Configuring the CFM over EFP Interface with Cross Connect Feature

Ethernet Connectivity Fault Management (CFM) is an end-to-end per-service-instance Ethernet layer OAM protocol that includes proactive connectivity monitoring, fault verification, and fault isolation. Currently, Ethernet CFM supports Up facing and Down facing Maintenance Endpoints (MEPs). For information on Ethernet Connectivity Fault Management, see http://www.cisco.com/en/US/docs/ios/12_2sr/12_2sra/feature/guide/srethcfm.html

The CFM over EFP Interface with xconnect feature allows you to:

- Forward continuity check messages (CCM) towards the core over cross connect pseudowires.
- Receive CFM messages from the core.
- Forward CFM messages to the access side (after Continuity Check Database [CCDB] based on maintenance point [MP] filtering rules).

Restrictions and Usage Guidelines

When configuring CFM over EFP Interface with cross connect, follow these restrictions and usage guidelines:

- Only a single down-facing MEP is allowed on the L2VFI.
- As the number of PEs in a VPLS instance scale up, the number of CFM CC messages processed increases. Accordingly, the configuration of the down-facing MEP on L2VFI for large fully meshed PW topologies should be considered for only premium valued networks.
- In the design of CFM domains, the maintenance level of an Down-facing MEP on the L2VFI interface must be lower than the level from the AC.
- Up MEP, Down MEP, and MIPs are supported.
- Offloaded CFM Up MEP sessions are not supported when using E type interface in an EFP using **encapsulation** command.

Configuring CFM over EFP with xconnect for the Cisco ASR 903 Series Router

The following sections describe how to configure CFM on EFP interfaces on the Cisco ASR 903 Series Router:

- Configuring CFM over EFP Interface with Cross Connect—Basic Configuration, page -26
- Configuring CFM over EFP Interface with Cross Connect—Single Tag VLAN Cross Connect, page -28

- Configuring CFM over EFP Interface with Cross Connect—Double Tag VLAN Cross Connect, page -31
- Configuring CFM over EFP Interface with Cross Connect—Selective QinQ Cross Connect, page -33
- Configuring CFM over EFP Interface with Cross Connect—Port-Based Cross Connect Tunnel, page -34
- Configuring CFM over EFP Interface with Cross Connect—Port Channel-Based Cross Connect Tunnel, page -36

Configuring CFM over EFP Interface with Cross Connect—Basic Configuration

This section describes how to configure CFM over EFP Interface with cross connect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** [*pw-class-name*]
4. **encapsulation mpls**
5. **exit**
6. **interface gigabitethernet** *slot/port* **or interface tengigabitethernet** *slot/port*
7. **service instance** *id* **{Ethernet [***service-name***}**
8. **encapsulation dot1q** *vlan_id*
9. **xconnect** *peer-ip-address vc-id* **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** *pw-class pw-class-name* **}[***pw-class pw-class-name***] [sequencing {transmit | receive | both}]**
10. **cfm mep domain** *domain-name* **[up | down] mpid** *mpid-value* **[cos** *cos-value***]**
11. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**pseudowire-class** [pw-class-name] **Example:** `Router(config)# pseudowire-class vlan-xconnect`	Specifies the name of a Layer 2 pseudowire class and enter pseudowire class configuration mode.

	Command	Purpose
Step 4	**encapsulation mpls** **Example:** `Router(config-if)# encapsulation mpls`	Specifies that Multiprotocol Label Switching (MPLS) is used as the data encapsulation method for tunneling Layer 2 traffic over the pseudowire.
Step 5	**exit** **Example:** `Router(config-if-srv)# exit`	Exits the pseudowire class configuration mode.
Step 6	**interface gigabitethernet** slot/port **or interface tengigabitethernet** slot/port **Example:** `Router(config-if-srv)# interface Gi2/0/2`	Specifies the Gigabit Ethernet or the Ten Gigabit Ethernet interface to configure.
Step 7	**service instance** id **ethernet** [service-name] **Example:** `Router(config-if-srv)# service instance 101 ethernet`	Creates a service instance (an instantiation of an EVC) on an interface and sets the device into the config-if-srv submode.
Step 8	**encapsulation dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **second-dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **Example:** `Router(config-if-srv)# encapsulation dot1q 100 second dot1q 200`	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q, QinQ, or untagged frames on an interface for the appropriate service instance.
Step 9	**xconnect** peer-ip-address vc-id **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** pw-class pw-class-name **}[**pw-class pw-class-name**] [sequencing {transmit | receive | both}]** **Example:** `Router(config-if-srv)# xconnect 10.0.3.201 123 pw-class vlan-xconnect`	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.

