Figure 7-9**. Go to **Step 4**.

Figure 7-9 Front Page screen (2)

Step 4 Use arrow keys to select **Device Manager** and press **Enter**.

The **Device Manager** screen is displayed, as shown in **Figure 7-10**.

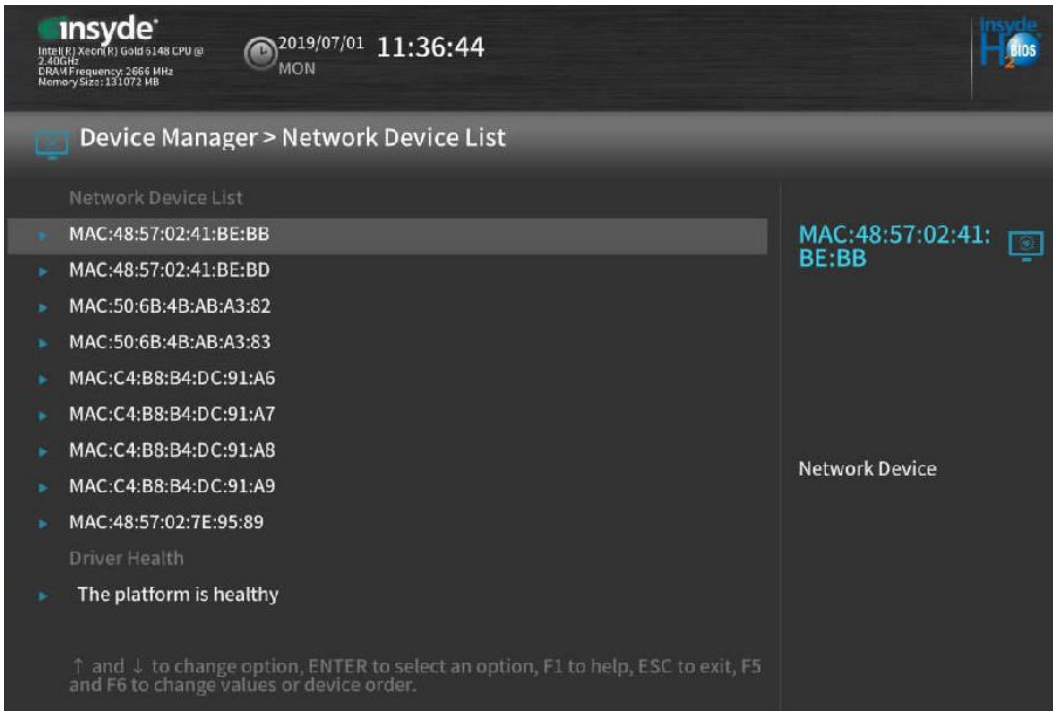
Figure 7-10 Device Manager screen



Step 5 Use arrow keys to select **Network Device List** and press **Enter**.

The NIC list is displayed, as shown in **Figure 7-11**.

