

# DHCP Relay Commands

This chapter contains the following sections:


- ip dhcp relay enable (Global), on page 2
- ip dhcp relay enable (Interface), on page 3
- ip dhcp relay address (Global), on page 4
- show ip dhcp relay, on page 5
- ip dhcp information option, on page 6
- ip dhcp information option numeric-token-format, on page 7
- ip dhcp information option circuit-id, on page 8
- ip dhcp information option remote-id, on page 13
- show ip dhcp information option tokens, on page 17
- show ip dhcp information option , on page 20

# ip dhcp relay enable (Global)

Use the **ip dhcp relay enable** Global Configuration mode command to enable the DHCP relay feature on the device. Use the **no** form of this command to disable the DHCP relay feature.

### Syntax

**ip dhcp relay enable**

**no ip dhcp relay enable**

### Default Configuration

DHCP relay feature is disabled.

### Command Mode

Global Configuration mode

### Example

The following example enables the DHCP relay feature on the device.

```
switchxxxxxx(config)# ip dhcp relay enable
```

# ip dhcp relay enable (Interface)

Use the **ip dhcp relay enable** Interface Configuration mode command to enable the DHCP relay feature on an interface. Use the **no** form of this command to disable the DHCP relay agent feature on an interface.

### Syntax

**ip dhcp relay enable**

**no ip dhcp relay enable**

### Default Configuration

Disabled

### Command Mode

Interface Configuration mode

### User Guidelines

The operational status of DHCP Relay on an interface is active if one of the following conditions exist:

- DHCP Relay is globally enabled, and there is an IP address defined on the interface.

Or

- DHCP Relay is globally enabled, there is no IP address defined on the interface, the interface is a VLAN, and option 82 is enabled.

### Example

The following example enables DHCP Relay on VLAN 21.

```
switchxxxxxx(config)# interface vlan 21
switchxxxxxx(config-if)# ip dhcp relay enable
```

# ip dhcp relay address (Global)

Use the **ip dhcp relay address** Global Configuration mode command to define the DHCP servers available for the DHCP relay. Use the **no** form of this command to remove the server from the list.

### Syntax

**ip dhcp relay address** *ip-address*

**no ip dhcp relay address** [*ip-address*]

### Parameters

- ***ip-address***—Specifies the DHCP server IP address. Up to 8 servers can be defined.

### Default Configuration

No server is defined.

### Command Mode

Global Configuration mode

### User Guidelines

Use the **ip dhcp relay address** *command to define a global DHCP Server IP address. To define a few DHCP Servers, use the command a few times.*

To remove a DHCP Server, use the **no** form of the command with the *ip-address* argument.

The **no** form of the command without the *ip-address* argument deletes all global defined DHCP servers.

### Example

The following example defines the DHCP server on the device.

```
switchxxxxxx(config)# ip dhcp relay address 176.16.1.1
```

# show ip dhcp relay

Use the **show ip dhcp relay** EXEC mode command to display the DHCP relay information.

### Syntax

**show ip dhcp relay**

### Command Mode

User EXEC mode

### Examples

Option 82 is disabled:

```
switchxxxxxx# show ip dhcp relay
DHCP relay is globally disabled
Option 82 is disabled
Maximum number of supported VLANs without IP Address: 0
Number of DHCP Relays enabled on VLANs without IP Address: 4
DHCP relay is enabled on Ports: gi1/0/1,po1-2
 Active:
 Inactive: gi1/0/1, po1-4
DHCP relay is enabled on VLANs: 1, 2, 4, 5
 Active:
 Inactive: 1, 2, 4, 5
Global Servers: 1.1.1.1 , 2.2.2.2
```

Option 82 is enabled:

```
switchxxxxxx# show ip dhcp relay
DHCP relay is globally enabled
Option 82 is enabled
Maximum number of supported VLANs without IP Address is 4
Number of DHCP Relays enabled on VLANs without IP Address: 2
DHCP relay is enabled on Ports: gi1/0/1,po1-2
 Active: gi1/0/1
 Inactive: po1-2
DHCP relay is enabled on VLANs: 1, 2, 4, 5
 Active: 1, 2, 4, 5
 Inactive:
Global Servers: 1.1.1.1 , 2.2.2.2
```

# ip dhcp information option

Use the **ip dhcp information option** Global Configuration mode command to enable DHCP option-82 data insertion. Use the **no** form of this command to disable DHCP option-82 data insertion.