	Command	Purpose
Step 10	**cfm mep domain domain-name [up | down] mpid mpid-value [cos cos-value]** **Example:** `Router(config-if-srv)# cfm mep down mpid 100 domain Core`	Configures a maintenance endpoint (MEP) for a domain.
Step 11	**exit** **Example:** `Router(config-if-srv)# exit`	Exits the interface configuration mode.

Examples

This example shows how to configure CFM over EVC using cross connect.

```
PE3#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
PE3(config)#ethernet cfm domain L6 level 6
PE3(config-ecfm)# service s256 evc 256
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end

PE3(config)#int ten 2/0/0
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
PE3#
PE3(config)#ethernet cfm domain L2 level 2
PE3(config-ecfm)# service s256 evc 256 direction down
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end
PE3#
PE3(config)#int ten 2/0/0
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
PE3#
```

Configuring CFM over EFP Interface with Cross Connect—Single Tag VLAN Cross Connect

This section describes how to configure CFM over EFP Interface with Single Tag VLAN cross connect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *slot/subslot/port* **or interface tengigabitethernet** *slot/port*

4. **service instance** *id* **{Ethernet [***service-name***}**
5. **encapsulation dot1q** {**any** | *vlan-id*[*vlan-id*[*vlan-id*]} **second-dot1q** {any |*vlan-id*[*vlan-id*[*vlan-id*]]}
6. **rewrite ingress tag** {**push {dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**]
7. **xconnect** *peer-ip***-address** *vc-id* **{encapsulation {l2tpv3 [manual] | mpls [manual]} | pw-class** *pw-class-name* **}[***pw-class pw-class-name***] [sequencing {transmit | receive | both}]**
8. **cfm mep domain** *domain-name* **[up | down] mpid** *mpid-value* **[cos** *cos-value***]**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters the global configuration mode.
Step 3	**interface gigabitethernet** *slot*/sub-slot*/port* **Example:** `Router(config)# interface Gi2/0/2`	Specifies the Gigabit Ethernet interface to configure, where: *slot/subslot/port*—Specifies the location of the interface.
Step 4	**service instance** id **Ethernet** [ser-vice-name] **Example:** `Router(config-if)# service instance 101 ethernet`	Creates a service instance (an instantiation of an EVC) on an interface and sets the device into the config-if-srv submode.
Step 5	**encapsulation dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **sec-ond-dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **Example:** `Router(config-if-srv)# encapsulation dot1q 100 second dot1q 100`	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q, QinQ, or untagged frames on an interface for the appropriate service instance.

	Command	Purpose
Step 6	**rewrite ingress tag** {**push {dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**] **Example:** `Router(config-if-srv)# rewrite dot1q single symmetric`	Specifies the tag manipulation that is to be performed on the frame ingress to the service instance.
Step 7	**xconnect** peer-ip-address vc-id **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** pw-class pw-class-name **}[**pw-class pw-class-name**] [sequencing {transmit | receive | both}]** **Example:** `Router(config)# xconnect 10.0.3.201 123 pw-class vlan-xconnect`	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.
Step 8	**cfm mep domain** domain-name **[up | down] mpid** mpid-value **[cos** cos-value**]** **Example:** `Router# cfm mep up mpid 100 domain Core`	Configures a maintenance endpoint (MEP) for a domain.