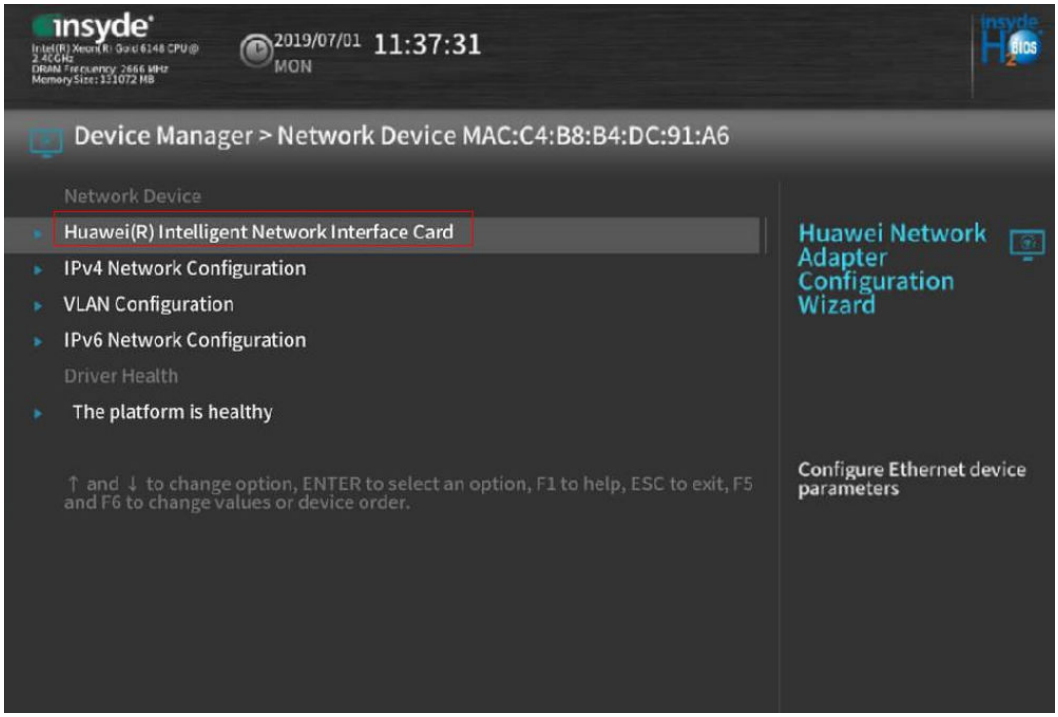
Figure 7-11 Network Device List screen



Step 6 Select the MAC address of the target NIC, and press **Enter**.

The NIC configuration screen is displayed, as shown in **Figure 7-12**.

Figure 7-12 NIC configuration screen



Step 7 Select **Huawei Network Adapter Configuration** and press **Enter**.

The **Huawei Network Adapter Configuration** screen is displayed, as shown in **Figure 7-13**.

Figure 7-13 Huawei Network Adapter Configuration screen

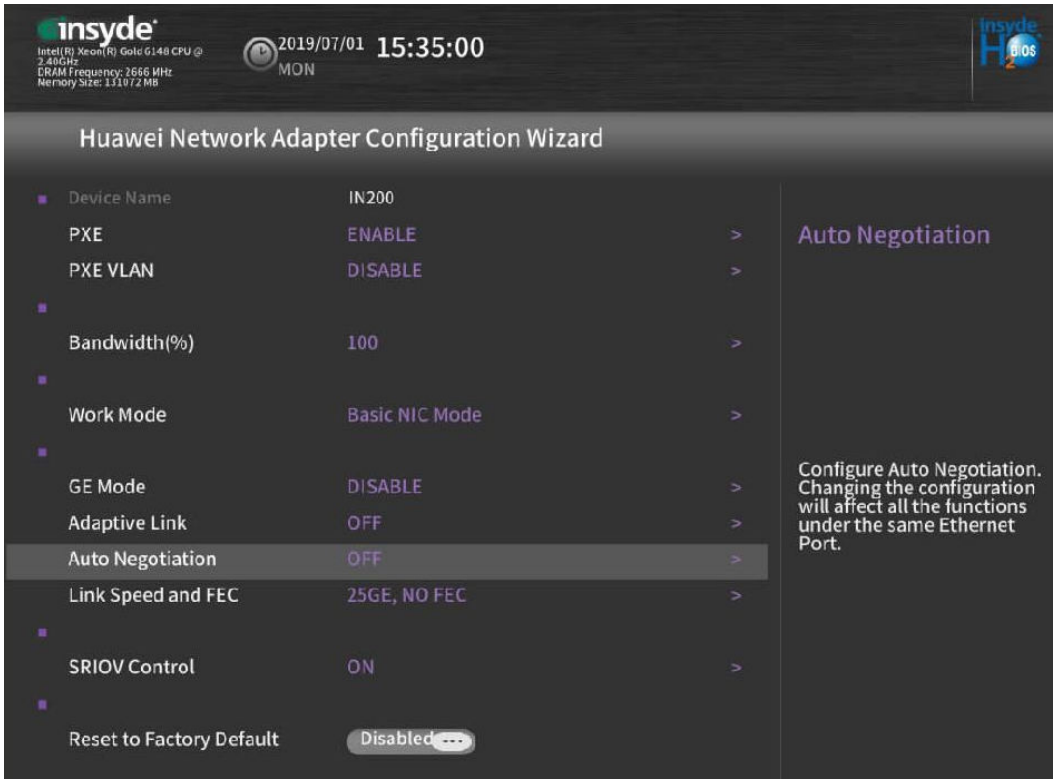


Table 7-1 describes the parameters on the **Huawei Network Adapter Configuration** screen.

