



CHAPTER 2

MWTM 6.1.3 Inventory API

MWTM 6.1.3 collects device configuration and status information via discovery, status polling, and SNMP trap processing. The MWTM 6.1.3 data model processes the device information and stores the information in the inventory database.

MWTM 6.1.3 provides Remote Procedure Call (RPC) style Inventory API to the Northbound OSS. The OSS can send a SOAP request to the MWTM 6.1.3 Inventory API, and the MWTM 6.1.3 responds with SOAP responses.

This chapter includes the following sections:

- [Network Elements and FQDNs, page 2-1](#)
- [MWTM 6.1.3 Inventory API Operations, page 2-15](#)

Network Elements and FQDNs

An inventory object is also referred to as a Network Element (NE). MWTM 6.1.3 inventory objects are organized in a tree structure. This tree structure is similar to what is visible in the MWTM 6.1.3 GUI. Each inventory object is identified by its location in the tree.

The following sections further describe MWTM network elements:

- [Understanding the MWTM Inventory Tree, page 2-1](#)
- [Building Fully Qualified Domain Names, page 2-6](#)
- [Understanding Network Element Attributes, page 2-9](#)
- [Understanding Network Element Information Types, page 2-11](#)

Understanding the MWTM Inventory Tree

The MWTM Inventory tree is similar to the navigation tree in the MWTM GUI (see the *User Guide for the Cisco Mobile Wireless Transport Manager 6.1.3* for more information about MWTM navigation tree). Here are some differences between the MWTM inventory tree and the MWTM navigation tree.

- A network element appears at different paths in the MWTM GUI tree. For example, the RAN shorthaul serial interface appears in the GUI tree at Node > Ran Backhaul > Serial Interface / GSM Shorthaul. It also appears in the GUI tree at Node > Physical Folder > T1/E1 interface > Serial Interface.
- A network element can only appear at one unique path in MWTM Inventory tree. It does not appear in multiple paths.

- The MWTM Inventory API accepts both paths from the inventory tree and the GUI tree for network element querying operations. MWTM responds with a normalized FQDN (inventory tree).

The following sections contain these inventory tree examples:

- [ITP Inventory Tree, page 2-2](#)
- [RAN Inventory Tree, page 2-3](#)
- [IPRAN Inventory tree, page 2-3](#)
- [ONS Inventory Tree, page 2-4](#)
- [BWG Tree Inventory, page 2-4](#)
- [CSG2 Tree Inventory, page 2-4](#)
- [GGSN Tree Inventory, page 2-5](#)
- [HA Tree Inventory, page 2-5](#)
- [PDSN Tree Inventory, page 2-5](#)



Note

Object types with an asterisk (*) indicate the object type is an abbreviation for a long name. When referencing an object in MWTM 6.1.3, the client code can use either a short or long name. This short name to long name mapping can be configured in:

```
${MWTM_INSTALL_BASEDIR}/properties/NEType.properties.
```

SP is short name for “SignalingPoint”

AS is short name for “ApplicationServer”

ASP is short name for “ApplicationServerProcess”

ASPA is short name for “ApplicationServerProcessAssociation”

SGMP is short name for “SignalingGatewayMatedPair”

RBH is short name for “RanBackhaul”

ITP Inventory Tree

The following is an example Inventory tree structure for an ITP node (does not include configured network elements):

```
Node (subtype=ITP)
|
+-- SP*
|   |
|   +-- Linkset
|   |   |
|   |   +-- Link
|   |
|   +-- AS*
|   |   |
|   |   +-- ASPA*
|
+-- SGMP*
|
+-- Interface
|   |
|   +-- Interface
|
+-- Interface
|
```

```

+-- Folder
|
+-- Folder

```

RAN Inventory Tree

The following is an example Inventory tree structure for a Radio Access Network (RAN) node:

```

Node (subtype=RAN)
|
+-- RBH*
|
+-- Interface
|   |
|   +-- Interface (subtype=GSM)
|
+-- Interface (subtype=UMTS)
|   |
|   +-- Interface (subtype=UMTS)
|
+-- Folder
|
+-- Folder