Examples

This example shows how to configure CFM over EFP Interface with Single Tag VLAN cross connect:

```
PE3(config)#ethernet cfm domain L2 level 2
PE3(config-ecfm)# service s256 evc 256 direction down
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end
PE3#
PE3(config)#int ten 2/0/0
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
PE3#
```

Configuring CFM over EFP Interface with Cross Connect—Double Tag VLAN Cross Connect

This section describes how to configure CFM over EFP Interface with Double Tag VLAN cross connect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *slot/subslot/port*
4. **service instance** *id* {**Ethernet** [*service-name*}
5. **encapsulation dot1q {any | *vlan-id*[*vlan-id*[*vlan-id*]} second-dot1q {any |*vlan-id*[*vlan-id*[*vlan-id*]]}**
6. **rewrite ingress tag** {**push {dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**]
7. **xconnect** *peer-ip-address vc-id* **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** *pw-class pw-class-name* **}[***pw-class pw-class-name***] [sequencing {transmit | receive | both}]**
8. **cfm mep domain** *domain-name* **[up | down] mpid** *mpid-value* **[cos** *cos-value***]**
9. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot*/sub-slot/*port* **Example:** `Router(config)# interface Gi2/0/2`	Specifies the Gigabit Ethernet interface to configure, where: *slot/subslot/port*—Specifies the location of the interface.
Step 4	**service instance** id **Ethernet** [service-name] **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and sets the device into the config-if-srv submode.

	Command	Purpose
Step 5	**encapsulation dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **second-dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **Example:** Router(config-if-srv)# encapsulation dot1q 100 second-dot1q 200	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q, QinQ, or untagged frames on an interface for the appropriate service instance.
Step 6	**rewrite ingress tag** {**push {dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**] **Example:** Router(config-if-srv)# rewrite dot1q double symmetric	Specifies the tag manipulation that is to be performed on the frame ingress to the service instance.
Step 7	**xconnect** peer-ip-address vc-id **{encapsulation {l2tpv3 [manual] | mpls [manual]}** | pw-class pw-class-name **}[**pw-class pw-class-name**] [sequencing {transmit | receive | both}]** **Example:** Router(config)# xconnect 1.1.1.1 100 pw-class vlan-xconnect	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.
Step 8	**cfm mep domain** domain-name **[up | down] mpid** mpid-value **[cos** cos-value**]** **Example:** Router# cfm mep down mpid 100 domain Core	Configures a maintenance endpoint (MEP) for a domain.

Examples

This example shows how to configure CFM over EFP Interface with Double Tag VLAN cross connect:

```
PE3(config)#ethernet cfm domain L2 level 2
PE3(config-ecfm)# service s256 evc 256 direction down
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end
PE3#
PE3(config)#int ten 2/0/0
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256 second-dot1q 257
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
```

```
PE3#
```

Configuring CFM over EFP Interface with Cross Connect—Selective QinQ Cross Connect

This section describes how to configure CFM over EFP Interface with Selective QinQ cross connect.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *slot/subslot/port*
4. **exit**
5. **service instance** *id* {**Ethernet** [*service-name*}
6. **encapsulation dot1q {any | *vlan-id*[*vlan-id*[*vlan-id*]} second-dot1q {any |*vlan-id*[*vlan-id*[*vlan-id*]]}**
7. **xconnect** *peer-ip-address vc-id* **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** *pw-class pw-class-name* **}[***pw-class pw-class-name***] [sequencing {transmit | receive | both}]**
8. **cfm mep domain** *domain-name* **[up | down] mpid** mpid-value **[cos** *cos-value***]**
9. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot*/sub-slot/*port* **Example:** `Router(config)# interface Gi2/0/2`	Specifies the Gigabit Ethernet interface to configure, where: *slot/subslot/port*—Specifies the location of the interface.
Step 4	**service instance** id **Ethernet** [ser-vice-name] **Example:** `Router(config-if)# service instance 101 ethernet`	Creates a service instance (an instance of an EVC) on an interface and sets the device into the config-if-srv submode.

	Command	Purpose
Step 5	**encapsulation dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **second-dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **Example:** Router(config-if-srv)# encapsulation default	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q, QinQ, or untagged frames on an interface for the appropriate service instance.
Step 6	**xconnect** peer-ip-address vc-id **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** pw-class pw-class-name **}[**pw-class pw-class-name**] [sequencing {transmit | receive | both}]** **Example:** Router(config)# xconnect 10.0.3.201 123 pw-class vlan-xconnect	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.
Step 7	**cfm mep domain** domain-name **[up | down] mpid** mpid-value **[cos** cos-value**]** **Example:** Router# cfm mep down mpid 100 domain Core	Configures a maintenance endpoint (MEP) for a domain.