### Syntax

**ip dhcp information option**

**no ip dhcp information option**

### Default Configuration

DHCP option-82 data insertion is disabled.

### Command Mode

Global Configuration mode

### User Guidelines

DHCP option 82 would be enabled only if DHCP snooping or DHCP relay are enabled.

### Example

```
switchxxxxxx(config)# ip dhcp information option
```

# ip dhcp information option numeric-token-format

Use the **ip dhcp information option numeric-token-format** Global Configuration mode command to define the format of numeric tokens included in the Circuit-ID and Remote-ID sub-options payload template. Use the **no** form of this command to return to default format.

### Syntax

**ip dhcp information option numeric-token-format {hex|ascii}**

**no ip dhcp information option numeric-token-format**

### Parameters

- **hex** - Hexadecimal (Numeric value) format will be used in packet for numeric token included in the Circuit-ID and Remote-ID payload template.
- **ascii** - ASCII format will be used in packet for numeric token included in the Circuit-ID and Remote-ID payload template. If this option is selected, each individual digit in a numeric token will be represented by its value in the ASCII table.

### Default Configuration

The default format used is the hexadecimal/numeric format

### Command Mode

Global Configuration mode

### User Guidelines

Use this command to configure the format of numeric token included in the Circuit-ID or Remote-ID sub-options payload templates commands: ip dhcp information option circuit-id and ip dhcp information option remote-id.

The following are the numeric tokens affected by this command:

- $int-ifindex$
- $int-portid$
- $switch-moduleid$
- $vlan-id$

### Example

The following example configures the device to use the ASCII format for insertion of numeric-tokens:

```
switchxxxxxx(config)# ip dhcp information option numeric-token-format ascii
```

# ip dhcp information option circuit-id

Use the **ip dhcp information option circuit-id** Global Configuration mode command to configure the template of DHCP option 82 Circuit-ID sub-option payload. Use the **no** form of this command to return to default template.

### Syntax

**ip dhcp information option circuit-id** *text*

**no ip dhcp information option circuit-id**

### Parameters

- ***text*** - Concatenation of free text and one or more tokens in the format of $tokenname$ (length 1-160).

### Default Configuration

The default Circuit-ID payload template is *$vlan-id$$switch-moduleid$$int-portid$*

### Command Mode

Global Configuration mode

### User Guidelines

Use this command to configure the template of the option 82 Circuit-ID sub-option payload, inserted by device. The payload section of the Circuit-ID sub option includes all bytes of sub option besides the first 4 bytes of the sub option which values are set by device as follows:

Circuit ID sub option Type (value = 1)

- Sub option total length (not including 1st byte and total length byte)

  Circuit ID type (value= 1); Note: if default sub option template is used - value of this field equals 0

  Sub option payload length

The *text* field is a concatenation of free text and one or more tokens in the format of *$tokenname$*. Tokens must be entered in the exact format specified (see table below) or it will not be recognized as a token

The *text* can begin or end with free text or a token. Tokens can be concatenated sequentially or separated by free text. If the free text includes a space character - the *text* parameter must be placed between quotation (e.g. "*text1 text2*").

The Circuit-ID payload template must include at least 1 token related to an interface parameter (token beginning with *$int-xxx$*). In addition - if the string does not include a VLAN related token - user will be asked to confirm setting.

The total length of the *text* field in the command cannot exceed 160 bytes. The byte count includes all bytes of the text parameter - including all free text and Tokens as written in the *text* field.

The combined length of the Circuit-ID payload Remote-ID payload cannot exceed 247 bytes. The payload byte count takes into account the count of free-text chars (1 byte each) and a predefined length reserved for each token (see in table below).