```

IPRAN Inventory tree

The following is an example Inventory tree structure for an IP Radio Access Network (IPRAN):

```

Node (subtype=ipran)
|
+-- PWE3 Backhaul
|   |
|   +-- PWE3 Virtual Circuit
|
+-- Interface
|   |
|   +-- Interface
|
+-- Interface
|   |
|   +-- Interface
|
+-- Folder
|
+-- Folder

```

ONS Inventory Tree

The following is an example of a tree structure for an Optical Networking System (ONS) node:

```

Node (subtype=ONS)
|
+--Node (subtype=RAN_SVC)
|
|   +-- RBH*
|   |
|   +-- Interface
|   |   |
|   |   +-- Interface (subtype=GSM)
|   |
|   +-- Interface (subtype=UMTS)
|   |   |
|   |   +-- Interface (subtype=UMTS)
|   |
|   +-- Folder
|   |
|   +-- Folder
|
+-- Card
|   |
|   +-- Interface
|   |
|   +-- Interface
|
+-- Card
|   |
|   +-- Interface
|
+-- Folder
|
+-- Folder
    
```

BWG Tree Inventory

The following is an example inventory tree structure for an Broadband Wireless Gateway (BWG) node:

```

Node (subtype=BWG)
|
+-- Interface
|   |
|   +-- Interface
|
+-- Interface
|   |
|   +-- Interface
|
+-- Folder
|
+-- Folder
    
```

CSG2 Tree Inventory

The following is an example inventory tree structure for a Content Services Gateway 2 (CSG2) node:

```

Node (subtype=CSG2)
|
+-- Interface
|   |
    
```

```

|   +-- Interface
|   |
+-- Interface
|   |
|   +-- Interface
|   |
+-- Folder
|   |
+-- Folder

```

GGSN Tree Inventory

The following is an example inventory tree structure for a Gateway GPRS Support node (GGSN):

```

Node (subtype=GGSN)
|
+-- APN Instance
|
+-- Interface
|   |
|   +-- Interface
|   |
+-- Interface
|   |
|   +-- Interface
|   |
+-- Folder
|   |
+-- Folder

```

HA Tree Inventory

The following is an example inventory tree structure for a Home Agent (HA) node:

```

Node (subtype=HA)
|
+-- Interface
|   |
|   +-- Interface
|   |
+-- Interface
|   |
|   +-- Interface
|   |
+-- Folder
|   |
+-- Folder

```

PDSN Tree Inventory

The following is an example inventory tree structure for a PDSN Tree Inventory node.

```

Node (subtype=PDSN)
|
+-- Interface
|   |
|   +-- Interface
|   |
+-- Interface
|   |
|   +-- Interface

```

```

|
+-- Folder
|
+-- Folder

```

SGW Tree Inventory

The following is an example inventory tree structure for a SGW Tree Inventory node.

```

Node (subtype=SGW)
|
+-- APN Instance
|
+-- Interface
| |
| +-- Interface
|
+-- Interface
| |
| +-- Interface
|
+-- Folder
|
+-- Folder

```

PDNGW Tree Inventory

The following is an example inventory tree structure for a PDNGW Tree Inventory node.

```

Node (subtype=PDNGW)
|
+-- APN Instance
|
+-- Interface
| |
| +-- Interface
|
+-- Interface
| |
| +-- Interface
|
+-- Folder
|
+-- Folder

```

Building Fully Qualified Domain Names

A Fully Qualified Domain Name (FQDN) identifies a network element in the MWTM 6.1.3 inventory. It is the full tree path to a network element from the root. An FQDN is composed of one or more relative component names, separated by a comma (,) character.

The following are examples of FQDN:

- Node=sgm-75-80a
- Node=sgm-75-80a,SP=itunet0
- Node=sgm-75-80a,SP=itunet0,Linkset=7580a_to_7692a0,Link=0
- Node=ems1900kk
- Node=ems1900kk,Interface=E1 0/0,Interface=Serial0/0:0
- Node=ems15454ea,Node=emsskyla2,Interface=GigabitEthernet0/0

Each of the component names is composed of an object type and an object identifier. The object identifier is also a Relative Distinguished Name (RDN). An RDN uniquely identifies a child inventory object under a given network element.