Examples

This example shows how to configure CFM over EFP Interface with Selective QinQ cross connect:

```
PE3(config)#ethernet cfm domain L2 level 2
PE3(config-ecfm)# service s256 evc 256 direction down
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end
PE3#
PE3(config)#int ten 2/0/0
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256 second-dot1q 257 cos 7
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
PE3#
```

Configuring CFM over EFP Interface with Cross Connect—Port-Based Cross Connect Tunnel

This section describes how to configure CFM over EFP Interface with Port-Based cross connect Tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *slot/subslot/port*
4. **service instance** *id* **{Ethernet [***service-name***}**

5. **encapsulation dot1q {any | *vlan-id*[*vlan-id*[*vlan-id*]} second-dot1q {any |*vlan-id*[*vlan-id*[*vlan-id*]]}**
6. **xconnect** *peer-ip-address vc-id* **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** *pw-class pw-class-name* **}[***pw-class pw-class-name***] [sequencing {transmit | receive | both}]**
7. **cfm mep domain** *domain-name* **[up | down] mpid** *mpid-value* **[cos** *cos-value***]**
8. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot*/sub-slot/*port* **Example:** `Router(config)# interface Gi2/0/2`	Specifies the Gigabit Ethernet interface to configure, where: *slot/subslot/port*—Specifies the location of the interface.
Step 4	**service instance** id **Ethernet** [service-name] **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and sets the device into the config-if-srv submode.
Step 5	**encapsulation dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **second-dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **Example:** `Router(config-if-srv)# encapsulation dot1q 10-20, 30, 50-60`	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q, QinQ, or untagged frames on an interface for the appropriate service instance.

	Command	Purpose
Step 6	**xconnect** peer-ip-address vc-id **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** pw-class pw-class-name**}[**pw-class pw-class-name**] [sequencing {transmit | receive | both}]** **Example:** `Router(config)# xconnect 1.1.1.1 100 pw-class vlan-xconnect`	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.
Step 7	**cfm mep domain** domain-name **[up | down] mpid** mpid-value **[cos** cos-value**]** **Example:** `Router# cfm mep up mpid 100 domain Core`	Configures a maintenance endpoint (MEP) for a domain.

Examples

This example shows how to configure CFM over EFP Interface with Port-Based cross connect Tunnel:

```
PE3(config)#ethernet cfm domain L2 level 2
PE3(config-ecfm)# service s256 evc 256 direction down
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end
PE3#
PE3(config)#int ten 2/0/0
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
PE3#
```

Configuring CFM over EFP Interface with Cross Connect—Port Channel-Based Cross Connect Tunnel

This section describes how to configure CFM over EFP Interface with Port Channel-Based cross connect Tunnel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *slot/subslot/port*
4. **service instance** *id* **{Ethernet [***service-name***}**
5. **encapsulation dot1q {any \|** *vlan-id***[***vlan-id***[***vlan-id***]} second-dot1q {any \|***vlan-id***[***vlan-id***[***vlan-id***]]}**

6. **rewrite ingress tag** {**push** {**dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**]
7. **xconnect** *peer-ip-address vc-id* **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** *pw-class pw-class-name* **}[***pw-class pw-class-name***] [sequencing {transmit | receive | both}]**
8. **cfm mep domain** *domain-name* **[up | down] mpid** *mpid-value* **[cos** *cos-value***]**
9. **exit**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot*/sub-slot/*port* **Example:** `Router(config)# interface Port-channel 1`	Specifies the Gigabit Ethernet interface to configure, where: *slot/subslot/port*—Specifies the location of the interface.
Step 4	**service instance** id **Ethernet** [service-name] **Example:** `Router(config-if)# service instance 101 ethernet`	Creates a service instance (an instance of an EVC) on an interface and sets the device into the config-if-srv submode.
Step 5	**encapsulation dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **second-dot1q** {any | vlan-id[vlan-id[-vlan-id]]} **Example:** `Router(config-if-srv)# encapsulation dot1q 20 second-dot1q 30`	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q, QinQ, or untagged frames on an interface for the appropriate service instance.

	Command	Purpose
Step 6	**rewrite ingress tag** {**push {dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**] **Example:** `Router(config-if-srv)# rewrite ingress tag pop 2 symmetric`	Specifies the tag manipulation that is to be performed on the frame ingress to the service instance.
Step 7	**xconnect** peer-ip-address vc-id **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** pw-class pw-class-name **}[**pw-class pw-class-name**] [sequencing {transmit | receive | both}]** **Example:** `Router(config)# xconnect 1.1.1.1 100 pw-class vlan-xconnect`	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.
Step 8	**cfm mep domain** domain-name **[up | down] mpid** mpid-value **[cos** cos-value**]** **Example:** `Router# cfm mep up mpid 100 domain Core`	Configures a maintenance endpoint (MEP) for a domain.