The following table details supported tokens, the device parameter they represent and the reserved and actual byte count per each token:

| Token Name | Description | Reserved Length | Actual Length |
|---|---|---|---|
| **$int-ifindex$** | Source interface ifIndex value | 4 bytes | Hex Format - 2 bytes<br>ASCII format - 4 bytes |
| **$int-portid$** | Source interface sequential number on the specific module (in stack). For LAG source interfaces - it is the LAG ID | 2 bytes | Hex Format - 1 bytes<br>ASCII format - 2 bytes |
| **$int-name$** | The full name of the source interface, as used in CLI commands | 32 bytes | The actual number of bytes needed for ASCII representation of the interface full name |
| **$int-abrvname$** | The abbreviated name of the source interface, as used in CLI commands | 8 bytes | The actual number of bytes needed for ASCII representation of the interface full name |
| **$int-desc-16$** | The description configured by user on source interface. If description is more than 16 bytes - only the 1st 16 bytes are used<br>If a description is not configured, the abbreviated interface name is used | 16 bytes | The actual number of bytes needed for ASCII representation of the interface description (max 16 bytes) |
| **$int-desc-32$** | The description configured by user on source interface. If description is more than 32 bytes - only the 1st 32 bytes are used<br>If a description is not configured, the abbreviated interface name is used | 32 bytes | The actual number of bytes needed for ASCII representation of the interface description (max 32 bytes) |

| Token Name | Description | Reserved Length | Actual Length |
|---|---|---|---|
| **$int-desc-64$** | The description configured by user on source interface.<br>If a description is not configured, the abbreviated interface name is used | 64 bytes | The actual number of bytes needed for ASCII representation of the interface description |
| **$int-mac$** | MAC address of the source interface (Hex value with no delimiter) | 6 bytes | 6 bytes |
| **$switch-mac$** | MAC address of the switch relaying/forwarding DHCP packet (Hex value with no delimiter) | 6 bytes | 6 bytes |
| **$switch-hostname-16$** | The hostname of the switch relaying/forwarding DHCP packet.<br>If the hostname is more than 16 bytes - only the 1st 16 bytes are used | 16 bytes | The actual number of bytes needed for ASCII representation of the hostname (max 16 bytes) |
| **$switch-hostname-32$** | The hostname of the switch relaying/forwarding DHCP packet.<br>If the hostname is more than 32 bytes - only the 1st 32 bytes are used | 32 bytes | The actual number of bytes needed for ASCII representation of the hostname (max 32 bytes) |
| **$switch-hostname-58$** | The hostname of the switch relaying/forwarding DHCP packet. | 58 bytes | The actual number of bytes needed for ASCII representation of the hostname |
| $switch-moduleid$ | The unit ID of the source interface upon which the DHCP client request was received | 2 bytes | Hex Format - 1 bytes<br>ASCII format - 2 bytes |
| $vlan-id$ | The Source VLAN ID (1-4094) | 4 bytes | Hex Format - 2 bytes<br>ASCII format - 4 bytes |

| Token Name | Description | Reserved Length | Actual Length |
|---|---|---|---|
| **$vlan-name-16$** | The VLAN name assigned by user to the VLAN. If the name is more than 16 bytes - only the 1st 16 bytes are used<br>If a name is not configure for the VLAN, the value is taken from the relevant VLAN ifDescr MIB field | 16 bytes | The actual number of bytes needed for ASCII representation of the VLAN name (max 16) |
| **$vlan-name-32$** | The VLAN name assigned by user to the VLAN.<br>If a name is not configure for the VLAN, the value is taken from the relevant VLAN ifDescr MIB field | 32 bytes | The actual number of bytes needed for ASCII representation of the VLAN name (max 32) |

**Notes:**

- Source Interface or VLAN int table refers to the Interface or VLAN on which the DHCP client packet (to which the option 82 is added) was received on.
- Reserved (Byte) Length - The maximum length the token may “consume” in the packet. This value is used for calculation of the 247 byte limit (for all sub options payload combined). The reserved length does not change if numeric tokens are filled in as Hexadecimal or ASCII values.
- Actual (Byte) Length - The actual number of bytes the token will “consume” in packet itself. The actual byte length may change (for relevant tokens) if token is filled in as exadecimal or ASCII values.

