

# Diameter, Configure Own Node

## OPERATING INSTRUCTIONS

**Copyright**

© Ericsson AB 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Description</b>	<b>1</b>
<b>2</b>	<b>Procedure</b>	<b>1</b>
2.1	Configure Own Node using CLI	1
2.2	Configure Own Node using NETCONF	3



Diameter, Configure Own Node



# 1 Description

This instruction describes how to configure the Diameter own node, acting either as a client, as a server, or as an agent.

The Diameter Base Protocol allows the establishment of direct connections between a generic Diameter node (called the own node) and one or more Diameter peer nodes. Diameter messages can be transferred over the Transmission Control Protocol (TCP) or the Stream Control Transmission Protocol (SCTP), over IPv4 or IPv6.

# 2 Procedure

## Prerequisites

- No documents are required.
- No tools are required.
- The following conditions must apply:
  - The following mandatory attributes are known:
    - The unique node identifier, formatted as a case-insensitive Fully Qualified Domain Name (FQDN).
    - The name of the Diameter product running on the own node, set to the default value `Ericsson Diameter Stack`.
    - The unique own node realm, formatted as a case-insensitive FQDN, used for Domain Name System (DNS) message routing.
    - The identifiers of all vendors whose Attribute-Value Pairs (AVPs) are supported by the own node, in a list.
  - The desired optional attributes are known.

## 2.1 Configure Own Node using CLI

### Prerequisites

- An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.

### Steps



1. Navigate to the `DIA-CFG-OwnNodeConfig` Managed Object (MO), for example:

```
>dn ManagedElement=NODE06ST,XYZFunction=xyz,DIA-CFG-Application=DIA,DIA-CFG-StackContainer=abc,DIA-CFG-OwnNodeConfig=abc
```

2. Enter Config mode:

```
(DIA-CFG-OwnNodeConfig=abc)>configure
```

3. Is it only needed to enable or disable the own node?

Yes: Proceed with Step 11.

No: Continue with the next step.

4. Set `hostId` to the node identifier of the own node, for example:

```
(config-DIA-CFG-OwnNodeConfig=abc)>hostId="dia.node"
```

5. Set `productName` to the name of the Diameter product running on the own node, for example:

```
(config-DIA-CFG-OwnNodeConfig=abc)>productName="Ericsson  
Diameter Stack"
```

6. Set `realm` to the name of the node realm, for example:

```
(config-DIA-CFG-OwnNodeConfig=abc)>realm="ericsson.com"
```

7. Set `supportedVendorsIds` to the vendor IDs, for example:

```
(config-DIA-CFG-OwnNodeConfig=abc)>supportedVendorsIds="193"
```

8. Set the `transportLayerType` and `ipAddressesList` attributes to the `DIA-CFG-OwnNodeConfig` MO, for example.

```
(config-DIA-CFG-OwnNodeConfig=abc)>transportLayerType=1
```

```
(config-DIA-CFG-OwnNodeConfig=abc)>ipAddressesList="0:10.1.137  
.2"
```

```
(config-DIA-CFG-OwnNodeConfig=abc)>ipAddressesList="1:2dea::66  
:2"
```

9. Commit the settings:

```
(config-DIA-CFG-OwnNodeConfig=abc)>commit -s
```

10. Verify the own node configuration result:

```
(DIA-CFG-OwnNodeConfig=abc)>show
```

The following is an example output:



```
allowConnectFromUnknownNode=false
enabled=false
firmwareRevision="0"
hostId="dia.node"
ipAddressesList
  "0:10.1.137.2"
  "1:2dea::66:2"
loadRegulationEnabled=false
productName="Ericsson Diameter Stack"
realm="ericsson.com"
sctpHandlerLogLevel="DEFAULT"
sendErrorAtOverload=false
supportedVendorsIds
  "193"
traceSctpHandler="DEFAULT"
transportLayerType="1"
```

11. Enable the own node:

```
(config-DIA-CFG-OwnNodeConfig=abc)>enabled=true
```

**Note:** To disable the own node, set enabled to false.

12. Commit the settings:

```
(config-DIA-CFG-OwnNodeConfig=abc)>commit
```

13. Verify the own node configuration result:

```
(DIA-CFG-OwnNodeConfig=abc)>show enabled
```

The following is an example output:

```
enabled=true
```

## 2.2 Configure Own Node using NETCONF

### Steps

1. Prepare an XML file according to the following template:



```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>]]>]]>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running/></target>
    <config>
      <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
        <managedElementId>1</managedElementId>
        <XYZFunctionBranch>
          <DIA-CFG-Application xmlns="urn:com:ericsson:ecim:dia_mim">
            <applicationName>DIA</applicationName>
            <DIA-CFG-StackContainer>
              <stackContainerId><applStackId></stackContainerId>
              <DIA-CFG-OwnNodeConfig operation="replace">
                <stackId><applStackId></stackId>
                <attribute1>value1</attribute1>
                ...
                <attributeN>valueN</attributeN>
              </DIA-CFG-OwnNodeConfig>
            </DIA-CFG-StackContainer>
          </DIA-CFG-Application>
        </XYZFunctionBranch>
      </ManagedElement>
    </config>
  </edit-config>
</rpc>]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>]]>]]>
```

**Note:** The XYZFunctionBranch part can be composed of many elements depending on where the Diameter subtree is connected in the managed element structure. In the example below, we assume that the parent MOs are already created and the XYZFunctionBranch is empty.