Examples

This example shows how to configure CFM over EFP Interface with Port Channel-Based cross connect Tunnel:

```
PE3(config)#ethernet cfm domain L2 level 2
PE3(config-ecfm)# service s256 evc 256 direction down
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end
PE3#
PE3(config)#int port-20
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
```

Verification

Use the following commands to verify a configuration:

- Use the **show ethernet cfm ma remote** commands to verify the CFM over EVC configuration. This command shows the basic configuration information for CFM.

  ```
  Router-30-PE1#show ethernet cfm ma local
  ```

```
Local MEPs:
--------------------------------------------------------------------------------
MPID Domain Name                                Lvl   MacAddress     Type  CC
     Domain Id                                  Dir   Port           Id
     MA Name                                          SrvcInst
     EVC name
--------------------------------------------------------------------------------
1    L6                                         6     000a.f393.56d0 XCON  Y
     L6                                         Down    Te2/0/0        N/A
     bbb                                              1
     bbb
3    L5                                         5     0007.8478.4410 XCON  Y
     L5                                         Up    Te2/0/0        N/A
     bbb                                              1
     bbb

Total Local MEPs: 2

Local MIPs:
* = MIP Manually Configured
--------------------------------------------------------------------------------
 Level Port           MacAddress     SrvcInst   Type    Id
--------------------------------------------------------------------------------
 7     Te2/0/0        0007.8478.4410 1          XCON    N/A

Total Local MIPs: 1
```

- Use the **show ethernet cfm ma remote** to verify the MEP configuration:

```
Router-30-PE1#show ethernet cfm ma remote
--------------------------------------------------------------------------------
MPID  Domain Name                                MacAddress          IfSt  PtSt
 Lvl  Domain ID                                  Ingress
 RDI  MA Name                                    Type Id             SrvcInst
      EVC Name                                                       Age
--------------------------------------------------------------------------------
4     L5                                         000a.f393.56d0      Up    Up
 5    L5                                         Te2/0/0:(2.2.2.2, 1)
 -    bbb                                        XCON N/A            1
      bbb                                                            9s
2     L6                                         000a.f393.56d0      Up    Up
 6    L6                                         Te2/0/0:(2.2.2.2, 1)
 -    bbb                                        XCON N/A            1
      bbb                                                            1s

Total Remote MEPs: 2
```

- Use the **show ethernet cfm mpdb** command to verify the catalouge of CC with MIP in intermediate routers.

```
PE2#show ethernet cfm mpdb
* = Can Ping/Traceroute to MEP
--------------------------------------------------------------------------------
MPID  Domain Name                                MacAddress          Version
Lvl   Domain ID                                  Ingress
Expd  MA Name                                    Type Id             SrvcInst
      EVC Name                                                       Age
--------------------------------------------------------------------------------
600 * L6                                         0021.d8ca.d7d0      IEEE-CFM
6     L6                                         Te2/1:(2.2.2.2, 1)
-     s1                                         XCON N/A            1
      1                                                              2s
700   L7                                         001f.cab7.fd01      IEEE-CFM
```

```
7     L7                                          Te2/1:(2.2.2.2, 1)
-     s1                                          XCON N/A            1
      1                                                               3s

Total Remote MEPs: 2
```

- Use the **show mpls l2 transport vc 1 detail** commaned to show detailed configuration information:

```
PE1#sh mpls l2 vc 1 deta
Local interface: Te8/0/1 up, line protocol up, Eth VLAN 200 up
  Interworking type is Ethernet
  Destination address: 3.3.3.3, VC ID: 1, VC status: up
    Output interface: Te8/0/0, imposed label stack {21}
    Preferred path: not configured
    Default path: active
    Next hop: 20.1.1.2
  Create time: 21:13:27, last status change time: 02:55:33
  Signaling protocol: LDP, peer 3.3.3.3:0 up
    Targeted Hello: 2.2.2.2(LDP Id) -> 3.3.3.3, LDP is UP
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last local SSS circuit status rcvd: No fault
      Last local SSS circuit status sent: No fault
      Last local  LDP TLV    status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 21, remote 21
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  VC statistics:
    transit packet totals: receive 37, send 1067452272
    transit byte totals:   receive 4181, send 72586757556
    transit packet drops:  receive 0, seq error 0, send 0
```