### Example

The following example configures the Circuit-ID payload template to a concatenation of free text and tokens representing interface name and VLAN name (up to 16 chars):

```
switchxxxxxx(config)# ip dhcp information option circuit-id aaa$int-name$bbb$vlan-name-16$ccc
```

The following example configures the Circuit-ID payload template to - where text parameter does not include a token related to an interface:

```
switchxxxxxx(config)# ip dhcp information option circuit-id aaa
Illegal Circuit-ID payload: Cicuit-ID must include at least 1 interface related Token
```

The following example configures the Circuit-ID payload template to use a concatenation of free text and tokens - where template does not include a token related to a VLAN:

```
switchxxxxxx(config)# ip dhcp information option circuit-id aaa$int-name$bbb
Circuit-ID payload does not include a token reflecting DHCP client source VLAN. Continue?
y/n[n] y
```

The following example configures the Circuit-ID payload template to use a concatenation of free text and tokens - resulting in a combined Circuit-ID and Remote-ID reserved payloads which exceed 247 bytes:

```
switchxxxxxx(config)# ip dhcp information option circuit-id
aaa$vlan-name-32$bbb$int-desc-64$ccc$switch-hostname-58$ddd
Illegal Circuit-ID payload: Circuit-ID and Remote-ID payload reserved byte count exceeds
247 bytes
```

# ip dhcp information option remote-id

Use the **ip dhcp information option remote-id** Global Configuration mode command to configure the template of DHCP option 82 Remote-ID sub-option payload. Use the **no** form of this command to return to default template.

### Syntax

**ip dhcp information option remote-id** *text*

**no ip dhcp information option remote-id**

### Parameters

- ***text*** - concatenation of free text and one or more tokens in the format of $tokenname$ (length 1-160).

### Default Configuration

The default Remote-ID payload template is *$switch-mac$*

### Command Mode

Global Configuration mode

### User Guidelines

Use this command to configure the template of the option 82 Remote-ID sub-option payload, inserted by device. The payload section of the Remote-ID sub option includes all bytes of sub option besides the first 4 bytes of the sub option which values are set by device as follows:

Remote-ID sub option Type (value = 2)

- Sub option total length (not including 1st byte and total length byte) Remote-ID type (value= 1); Note: if default sub option template is used - value of this field equals 0. Sub option payload length

The *text* field is a concatenation of free text and one or more tokens in the format of *$tokenname$*. Tokens must be entered in the exact format specified (see table below) or it will not be recognized as a token

The *text* can begin or end with free text or a token. Tokens can be concatenated sequentially or separated by free text. If the free text includes a space character - the *text* parameter must be placed between quotation (e.g. "*text1 text2*").

The Remote-ID payload template may include 1 token, multiple tokens or no tokens at all.

The total length of the *text* field in the command cannot exceed 160 bytes. The byte count includes all bytes of the text parameter - including all free text and Tokens as written in the *text* field.

The combined length of the Circuit-ID payload Remote-ID payload cannot exceed 247 bytes. The payload byte count takes into account the count of free-text chars (1 byte each) and a predefined length reserved for each token (see in table below).

The following table details supported tokens, the device parameter they represent and the reserved and actual byte count per each token:

| Token Name | Description | Reserved Length | Actual Length |
|---|---|---|---|
| **$int-ifindex$** | Source interface ifIndex value | 4 bytes | Hex Format - 2 bytes<br>ASCII format - 4 bytes |
| **$int-portid$** | Source interface sequential number on the specific module (in stack). For LAG source interfaces - it is the LAG ID | 2 bytes | Hex Format - 1 bytes<br>ASCII format - 2 bytes |
| **$int-name$** | The full name of the source interface, as used in CLI commands | 32 bytes | The actual number of bytes needed for ASCII representation of the interface full name |
| **$int-abrvname$** | The abbreviated name of the source interface, as used in CLI commands | 8 bytes | The actual number of bytes needed for ASCII representation of the interface full name |
| **$int-desc-16$** | The description configured by user on source interface. If description is more than 16 bytes - only the 1st 16 bytes are used<br>If a description is not configured, the abbreviated interface name is used | 16 bytes | The actual number of bytes needed for ASCII representation of the interface description (max 16 bytes) |
| **$int-desc-32$** | The description configured by user on source interface. If description is more than 32 bytes - only the 1st 32 bytes are used<br>If a description is not configured, the abbreviated interface name is used | 32 bytes | The actual number of bytes needed for ASCII representation of the interface description (max 32 bytes) |
| **$int-desc-64$** | The description configured by user on source interface.<br>If a description is not configured, the abbreviated interface name is used | 64 bytes | The actual number of bytes needed for ASCII representation of the interface description |