The following are example RDNs from previous FQDN examples:

- Node=sgm-75-80a
- SP=itunet0
- Linkset=7580a_to_7692a0
- Link=0
- Node=ems1900kk
- Interface=E1 0/0
- Interface=Serial0/0:0
- Node=ems15454ea
- Node=emsskyla2
- Interface=GigabitEthernet0/0

[Table 2-1 on page 2-8](#) lists all the object types and object identifiers in MWTM 6.1.3. In this table:

- An RDN uniquely identifies a child inventory object under a given network element.
- The object subtype is not part of the FQDN format. This attribute is used to identify subtypes for nodes and interfaces.

Table 2-1 Object Type and Object Identifier

Object Type	Object Identifier (RDN)	Possible Object Subtypes	Notes
Node	Node Name or Node IP Address	BWG, CSG1, CSG2, GGSN, ITP, ONS, IP-RAN, RANSVC (RAN Service Module), HA, CDT, mSEF, MetroE Switch, Generic, and PDSN.	A node network element supports multiple ways of lookup. It can be accessed either by name or by IP address. For example, because the sgm-75-80a node has two IP addresses, 172.18.16.18 and 172.18.16.242, the following FQDN representations can refer to the same node object: <ul style="list-style-type: none"> • Node=sgm-75-80a • Node=172.18.16.18 • Node=172.18.16.242
SP (SignalingPoint)	Network Name, Instance Number, or SP Name	.	The object identifier for the signaling point is the network name. If a signaling point did not define a network name, the object identifier will be an empty string. In this scenario, the FQDN for the signaling point is: "Node=xxxxx,SP=".
Linkset	Linkset Name	.	
Link	Link SLC Number	.	
AS (ApplicationServer)	AS Name	.	
ASP (ApplicationServerProcess)	ASP Name	.	An application server process (ASP) might have multiple names. An object identifier with any of these names represents the same ASP.
ASPA (ApplicationServerProcess Association)	ASP Name	.	
SGMP (SignalingGatewayMated Pair)	SGMP Name	.	

Table 2-1 Object Type and Object Identifier (continued)

Object Type	Object Identifier (RDN)	Possible Object Subtypes	Notes
Interface	Interface Name	GSM (Global System for Mobile Communication) or UMTS (Universal Mobile Telecommunications System), E1, T1, Serial, CEM, Ethernet, and FastEthernet, GigabitEthernet, ATM, Loopback, Tunnel, IMA, VLAN, SONET, ATMSubInf, GigabitEthernetSubInf, PortChannel, Multilink, FastEthernetSubInf, TenGigabitEthernet, TenGigabitEthernetSubInf, Virtual-Template.	
RBH (RAN Backhaul)	Ran Backhaul Name		
Card	Card Slot Number		
Folder	Folder Name		
APN (Access Point Name)	Name		Root level object
APN Instance (Access Point Name Instance)	Name		Descendant of a GGSN node
PWE3Backhaul	Name		Name consists of the local loopback address concatenated to the remote peer IP address (i.e 127.0.0.1/1.1.1.1)
PWE3VirtualCircuit	Name		Name consists of the VC ID concatenated to the remote peer IP address (i.e 1/1.1.1.1)

Understanding Network Element Attributes

The MWTM 6.1.3 Inventory API represents a network element in XML (Extensible Markup Language) format. Each network element is represented by its parent FQDN, object type, and a list of attributes.

An attribute can be either:

- A simple name/value pair, represented by an “<Attribute>” tag.
- A group of attributes, represented by an “<AttributeGroup>” tag.

For a complete definition of the inventory XML syntax, see [Appendix A, “MWTM NAPI WSDL and XSD Definitions.”](#)