- Use **show mpls forwarding-table** command to verify the cross connect VC:

```
PE1#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
17         Pop Label  3.3.3.3/32       23038746624   Te8/0/0    20.1.1.2
21         No Label   l2ckt(1)         4181          Te8/0/1    point2point
```

- Use **show ethernet cfm error** command to view the error report:

```
PE2#show ethernet cfm error
------------------------------------------------------------------------------
MPID Domain Id                                 Mac Address     Type   Id  Lvl
     MAName                                    Reason                 Age
------------------------------------------------------------------------------
  -  L3                                        001d.45fe.ca81  BD-V   200  3
     s2                                        Receive AIS            8s
PE2#
```

Configuring CFM over EFP Interface with xconnect—Port Channel-Based xconnect Tunnel

Use the following commands at the customer facing port:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface type** *slot/subslot/port*
4. **service instance** *id* {**Ethernet** [*service-name*]}
5. **encapsulation untagged | dot1q {any | *vlan-id*[*vlan-id*[*vlan-id*]]} second-dot1q {any |*vlan-id*[*vlan-id*[*vlan-id*]]}**
6. **rewrite ingress tag** {**push {dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**]
7. **xconnect** *peer-ip-address vc-id* **{encapsulation {l2tpv3 [manual] | mpls [manual]} | pw-class** *pw-class-name* **}[pw-class** *pw-class-name*] **[sequencing {transmit | receive | both}]**
8. **cfm mep domain** *domain-name* **mpid** *mpid-value* **[cos** *cos-value*]

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface gigabitethernet** *slot*/sub-slot/*port* **Example:** `Router(config)# interface Port-channel 1`	Specifies the Gigabit Ethernet interface to configure, where: *slot/subslot/port*—Specifies the location of the interface.
Step 4	**service instance** id **Ethernet** [service-name] **Example:** `Router(config-if)# service instance 101 ethernet`	Creates a service instance (an instance of an EVC) on an interface and sets the device into the config-if-srv submode.

	Command	Purpose
Step 5	**encapsulation untagged dot1q** {any | vlan-id[vlan-id[vlain-id]]} **second-dot1q** {any | vlan-id[vlan-id[vlan-id]]} **Example:** Router(config-if-srv)# encapsulation dot1q 20 second-dot1q 30	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q, QinQ, or untagged frames on an interface for the appropriate service instance.
Step 6	**rewrite ingress tag** {**push {dot1q** *vlan-id* | **dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **pop** {**1** | **2**} | **translate** {**1-to-1** {**dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **2-to-1 dot1q** *vlan-id* | **dot1ad** *vlan-id*}| **1-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*} | **2-to-2** {**dot1q** *vlan-id* **second-dot1q** *vlan-id* | **dot1ad** *vlan-id* **dot1q** *vlan-id*}} [**symmetric**] **Example:** Router(config-if-srv)# rewrite ingress tag pop 2 symmetric	Specifies the tag manipulation that is to be performed on the frame ingress to the service instance.
Step 7	**xconnect** peer-ip-address vc-id **{encapsulation {l2tpv3 [manual] | mpls [manual]} |** pw-class pw-class-name **}[**pw-class pw-class-name**] [sequencing {transmit | receive | both}]** **Example:** Router(config)# xconnect 1.1.1.1 100 pw-class vlan-xconnect	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.
Step 8	**cfm mep domain** domain-name **[up | down] mpid** mpid-value **[cos** cos-value**]** **Example:** Router# cfm mep up mpid 100 domain Core	Configures a maintenance endpoint (MEP) for a domain.