| Token Name | Description | Reserved Length | Actual Length |
|---|---|---|---|
| **$int-mac$** | MAC address of the source interface (Hex value with no delimiter) | 6 bytes | 6 bytes |
| **$switch-mac$** | MAC address of the switch relaying/forwarding DHCP packet (Hex value with no delimiter) | 6 bytes | 6 bytes |
| **$switch-hostname-16$** | The hostname of the switch relaying/forwarding DHCP packet.<br>If the hostname is more than 16 bytes - only the 1st 16 bytes are used | 16 bytes | The actual number of bytes needed for ASCII representation of the hostname (max 16 bytes) |
| **$switch-hostname-32$** | The hostname of the switch relaying/forwarding DHCP packet.<br>If the hostname is more than 32 bytes - only the 1st 32 bytes are used | 32 bytes | The actual number of bytes needed for ASCII representation of the hostname (max 32 bytes) |
| **$switch-hostname-58$** | The hostname of the switch relaying/forwarding DHCP packet. | 58 bytes | The actual number of bytes needed for ASCII representation of the hostname |
| **$switch-moduleid$** | The unit ID of the source interface upon which the DHCP client request was received | 2 bytes | Hex Format - 1 bytes<br>ASCII format - 2 bytes |
| **$vlan-id$** | The Source VLAN ID (1-4094) | 4 bytes | Hex Format - 2 bytes<br>ASCII format - 4 bytes |
| **$vlan-name-16$** | The VLAN name assigned by user to the VLAN. If the name is more than 16 bytes - only the 1st 16 bytes are used<br>If a name is not configure for the VLAN, the value is taken from the relevant VLAN ifDescr MIB field | 16 bytes | The actual number of bytes needed for ASCII representation of the VLAN name (max 16) |

| Token Name | Description | Reserved Length | Actual Length |
|---|---|---|---|
| **$vlan-name-32$** | The VLAN name assigned by user to the VLAN.<br>If a name is not configure for the VLAN, the value is taken from the relevant VLAN ifDescr MIB field | 32 bytes | The actual number of bytes needed for ASCII representation of the VLAN name (max 32) |

**Notes:**

- Source Interface or VLAN int table refers to the Interface or VLAN on which the DHCP client packet (to which the option 82 is added) was received on.
- Reserved (Byte) Length - The maximum length the token may “consume” in the packet. This value is used for calculation of the 247 byte limit (for all sub options payload combined). The reserved length does not change if numeric tokens are filled in as Hexadecimal or ASCII values.
- Actual (Byte) Length - The actual number of bytes the token will “consume” in packet itself. The actual byte length may change (for relevant tokens) if token is filled in as hexadecimal or ASCII values.

### Example

The following example configures the device to use a Remote-ID which is a concatenation of free text and the full device hostname:

```
switchxxxxxx(config)# ip dhcp information option remote-id aaa$switch-hostname-58$bbb
```

# show ip dhcp information option tokens

Use the **show ip dhcp information option tokens** User EXEC mode command to display tokens which can be used when configuring the payloads of Circuit-ID and Remote-ID sub options (DHCP option 82):

### Syntax

**show ip dhcp information option tokens [brief]**

### Parameters

- **Brief** - Displays the name of the tokens without token information.

### Default Configuration

Full token information is displayed

### Command Mode

User EXEC mode

### User Guidelines

Use this command to display the tokens that can be used as part of the text parameter in commands ip dhcp information option circuit-id or ip dhcp information option remote-id. The tokens represent various system info, which the user can include in either of the sub-options' payloads. This allows automatic update of value based on current system information and also based on relevant interfaces.

The tokens have meaningful and pre-determined names based on the information they represent. A $ symbol is placed before and after the Token name ($token-name$).

Tokens can be divided, in general, into 3 groups: Tokens which represent interface level information (format of $int-xxx$); Tokens which represent switch level information (format of $switch-xxx$), and tokens which represent VLAN related information (format of $vlan-xxx$).