The following is an example XML representation of an ITP node:

```
<NetworkElement type="Node" subtype="ITP" ParentFQDN="ParentFQDN=" ">
  <Attribute name="RDN">sgm-75-92b.cisco.com</Attribute>
  <Attribute name="State">Warning</Attribute>
  <Attribute name="StateReason">Interface Down</Attribute>
```

```

<Attribute name="StateTimestamp">2006-08-28T23:56:01.360-04:00</Attribute>
<Attribute name="IgnoreState">false</Attribute>
<Attribute name="DefaultIcon">Cisco7507</Attribute>
<Attribute name="CustomIcon"></Attribute>
<Attribute name="CustomName"></Attribute>
<Attribute name="SysUpTime">P3DT6H40M47.550S</Attribute>
<Attribute name="RebootReason">reload</Attribute>
<AttributeGroup name="IPAddresses">
  <Attribute name="Addresses">172.18.17.35</Attribute>
  <Attribute name="Addresses">172.18.17.163</Attribute>
  <Attribute name="PollingTimestamps">2006-08-28T23:58:03.410-04:00</Attribute>
  <Attribute name="PollingTimestamps"></Attribute>
  <Attribute name="SNMPFlags">1</Attribute>
  <Attribute name="SNMPFlags">2</Attribute>
  <Attribute name="Statuses">Active</Attribute>
  <Attribute name="Statuses">Active</Attribute>
  <Attribute name="PrimaryAddress">172.18.17.35</Attribute>
  <Attribute name="LastPolledAddress">172.18.17.35</Attribute>
</AttributeGroup>
<Attribute name="EnableProcessTraps">>true</Attribute>
<Attribute name="DiscoveredTimestamp">2006-08-28T05:52:48.819-04:00</Attribute>
<Attribute name="DeviceType">Cisco7507z</Attribute>
<Attribute name="SysDescr">Cisco IOS Software, RSP Software (RSP-ITPK91V-M),
Experimental Version 12.2(2006
0825:064354) [stklein-topsail_s_nightly 104]
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Fri 25-Aug-06 06:26 by stklein</Attribute>
<Attribute name="SysName">sgm-75-92b.cisco.com</Attribute>
<Attribute name="LastPollTimestamp">2006-08-28T23:58:03.410-04:00</Attribute>
<Attribute name="LastPollTimePeriod">PT17.096S</Attribute>
<Attribute name="AveragePollTimePeriod">PT16.420S</Attribute>
<Attribute name="TelnetAddress"></Attribute>
<Attribute name="MIBLevel">12.2(25)SW4</Attribute>
<Attribute name="SerialNumber"></Attribute>
<Attribute name="EnableTrapPolling">false</Attribute>
<Attribute name="EnableReportPolling">false</Attribute>
<Attribute name="CLLICode">clli_9572b</Attribute>
<Attribute name="NSOConfig">None</Attribute>
<Attribute name="RFState">Standby Hot</Attribute>
<Attribute name="Mtp3Offload">Main</Attribute>
</NetworkElement>

```

The following is an example XML representation of an ITP link:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:NetworkElement xmlns:ns2="http://cisco.com/mwtm" type="Link"
parentFQDN="Node=sgm-26-96a.cisco.com,SP=ansi-config,Linkset=2696a_to_2696b">
  <Attribute name="RDN">0</Attribute>
  <Attribute name="State">Active</Attribute>
  <Attribute name="StateReason">None</Attribute>
  <Attribute name="StateTimestamp">2006-11-27T11:38:00.602-05:00</Attribute>
  <Attribute name="IgnoreState">false</Attribute>
  <Attribute name="QOS">0</Attribute>
  <Attribute name="ConfigRemotePort">0</Attribute>
  <Attribute name="RemotePort">0</Attribute>
  <Attribute name="ConfigLocalPort">0</Attribute>
  <Attribute name="LocalPort">0</Attribute>
  <Attribute name="EffectiveLocalAddress"></Attribute>
  <Attribute name="EffectiveLocalInterface"></Attribute>
  <Attribute name="CongestionState">none</Attribute>
  <Attribute name="EffectiveRemoteAddress"></Attribute>
  <Attribute name="PrimaryRemoteAddress"></Attribute>
  <Attribute name="SCTPState">None</Attribute>
  <Attribute name="SLC">0</Attribute>

```