Examples

This example shows how to configure CFM over EFP Interface with Port Channel-Based xconnect Tunnel:

```
PE3(config)#ethernet cfm domain L2 level 2
PE3(config-ecfm)# service s256 evc 256 direction down
PE3(config-ecfm-srv)#  continuity-check
PE3(config-ecfm-srv)#end
PE3#
PE3(config)#int port-20
PE3(config-if)#no ip address
PE3(config-if)# service instance 256 ethernet 256
PE3(config-if-srv)#  encapsulation dot1q 256
PE3(config-if-srv)#  xconnect 1.1.1.1 1 encapsulation mpls
PE3(cfg-if-ether-vc-xconn)#  cfm mep domain L6 mpid 256
PE3(config-if-srv-ecfm-mep)#end
```

Verification

Use the following commands to verify a configuration:

- Use **show ethernet cfm ma remote** commands to verify the CFM over EFP configuration. This command shows the basic configuration information for CFM.

```
Router-30-PE1#show ethernet cfm ma local
Local MEPs:
--------------------------------------------------------------------------------
MPID Domain Name                                Lvl   MacAddress     Type  CC
     Domain Id                                  Dir   Port           Id
     MA Name                                          SrvcInst
     EVC name
--------------------------------------------------------------------------------
1    L6                                         6     000a.f393.56d0 XCON  Y
     L6                                         Down    Te2/0/0        N/A
     bbb                                              1
     bbb
3    L5                                         5     0007.8478.4410 XCON  Y
     L5                                         Up    Te2/0/0        N/A
     bbb                                              1
     bbb

Total Local MEPs: 2

Local MIPs:
* = MIP Manually Configured
--------------------------------------------------------------------------------
 Level Port           MacAddress     SrvcInst   Type    Id
--------------------------------------------------------------------------------
 7     Te2/0/0        0007.8478.4410 1          XCON    N/A

Total Local MIPs: 1
```

- Use **show ethernet cfm ma remote** to verify the MEP configuration:

```
Router-30-PE1#show ethernet cfm ma remote
--------------------------------------------------------------------------------
MPID  Domain Name                                MacAddress          IfSt  PtSt
 Lvl  Domain ID                                  Ingress
 RDI  MA Name                                    Type Id             SrvcInst
      EVC Name                                                       Age
--------------------------------------------------------------------------------
4     L5                                         000a.f393.56d0      Up    Up
 5    L5                                         Te2/0/0:(2.2.2.2, 1)
 -    bbb                                        XCON N/A            1
      bbb                                                            9s
2     L6                                         000a.f393.56d0      Up    Up
 6    L6                                         Te2/0/0:(2.2.2.2, 1)
 -    bbb                                        XCON N/A            1
      bbb                                                            1s

Total Remote MEPs: 2
```

- Use **show ethernet cfm mpdb** command to verify the catalogue of CC with MIP in intermediate routers.

```
PE2#show ethernet cfm mpdb
* = Can Ping/Traceroute to MEP
--------------------------------------------------------------------------------
MPID  Domain Name                                MacAddress          Version
Lvl   Domain ID                                  Ingress
```

```
Expd  MA Name                                          Type Id             SrvcInst
      EVC Name                                                             Age
-------------------------------------------------------------------------------
600 * L6                                               0021.d8ca.d7d0      IEEE-CFM
6     L6                                               Te2/1:(2.2.2.2, 1)
-     s1                                               XCON N/A            1
      1                                                                    2s
700   L7                                               001f.cab7.fd01      IEEE-CFM
7     L7                                               Te2/1:(2.2.2.2, 1)
-     s1                                               XCON N/A            1
      1                                                                    3s

Total Remote MEPs: 2
```

- Use **show mpls l2 transport vc 1 detail** command to show detailed configuration information:

```
PE1#sh mpls l2 vc 1 deta
Local interface: Te8/0/1 up, line protocol up, Eth VLAN 200 up
  Interworking type is Ethernet
  Destination address: 3.3.3.3, VC ID: 1, VC status: up
    Output interface: Te8/0/0, imposed label stack {21}
    Preferred path: not configured
    Default path: active
    Next hop: 20.1.1.2
  Create time: 21:13:27, last status change time: 02:55:33
  Signaling protocol: LDP, peer 3.3.3.3:0 up
    Targeted Hello: 2.2.2.2(LDP Id) -> 3.3.3.3, LDP is UP
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last local SSS circuit status rcvd: No fault
      Last local SSS circuit status sent: No fault
      Last local  LDP TLV    status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 21, remote 21
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  VC statistics:
    transit packet totals: receive 37, send 1067452272
    transit byte totals:   receive 4181, send 72586757556
    transit packet drops:  receive 0, seq error 0, send 0
```