### Example

The following example displays all supported tokens and all the information related to each token:

```
switchxxxxxx# show ip dhcp information option tokens
Interface level Tokens - relates to the interface upon which the DHCP client packet was
received:
Token Name: $int-ifindex$
Token value: ifIndex of the interface
Token format: Hex (default) or ASCII
Token reserved length: 4 bytes.
Token actual payload length: 2(HEX)/4(ASCII) bytes.
Token Name: $int-portid$
Token value: interface number relative to the specific unit (standalone or stacking unit)
Token format: Hex (default) or ASCII
Token reserved length: 2 bytes
Token actual payload length: 1(HEX)/2(ASCII) bytes
Token Name: $int-name$
Token value: The interface full name based as used in CLI
Token format: ASCII
Token reserved length: 32 bytes
```

```
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $int-abrvname$
Token value: The interface abbreviated name as used in CLI
Token format: ASCII
Token reserved length: 8 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $int-desc-16$
Token value: (up to) The first 16 bytes of the description user configured for the interface
Token format: ASCII
Token reserved length: 16 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $int-desc-32$
Token value: (up to) The first 32 bytes of the description user configured for the interface
Token format: ASCII
Token reserved length: 32 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $int-desc-64$
Token value: The full description user configured for the interface (even if more than 32
bytes)
Token format: ASCII
Token reserved length: 64 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $int-mac$
Token value: The MAC address of the physical interface
Token format: HEX
Token reserved length: 6 bytes
Token actual payload length:6 bytes
Device level Tokens - relates to switch level information:
Token Name: $switch-mac$
Token value: Device base MAC address
Token format: HEX
Token reserved length: 6 bytes
Token actual payload length:6 bytes
Token Name: $switch-hostname-16$
Token value: (Up to) The first 16 bytes of the hostname of the device
Token format: ASCII
Token reserved length: 16 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $switch-hostname-32$
Token value: (Up to) The first 32 bytes of the hostname of the device
Token format: ASCII
Token reserved length: 32 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $switch-hostname-58$
Token value: Device full hostname (even if more than 32 bytes)
Token format: ASCII
Token reserved length: 58 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
Token Name: $switch-moduleid$
Token value: The unit ID of the unit within the stack
Token format: Hex (default) or ASCII
Token reserved length: 2 bytes
Token actual payload length: 1(HEX)/2(ASCII) bytes
VLAN level Tokens - relates to the VLAN upon which the DHCP client packet was received:
Token Name: $vlan-id$
Token value: VLAN ID (1-4094)
Token format: Hex (default) or ASCII
Token reserved length: 4 bytes
Token actual payload length: 2(HEX)/4(ASCII) bytes
Token Name: $vlan-name-16$
Token value: (Up to) The first 16 bytes of the VLAN name
Token format: ASCII
Token reserved length: 16 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
```

```
Token Name: $vlan-name-32$
Token value: The full VLAN name (even if more than 16 bytes)
Token format: ASCII
Token reserved length: 32 bytes
Token actual payload length: Actual number of bytes (ASCII) inserted to sub-option
```

The following example displays only the names of the supported Tokens:

```
switchxxxxxx# show ip dhcp information option tokens brief
Interface level Tokens:
$int-ifindex$
$int-portid$
$int-name$
$int-abrvname$
$int-desc-16$
$int-desc-32$
$int-desc-64$
$int-mac$
Device level Tokens:
$switch-mac$
$switch-hostname-16$
$switch-hostname-32$
$switch-hostname-58$
$switch-moduleid$
VLAN level Tokens:
$vlan-id$
$vlan-name-16$
$vlan-name-32$
```

# show ip dhcp information option

The **show ip dhcp information option** User EXEC mode command displays the DHCP Option 82 and sub option configuration.

### Syntax

**show ip dhcp information option [{interface** *interface-id***} {vlan** *vlan***}]**

### Parameters

- **interface** *interface-id* - Specifies an interface ID. The interface ID can be one of the following types: Ethernet port or Port-channel. Use this parameter together with the **vlan** parameter to display actual option 82 payload for a DHCP client message received on the specified interface and VLAN
- **vlan** *vlan*- Specifies a VLAN ID Use this parameter together with the **interface** parameter to display actual option 82 payload for a DHCP client message received on the specified interface and VLAN

### Default Configuration

If no parameter is entered the general settings of option 82 will be displayed.