```

<Attribute name="InterfaceName">Serial0/0:0</Attribute>
<Attribute name="LinkType">Serial</Attribute>
<Attribute name="ReceiveUtilizationState">underThreshold</Attribute>
<Attribute name="SendUtilizationState">underThreshold</Attribute>
<Feature name="Basic">
  <AttributeGroup name="Interface">
    <Attribute name="InterfaceName">Serial0/0:0</Attribute>
  </AttributeGroup>
  <Attribute name="LinkType">MTP2</Attribute>
</Feature>
<Feature name="MTP2Timer">
  <Attribute name="MtT1Value">12500</Attribute>
  <Attribute name="MtT2Value">6000</Attribute>
  <Attribute name="MtT3Value">5000</Attribute>
  <Attribute name="MtT4nValue">2007</Attribute>
  <Attribute name="MtT4eValue">550</Attribute>
  <Attribute name="MtT5Value">90</Attribute>
  <Attribute name="MtT7Value">900</Attribute>
</Feature>
<Feature name="MTP2">
  <Attribute name="PCR">true</Attribute>
  <Attribute name="PCRN1">34</Attribute>
  <Attribute name="TxDepthValue">50</Attribute>
</Feature>
<Feature name="Description">
  <Attribute name="Description">Link Description</Attribute>
  <Attribute name="DisplayName">Link Name</Attribute>
</Feature>
<Feature name="CTParams">
  <Attribute name="CapacitySend">570000</Attribute>
  <Attribute name="ThresholdSend">7</Attribute>
  <Attribute name="ThresholdRcvd">7</Attribute>
</Feature>
<Feature name="LinkTimer">
  <Attribute name="LkT01Value">900</Attribute>
  <Attribute name="LkT02Value">900</Attribute>
  <Attribute name="LkT03Value">900</Attribute>
  <Attribute name="LkT04Value">900</Attribute>
  <Attribute name="LkT05Value">900</Attribute>
  <Attribute name="LkT12Value">900</Attribute>
  <Attribute name="LkT13Value">900</Attribute>
  <Attribute name="LkT14Value">2500</Attribute>
  <Attribute name="LkT17Value">800</Attribute>
  <Attribute name="LkT19Value">480000</Attribute>
  <Attribute name="LkT20Value">91111</Attribute>
  <Attribute name="LkT21Value">90000</Attribute>
  <Attribute name="LkT31Value">11111</Attribute>
  <Attribute name="LkT32Value">11111</Attribute>
  <Attribute name="LkSLTT01Value">4000</Attribute>
  <Attribute name="LkSLTT02Value">90000</Attribute>
  <Attribute name="LkRetryValue">90000</Attribute>
</Feature>
</ns2:NetworkElement>

```

Understanding Network Element Information Types

For all the network element attributes, the MWTM 6.1.3 Inventory API distinguishes two different information types:

- Configuration information

- Monitor information



Note There are certain network elements that do not have any monitor attributes. (MWTM 6.1.3 does not monitor the status of these network elements). These network elements are referred to as configuration network elements.

The following is an example of a tree that represents configured network element:

```
Node (subtype=ITP)
|
+-- SP*
|   |
|   +-- Linkset*
|       |
|       +-- Link*
|
|   +-- AS*
|       |
|       +-- ASPA
|
+-- SGMP*
|
+-- Interface T1/E1*
|   |
|   +-- Interface Serial*
|
+-- Interface FE*
|
+-- Interface ATM*
|
+-- Folder
|
+-- Folder
|
+-- ASP**
|
+-- LocalPeer**
|
+-- M3UA**
|
+-- SUA**
```



- Note**
- A network element with no markings indicates that it is a monitor network element (no configuration attributes).
 - A network element with an asterisk (*) indicates that it supports MWTM 6.1.3 provisioning. These network elements have both monitor and configuration attributes.
 - A network element with two asterisks (**) indicates that it supports MWTM 6.1.3 provisioning, and it is a configuration network element (no monitoring attributes).

The following sections describe these two different information types:

- [Configuration Information, page 2-13](#)
- [Monitor Information, page 2-14](#)

Configuration Information

Configuration information refers to those network element attributes that are usually configured on the device. This type of information is relatively static. The information usually changes only if you provision the device.

The following are example attributes that are considered configuration information for an ITP node:

```
<NetworkElement type="Node" subtype="ITP" ParentFQDN="ParentFQDN=" ">
  <Attribute name="RDN">sgm-75-92b.cisco.com</Attribute>
</NetworkElement>
```

MWTM 6.1.3 does not have node level provision capability. It only provisions Linksets, Links, Application Servers (AS), Application Server Processes (ASP) and interfaces. As such, an ITP node does not contain any configuration information other than the RDN.

The following is an example XML representation of an ITP link:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:NetworkElement xmlns:ns2="http://cisco.com/mwtm" type="Link"
parentFQDN="Node=sgm-26-96a.cisco.com,SP=ansi-config,Linkset=2696a_to_2696b">
  <Attribute name="RDN">0</Attribute>
  <Feature name="Basic">
    <AttributeGroup name="Interface">
      <Attribute name="InterfaceName">Serial0/0:0</Attribute>
    </AttributeGroup>
    <Attribute name="LinkType">MTP2</Attribute>
  </Feature>
  <Feature name="MTP2Timer">
    <Attribute name="MtT1Value">12500</Attribute>
    <Attribute name="MtT2Value">6000</Attribute>
    <Attribute name="MtT3Value">5000</Attribute>
    <Attribute name="MtT4nValue">2007</Attribute>
    <Attribute name="MtT4eValue">550</Attribute>
    <Attribute name="MtT5Value">90</Attribute>
    <Attribute name="MtT7Value">900</Attribute>
  </Feature>
  <Feature name="MTP2">
    <Attribute name="PCR">true</Attribute>
    <Attribute name="PCRN1">34</Attribute>
    <Attribute name="TxDepthValue">50</Attribute>
  </Feature>
  <Feature name="Description">
    <Attribute name="Description">Link Description</Attribute>
    <Attribute name="DisplayName">Link Name</Attribute>
  </Feature>
  <Feature name="CTParams">
    <Attribute name="CapacitySend">570000</Attribute>
    <Attribute name="ThresholdSend">7</Attribute>
    <Attribute name="ThresholdRcvd">7</Attribute>
  </Feature>
  <Feature name="LinkTimer">
    <Attribute name="LkT01Value">900</Attribute>
    <Attribute name="LkT02Value">900</Attribute>
    <Attribute name="LkT03Value">900</Attribute>
    <Attribute name="LkT04Value">900</Attribute>
    <Attribute name="LkT05Value">900</Attribute>
    <Attribute name="LkT12Value">900</Attribute>
    <Attribute name="LkT13Value">900</Attribute>
    <Attribute name="LkT14Value">2500</Attribute>
    <Attribute name="LkT17Value">800</Attribute>
    <Attribute name="LkT19Value">480000</Attribute>
    <Attribute name="LkT20Value">91111</Attribute>
    <Attribute name="LkT21Value">90000</Attribute>
  </Feature>
</ns2:NetworkElement>
```

```

    <Attribute name="LkT31Value">11111</Attribute>
    <Attribute name="LkT32Value">11111</Attribute>
    <Attribute name="LkSLTT01Value">4000</Attribute>
    <Attribute name="LkSLTT02Value">90000</Attribute>
    <Attribute name="LkRetryValue">90000</Attribute>
  </Feature>
</ns2:NetworkElement>

```

Monitor Information

Monitor information refers to those network element attributes that are not configured on the device, but their values could change depending on the network conditions. This type of information is relatively dynamic.

The following are example attributes that are considered monitor information for an ITP node:

```

<NetworkElement type="Node" subtype="ITP" ParentFQDN="ParentFQDN="">
  <Attribute name="RDN">sgm-75-92b.cisco.com</Attribute>
  <Attribute name="State">Warning</Attribute>
  <Attribute name="StateReason">Interface Down</Attribute>
  <Attribute name="StateTimestamp">2006-08-28T23:56:01.360-04:00</Attribute>
  <Attribute name="IgnoreState">false</Attribute>
  <Attribute name="DefaultIcon">Cisco7507</Attribute>
  <Attribute name="CustomIcon"><</Attribute>
  <Attribute name="CustomName"><</Attribute>
  <Attribute name="SysUpTime">P3DT6H40M47.550S</Attribute>
  <Attribute name="RebootReason">reload</Attribute>
  <AttributeGroup name="IPAddresses">
    <Attribute name="Addresses">172.18.17.35</Attribute>
    <Attribute name="Addresses">172.18.17.163</Attribute>
    <Attribute name="PollingTimestamps">2006-08-28T23:58:03.410-04:00</Attribute>
    <Attribute name="PollingTimestamps"><</Attribute>
    <Attribute name="SNMPFlags">1</Attribute>
    <Attribute name="SNMPFlags">2</Attribute>
    <Attribute name="Statuses">Active</Attribute>
    <Attribute name="Statuses">Active</Attribute>
    <Attribute name="PrimaryAddress">172.18.17.35</Attribute>
    <Attribute name="LastPolledAddress">172.18.17.35</Attribute>
  </AttributeGroup>
  <Attribute name="EnableProcessTraps">>true</Attribute>
  <Attribute name="DiscoveredTimestamp">2006-08-28T05:52:48.819-04:00</Attribute>
  <Attribute name="DeviceType">Cisco7507z</Attribute>
  <Attribute name="SysDescr">Cisco IOS Software, RSP Software (RSP-ITPK91V-M),
Experimental Version 12.2(2006
0825:064354) [stklein-topsail_s_nightly 104]
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Fri 25-Aug-06 06:26 by stklein</Attribute>
  <Attribute name="SysName">sgm-75-92b.cisco.com</Attribute>
  <Attribute name="LastPollTimestamp">2006-08-28T23:58:03.410-04:00</Attribute>
  <Attribute name="LastPollTimePeriod">PT17.096S</Attribute>
  <Attribute name="AveragePollTimePeriod">PT16.420S</Attribute>
  <Attribute name="TelnetAddress"><</Attribute>
  <Attribute name="MIBLevel">12.2(25)SW4</Attribute>
  <Attribute name="SerialNumber"><</Attribute>
  <Attribute name="EnableTrapPolling">false</Attribute>
  <Attribute name="EnableReportPolling">false</Attribute>
  <Attribute name="CLLIDCode">ccli_9572b</Attribute>
  <Attribute name="NSOConfig">None</Attribute>
  <Attribute name="RFState">Standby Hot</Attribute>
  <Attribute name="Mtp3Offload">Main</Attribute>
</NetworkElement>