- Use **show mpls forwarding-table** command to verify the xconnect VC:

```
PE1#show mpls forwarding-table
Local      Outgoing   Prefix           Bytes Label   Outgoing   Next Hop
Label      Label      or Tunnel Id     Switched      interface
17         Pop Label  3.3.3.3/32       23038746624   Te8/0/0    20.1.1.2
21         No Label   l2ckt(1)         4181          Te8/0/1    point2point
```

- Use **show ethernet cfm error** command to view the error report:

```
PE2#show ethernet cfm error
-------------------------------------------------------------------------------
MPID Domain Id                                 Mac Address     Type   Id  Lvl
     MAName                                    Reason                 Age
-------------------------------------------------------------------------------
```

```
          -  L3                                          001d.45fe.ca81  BD-V   200  3
             s2                                          Receive AIS            8s
PE2#
```

Troubleshooting CFM Features

provides troubleshooting solutions for the CFM features.

Table 1 Troubleshooting Scenarios for CFM Features

Problem	Solution
When you configure CFM, the message “Match registers are not available” is displayed.	Use the **show platform mrm info** command on the SP console to verify the match registers. Based on the derived output, perform these tasks: 1. Check if the line card supports the CFM feature. 2. Enable CFM across the system to allow co-existence with other protocols. 3. Ensure that no CFM traffic is present in any supervisor or ports. 4. Configure STP mode to Multiple Spanning Tree (MST) and re-enable CFM or disable CFM completely. For more information on match registers, see *Ethernet Connectivity Fault Management* at http://www.cisco.com/en/US/docs/ios/12_2sr/12_2sra/feature/guide/srethcfm.html. CFM uses two match registers to identify the control packet type and each VLAN spanning tree also uses a match register to identify its control packet type. For both protocols to work on the same system, each line card should support three match registers, and at least one supporting only a 44 bit MAC match.
CFM configuration errors	CFM configuration error occurs when when a MEP receives a continuity check with an overlapping MPID. To verify the source of the error, use the command **show ethernet cfm errors configuration** or **show ethernet cfm errors**.
CFM ping and traceroute result is "not found"	Complete these steps: 1. Use **show run ethernet cfm** to view all CFM global configurations. 2. Use **show ethernet cfm location main** to view local MEPs and their CCM statistics 3. Use **show ethernet cfm peer meps** command to View CFM CCM received from Peer MEPs. 4. Use **trace ethernet cfm** command to start a CFM trace.

Table 1 Troubleshooting Scenarios for CFM Features

Problem	Solution
CFM connectivity is down and issues at the maintenance domain levels	Use the **ping ethernet {mac-address | mpid id | multicast} domain domain-name { vlan vlan-id | port | evc evc-name }** or t**raceroute ethernet {mac-address | mpid id } domain domain-name { vlan vlan-id | port | evc evc-name }** commands to verify ethernet CFM connectivity. Share the output with TAC for further investigation.

Loop trap error	Use the **show ethernet cfm error** command to check for Loop Trap errors as shown here:
	`CE(config-if)#do sh ethernet cfm err`
	`----------`
	`----------`
	`Level Vlan MPID Remote MAC Reason`
	`Service ID`
	`----------`
	`----------`
	`5 711 550 1001.1001.1001 Loop Trap Error`
	`OUT`
	`PE#sh ethernet cfm err`
	`----------`
	`----------`
	`Level Vlan MPID Remote MAC Reason`
	`Service ID`
	`----------`
	`----------`
	`5 711 550 1001.1001.1001 Loop Trap Error`
	`OUT`

Module has insufficient match registers	Complete these steps: 1. Verify and confirm if a unsupported line card is inserted into the router. 2. If yes, perform an OIR of the unsupported line card.
CFM is deactivated	Complete these steps: 1. Check if all the line cards have free match reagisters. 2. Check if CFM is activated on supervisor cards. CFM is not supported on supervisor cards that has two match registers. In this scenario, CFM is automatically disabled on the SUP ports and enabled on the remaining line cards.
ethernet cfm logging	In a scale scenario, you configure either the console logging rate-limiting using **logging rate-limit** or using **logging buffered** instead of using **logging console**. The suggested rate-limit is around 30 messages per second.