IP addresses specified by the sctpAddressesList attribute can be supplemented optionally by a VPN (ALB) name suffix, separated by semicolon.

Example to Modify Own Node using NETCONF:





```
<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>]]>]]>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running/></target>
    <config>
      <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
        <managedElementId>1</managedElementId>
        <DIA-CFG-Application xmlns="urn:com:ericsson:ecim:dia_mim">
          <applicationName>DIA</applicationName>
          <DIA-CFG-StackContainer>
            <stackContainerId>TST_SRV</stackContainerId>
            <DIA-CFG-OwnNodeConfig operation="replace">
              <stackId>TST_SRV</stackId>
              <hostId>hss1.ericsson.se</hostId>
              <allowConnectFromUnknownNode>true</allowConnectFromUnknownNode>
              <watchdogTimeIdle>25</watchdogTimeIdle>
              <maxNumberOfRetries>8</maxNumberOfRetries>
              <maxRequestPendingTime>5</maxRequestPendingTime>
              <tcTimer>45</tcTimer>
              <realm>ericsson.se</realm>
              <ipAddressesList>0:10.1.137.2</ipAddressesList>
              <sctpAddressesList>0:10.1.137.4;alb_0</sctpAddressesList>
              <sctpAddressesList>0:10.1.137.5;alb_1</sctpAddressesList>
              <transportLayerType>3</transportLayerType>
              <maxOutboundSctpStreams>3</maxOutboundSctpStreams>
              <maxInboundSctpStreams>5</maxInboundSctpStreams>
              <diaVendorId>7</diaVendorId>
              <productName>Ericsson Diameter Stack</productName>
              <firmwareRevision>3</firmwareRevision>
              <supportedAuthAppIds>0</supportedAuthAppIds>
              <portNr>40000</portNr>
              <enabled>false</enabled>
              <loadRegulationEnabled>true</loadRegulationEnabled>
              <sendErrorAtOverload>true</sendErrorAtOverload>
            </DIA-CFG-OwnNodeConfig>
          </DIA-CFG-StackContainer>
        </DIA-CFG-Application>
      </ManagedElement>
    </config>
  </edit-config>
</rpc>]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>]]>]]>
```

2. Pass the prepared XML file to shell:

```
ssh -p 830 <user_name>@<node_vip_address> -s -t =>
netconf < <xml_file>
```

<xml\_file> refers to the XML file name.

3. Enable Own Node. Prepare an XML file according to the following template:



```

<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>]]>]]>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running/></target>
    <config>
      <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
        <managedElementId>1</managedElementId>
        <XYZFunctionBranch>
          <DIA-CFG-Application xmlns="urn:com:ericsson:ecim:dia_mim">
            <applicationName>DIA</applicationName>
            <DIA-CFG-StackContainer>
              <stackContainerId><applStackId></stackContainerId>
              <DIA-CFG-OwnNodeConfig operation="replace">
                <stackId><applStackId></stackId>
                <enabled>true</enabled>
              </DIA-CFG-OwnNodeConfig>
            </DIA-CFG-StackContainer>
          </DIA-CFG-Application>
        </XYZFunctionBranch>
      </ManagedElement>
    </config>
  </edit-config>
</rpc>]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>]]>]]>

```

**Note:** The XYZFunctionBranch part can be composed of many elements depending on where the Diameter subtree is connected in the managed element structure. In the example below, we assume that the parent MOs are already created and the XYZFunctionBranch is empty.

Example to enable Own Node using NETCONF:

```

<?xml version="1.0" encoding="UTF-8"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:netconf:base:1.0</capability>
  </capabilities>
</hello>]]>]]>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running/></target>
    <config>
      <ManagedElement xmlns="urn:com:ericsson:ecim:ComTop">
        <managedElementId>1</managedElementId>
        <DIA-CFG-Application xmlns="urn:com:ericsson:ecim:dia_mim">
          <applicationName>DIA</applicationName>
          <DIA-CFG-StackContainer>
            <stackContainerId>TST_SRV</stackContainerId>
            <DIA-CFG-OwnNodeConfig operation="replace">
              <stackId>TST_SRV</stackId>
              <enabled>true</enabled>
            </DIA-CFG-OwnNodeConfig>
          </DIA-CFG-StackContainer>
        </DIA-CFG-Application>
      </ManagedElement>
    </config>
  </edit-config>
</rpc>]]>]]>
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <close-session/>
</rpc>]]>]]>

```



To disable Peer Node, set `enabled` to `false` in the provided example.

4. Pass the prepared XML file to shell:

```
ssh -p 830 <user_name>@<node_vip_address> -s -t =>  
netconf < <xml_file>
```

`<xml_file>` refers to the XML file name.