```

The following are example attributes that are considered monitor information for an ITP link:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:NetworkElement xmlns:ns2="http://cisco.com/mwtm" type="Link"
parentFQDN="Node=sgm-26-96a.cisco.com,SP=ansi-config,Linkset=2696a_to_2696b">
  <Attribute name="RDN">0</Attribute>
  <Attribute name="State">Active</Attribute>
  <Attribute name="StateReason">None</Attribute>
  <Attribute name="StateTimestamp">2006-11-27T11:38:00.602-05:00</Attribute>
  <Attribute name="IgnoreState">>false</Attribute>
  <Attribute name="QOS">0</Attribute>
  <Attribute name="ConfigRemotePort">0</Attribute>
  <Attribute name="RemotePort">0</Attribute>
  <Attribute name="ConfigLocalPort">0</Attribute>
  <Attribute name="LocalPort">0</Attribute>
  <Attribute name="EffectiveLocalAddress"></Attribute>
  <Attribute name="EffectiveLocalInterface"></Attribute>
  <Attribute name="CongestionState">none</Attribute>
  <Attribute name="EffectiveRemoteAddress"></Attribute>
  <Attribute name="PrimaryRemoteAddress"></Attribute>
  <Attribute name="SCTPState">None</Attribute>
  <Attribute name="SLC">0</Attribute>
  <Attribute name="InterfaceName">Serial0/0:0</Attribute>
  <Attribute name="LinkType">Serial</Attribute>
  <Attribute name="ReceiveUtilizationState">underThreshold</Attribute>
  <Attribute name="SendUtilizationState">underThreshold</Attribute>
</ns2:NetworkElement>
```

For a complete list of MWTM 6.1.3 network element attributes, see [Appendix E, “MWTM Monitor Attributes.”](#)

MWTM 6.1.3 Inventory API Operations

This section lists the MWTM 6.1.3 Inventory API Operations.



Note

- All the operations are listed as pseudocode with comments. Syntax for these operations is defined in Web Service Definition Language (WSDL). This syntax is described in [Appendix A, “MWTM NAPI WSDL and XSD Definitions.”](#)
- All MWTM 6.1.3 Inventory API operations use the SOAP fault for error handling. The error codes are defined in [Appendix B, “MWTM NAPI Error Codes.”](#)
- MWTM 6.1.3 Inventory API only retrieves information from the MWTM 6.1.3 internal data model. It does not modify or change inventory objects. The MWTM 6.1.3 Provision API provides methods to provision the device and change inventory objects (see [Chapter 4, “MWTM 6.1.3 Provision API”](#)).

Get All Network Elements from MWTM