Table 7-1 Parameter description

Parameter	Description	Value Range
PXE	Indicates whether to enable the PXE function.	● ENABLE ● DISABLE
PXE VLAN	Indicates whether to enable the VLAN function.	● ENABLE ● DISABLE
Bandwidth(%)	Indicates the rate limit of the network port.	1 to 100
Work Mode	Indicates the NIC working mode.	● Basic NIC Mode ● OVS Offload Mode

Parameter	Description	Value Range
GE Mode	Indicates the connection mode of a network port. After the GE mode is configured, the port supports GE connections. **NOTE** The mode setting of port 0 correlates with port 1, and the setting of port 2 correlates with port 3. For example, if mode is changed from **enabled** to **disabled** for port 0 and takes effect, mode will be automatically changed to **disabled** for port 1.	● Disable ● Enable
Adaptive Link	Indicates whether to enable the network port adaptive mode. **NOTE** When **Adaptive Link** is set to **OFF**, **Auto Negotiation** can be configured.	● ON ● OFF
Auto Negotiation	When GE Mode is set to Disable, the auto-negotiation function of the port is enabled. **NOTE** When **Auto Negotitaion** is set to **OFF**, **Link Speed and FEC** can be configured.	● ON ● OFF
Link Speed and FEC	Indicates the network port speed and the FEC mode.	● If the NIC supports 25GE and 10GE, the network port speed and FEC mode options are as follows: – 25GE, NO FEC – 25GE, BASE FEC – 25GE, RS FEC – 10GE, NO FEC – 10GE, BASE FEC ● If the NIC supports 100GE and 40GE, the network port speed and FEC mode options are as follows: – 100GE, RS FEC – 100GE, NO FEC – 40GE, NO FEC – 40GE, BASE FEC
SRIOV Control	Specifies whether to enable the SRIOV function for the port. The default value is ON.	● ON ● OFF

Parameter	Description	Value Range
Reset to Factory Default	Specifies whether to restore the port to factory settings.	● Disable ● Enable

Step 8 Set parameters as required.

Step 9 Save the settings and exit.

----End

A FAQ

A.1 An Exception Occurs During Driver Installation or Uninstallation

On the Windows OS, if the server is restarted or powered off during the driver installation or uninstallation, the residual entries may exist in the registry. As a result, driver installation or uninstallation may fail. You can use **PsExec.exe** to delete the residual registry entries.

Use the PsExec.exe tool to delete the residual registry entries

Step 1 Download the **PsExec.exe** tool.

Step 2 Open the registry and find the name of the residual registry entries. The paths are **HKEY_LOCAL_MACHINE\DRIVERS\DriverDatabase\DriverPackages\hinic.inf_amd64xxxx** .

Step 3 Open the cmd window and run the following command to delete the residual registry entries.

PsExec.exe -d -i -s reg delete HKEY_LOCAL_MACHINE\DRIVERS\DriverDatabase\DriverPackages\hinic.inf_amd64xxxx /f

----End

B Acronyms and Abbreviations

A	
AC	Alternating Current
B	
BIOS	Basic Input and Output System
D	
DCB	Data Center Bridging
DPDK	Data Plane Development Kit
E	
ETS	Enhanced Transmission Selection
F	
FEC	Forward Error Correction
I	
I^2C	Inter-integrated Circuit
L	
LRO	large receive offload
LACP	Link Aggregation Control Protocol
M	
MAC	Media Access Control
MCTP	Management Component Transport Protocol

N	
NC-SI	Network Controller Sideband Interface
P	
PCIe	Peripheral Component Interconnect Express
PFC	Priority Flow Control
PXE	Preboot eXecution Enviroment
R	
RoCE	Remote Direct Memory Access over Converged Ethernet
RSS	Receive Side Scaling
S	
SMBus	System Management Bus
SR-IOV	Single Root I/O Virtualization
T	
TC	Traffic Classifier
TSO	TCP Segmentation Offload
U	
UEFI	Unified Extensible Firmware Interface
V	
VLAN	Virtual Local Area Network
VXLAN	Virtual eXensible Local Area Network