### Command Mode

User EXEC mode

### Example

The following example displays global information for DHCP option 82, including sub options, when user did not change the settings of Option 82 sub-options.

```
switchxxxxxx# show ip dhcp information option
Relay agent Information option is Enabled
Numeric Token format: hex
Circuit-id payload template: (default)
Remote-id payload template: (default)
Total sub Options reserved payload: 14/247 bytes
```

The following example displays global information for DHCP option 82, including sub options, where user modified both Circuit-ID and Remote-ID sub-options.

```
switchxxxxxx# show ip dhcp information option
Relay agent Information option is Enabled
Numeric Token format: hex
Circuit-id payload template: aaa$int-name$bbb$vlan-name$ccc
Remote-id payload template: aaa$switch-hostname-58$bbb
Total sub Options reserved payload: 143/247 bytes
```

The following example displays specific interface and VLAN information for DHCP option 82, where user modified both Circuit-ID and Remote-ID sub-options.

```
switchxxxxxx# show ip dhcp information option interface te1/0/1 vlan 2
Relay agent Information option is Enabled
Numeric Token format: hex
Circuit-id payload template: aaa$int-name$bbb$vlan-name$ccc
Remote-id payload template: aaa$switch-hostname-58$bbb
Total sub Options reserved payload: 143/247 bytes
```

```
Interface te1/0/1 vlan 2:
Circuit-id header content: 0131012f
Circuit-id payload content: 61616154656e67696761626974457461865726e657431
2f302f31626262241502d564c414e636363
Circuit-id payload textual resolution: aaaTengigabitEthernet1/0/1bbbAP-VLANccc //removed
support 31-Jul-18//
Circuit-id Total Length: 43
Remote-id header content: 0211010f
Remote-id payload content: 616161466c6f6f723453777697463686262 62
Remote-id payload textual resolution: aaaFloor4Switchbbbb //removed support 31-Jul-18//
Remote-id Total Length: 22
```

The following example displays specific interface and VLAN information for DHCP option 82, including sub options, when user modified only the Circuit-ID sub-option.

```
switchxxxxxx# show ip dhcp information option interface te1/0/10 vlan 13
Relay agent Information option is Enabled
Numeric Token format: hex
Circuit-id payload template: $int-portid$aaa$vlan-id$zzz
Remote-id payload template: (default)
Total sub Options reserved payload: 18/247 bytes
Interface te1/0/10 vlan 13:
Circuit-id header content: 010b012f
Circuit-id payload content: 0a616161000d7a7a7a
Circuit-id payload textual resolution: 10aaa13zzz //removed support 31-Jul-18//
Circuit-id Total Length: 13
Remote-id header content: 02080006
Remote-id payload content: 000000112233
Remote-id payload textual resolution: 000000112233 //removed support 31-Jul-18//
Remote-id Total Length: 10
```

The following example displays specific interface and VLAN information for DHCP option 82, when user set Numeric Token format to ASCII and configured Circuit-ID sub-option.

```
switchxxxxxx# show ip dhcp information option interface te1/0/10 vlan 13
Relay agent Information option is Enabled
Numeric Token format: ascii
Circuit-id payload template: $int-portid$aaa$vlan-id$zzz
Remote-id payload template: (default)
Total sub Options reserved payload: 18/247 bytes
Interface te1/0/10 vlan 13:
Circuit-id header content: 010e012f
Circuit-id payload content: 31306161613030313337a7a7a
Circuit-id payload textual resolution: 10aaa13zzz ////removed support 31-Jul-18//
Circuit-id Total Length: 16
Remote-id header content: 0211000f
Remote-id payload content: 000000112233
Remote-id payload textual resolution: 000000112233 //removed support 31-Jul-18//
Remote-id Total Length: 10
```

The following example requested display is for specific interface and VLAN information for DHCP option 82, where $vlan-name-32$ is one of the tokens - but the specific VLAN was not created on device.

```
switchxxxxxx# show ip dhcp information option interface te1/0/1 vlan 2
Relay agent Information option is Enabled
Numeric Token format: hex
Circuit-id payload template: aaa$int-name$bbb$vlan-name-32$ccc
Remote-id payload template: aaa$switch-hostname-58$bbb
Total sub Options reserved payload: 137/247 bytes
Interface te1/0/1 vlan 2:
Error - Cannot calculate Circuit-ID info - sub-option contains VLAN related Token which
does not exist on device.
```