```
NetworkElementList getAllNEs (InventoryContext)
```

This method retrieves all the inventory objects from MWTM 6.1.3.

Parameters

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information.

Return Value

A list of all the network elements in MWTM 6.1.3.

Get All Network Elements With Feature from MWTM

```
NetworkElementList getAllNEsWithFeature (String featureStr, InventoryContext)
```

This method retrieves all the inventory objects with the given feature from MWTM 6.1.3.

Parameters

String featureStr—Specifies the Feature

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information

Return Value

A list of all the network elements in MWTM 6.1.3.

Get Root Network Elements from MWTM

```
NetworkElementList getRootNEs (InventoryContext)
```

This method retrieves the root inventory objects from MWTM 6.1.3. Root objects can be of type Node, RanBackhaul, APN, or Group.

Parameters

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information.

Return Value

A list of all the network elements in MWTM 6.1.3.

Get Root Network Elements With Feature from MWTM

```
NetworkElementList getRootNEsWithFeature (String featureStr, InventoryContext)
```


This method retrieves the root inventory objects with the given feature from MWTM 6.1.3.

Parameters

String featureStr— Specifies the Feature

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information

Return Value

A list of all the network elements in MWTM 6.1.3.

Get One Network Element from MWTM

```
NetworkElementList getNE (String FQDN, InventoryContext)
```

This method retrieves one network element object from MWTM 6.1.3.

Parameters

String FQDN—Fully qualified domain name

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information.

Return Value

The network elements in MWTM 6.1.3 for given FQDN.

Get Child Network Elements from MWTM

```
NetworkElementList getChildNEs (String FQDN, InventoryContext)
```

This method retrieves a list of children network element objects from MWTM 6.1.3.

Parameters

String FQDN—Fully qualified domain name

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information.

Return Value

A list of child network elements in MWTM 6.1.3 for a given FQDN.

Get Child Network Elements With Feature from MWTM

```
NetworkElementList getChildNESWithFeature (String FQDN, String featureStr,  
InventoryContext)
```

This method retrieves a list of children network element objects with the given feature from MWTM 6.1.3

Parameters

String FQDN—Fully qualified domain name

String featureStr—Specifies the Feature

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information

Return Value

A list of all the network elements in MWTM 6.1.3.

Get Descendant Network Elements from MWTM

```
NetworkElementList getDescendentNES (String FQDN, InventoryContext)
```

This method retrieves a list of descendants network element objects from MWTM 6.1.3.

Parameters

String FQDN—Fully qualified domain name

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information.

Return Value

A list of descendant network elements in MWTM 6.1.3 for a given FQDN.

Get Descendant Network Elements With Feature from MWTM

```
NetworkElementList getDescendantNESWithFeature (String FQDN, String featureStr,  
InventoryContext)
```

This method retrieves a list of descendants network element objects with the given feature from MWTM 6.1.3.

Parameters

String FQDN—Fully qualified domain name

String featureStr—Specifies the Feature

InventoryContext—You can specify the inventory context you want to retrieve:

- All device information.
- Device configuration information.
- Device monitor information

Return Value

A list of all the network elements in MWTM 6.1.3.

Get Note for an Inventory Object

```
String getNote (String FQDN)
```

This method gets attached note for an inventory object.

Parameters

String FQDN—Inventory FQDN to retrieve the note

Return Value

Note string.

Set Note for an Inventory Object

```
String setNote(String FQDN, String userid, String note)
```

This method sets the attached note for an inventory object.

Parameters

String FQDN—Inventory FQDN to retrieve the note

String userid—User ID who sets the note

String note—Note text to set

Return Value

None

Append Note to an Inventory Object

```
String setNote (String FQDN, String userid, String note)
```

This method appends the note to an inventory object.

Parameters

String FQDN—Inventory FQDN to retrieve the note

String userid—User ID who appends text to the inventory note

String note—Text to append to the inventory note

Return Value

None

Ignore an Inventory Object

`String ignoreNE (String FQDN)`

This method ignores an inventory object.

Parameters

String FQDN—Inventory FQDN to set as ignored.

Return Value

Status of the operation: either *successful* or an error message.

Unignore an Inventory Object

`String unIgnoreNE (String FQDN)`

This method unignores an inventory object.

Parameters

String FQDN—Inventory FQDN to set as unignored.

Return Value

Status of the operation: either *successful* or an error message.