MWMT 6.1.1 Provision API

The MWMT 6.1.1 Provision API allows the Operations Support and System (OSS) to provision the IP Transfer Point (ITP) Linkset, Link, Application Server (AS), and ASP using a programming interface.

The MWMT 6.1.1 provides Remote Procedure Call (RPC) style Inventory API to the Northbound OSS. The OSS can send Simple Object Access Protocol (SOAP) requests to the MWMT 6.1.1 Inventory API, and the MWMT 6.1.1 responds with SOAP responses.

This chapter includes the following sections:

- Setting up MWMT 6.1.1 to Retrieve Config Attributes from a Device, page 4-2
- Issuing Provision Requests, page 4-3
- Provision Request Logging, page 4-6
- MWMT 6.1.1 Provision API Operations, page 4-7

A block diagram illustrating the MWMT 6.1.1 Provision system is shown in Figure 4-1.
As shown in the preceding block diagram, the MWTM 6.1.1 provides the Provision API to the Northbound OSS.

The OSS can send a SOAP request to the MWTM 6.1.1 Provision API. The MWTM 6.1.1 attempts to provision the device from the SOAP request. It then responds with SOAP responses.

The MWTM 6.1.1 Provision API is defined in the Web Service Definition Language (WSDL)\(^1\). It defines communication protocol and message exchanges between two remote applications.

### Setting up MWTM 6.1.1 to Retrieve Config Attributes from a Device

The MWTM 6.1.1 Inventory API provides two types of attributes (see Chapter 2, “MWTM 6.1.1 Inventory API,” for more detailed information):

- monitor attributes
- config attributes

**Monitor Attributes**

The Monitor attributes are those attributes obtained from the Simple Network Management Protocol (SNMP) polling/status monitoring.

**Config Attributes**

The Config attributes are those attributes obtained from the IOS running-config. The MWTM 6.1.1 provision feature requires the MWTM 6.1.1 to successfully retrieve the config attributes from a device.

To retrieve the config attributes from the device, you must supply the device credentials for the target device. You also need to ensure that the MWTM 6.1.1 is getting the IOS running-config from the device successfully.

### Setting up Device Credentials

You can specify the device credentials for all the nodes in an MWTM 6.1.1, or specify the credentials for an individual node (see the User Guide for the Cisco Mobile Wireless Transport Manager 6.1.1 on how to specify credentials in MWTM 6.1.1).

### Getting IOS running-config from a Device

There are two ways the MWTM 6.1.1 can get an IOS running-config from a device:

- Automatic config sync
- Manual config sync

**Automatic Config Sync**

By default, the MWTM 6.1.1 is installed with the automatic config sync option turned on. This option is specified by the AUTO_CONFIG_SYNC property in this file:

\(^1\) WSDL is a W3C recommendation (see http://www.w3.org/TR/wsdl).
When the automatic config sync option is turned on, the MWTM 6.1.1 will try to get the IOS running-config from the device the following three ways:

- During every status poll, the MWTM 6.1.1 checks to see whether the running-config has changed on the device. If it has changed, then the MWTM 6.1.1 retrieves the running-config from the device.
- After the MWTM 6.1.1 processes a provision operation (from NBAPI or GUI), it automatically retrieves the running-config from the device.

When the automatic config sync option is turned off, then the MWTM 6.1.1 will not try to get the running-config from the device in the previous scenarios.

**Manual Config Sync**

You might want to perform a manual config sync in certain scenarios. For example, you might turn off the automatic config sync and choose to manage the config sync from the device manually.

You can request a config sync from the device manually via the Northbound API (NBAPI), or via the Command Line Interface (CLI) tool. To request a config sync via the NBAPI, see the “Sync from Device” section on page 4-6. To request a config sync via the CLI tool, see Chapter 8, “MWTM NBAPI CLI Tools.”

**Issuing Provision Requests**

The Northbound OSS can issue a provision request via an Extensible Markup Language (XML) based provision request. A provision request is defined in Web Service Definition Language/XML Schema Definition (WSDL/XSD). See Appendix A, “MWTM NBAPI WSDL and XSD Definitions.”

A provision request contains one or more operations. MWTM 6.1.1 executes this list of provision operations sequentially. These operations are described in the following sections:

- Specifying Provision Operations, page 4-3
- IOS Write to Startup, page 4-6
- Sync from Device, page 4-6

**Specifying Provision Operations**

A provision operation operates on one network element (NE). On the network element, you can specify the following operations:

- ADD—See ADD Operation, page 4-3
- DELETE—See MODIFY Operation, page 4-5
- MODIFY—See DELETE Operation, page 4-6

**ADD Operation**

In the ADD operation, you need to specify the ParentFQDN, type, and Relative Distinguished Name (RDN) attribute to identify the network element to add. You also need to specify any features to be added for this network element.
A “Basic” feature is required for any provisionable NE.

The following is an example XML script for an ADD operation:

```xml
<ProvisionOperation type="Link" operation="add"
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="MtT4Value">2007</Attribute>
    <Attribute name="MtT5Value">550</Attribute>
    <Attribute name="MtT7Value">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="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">490000</Attribute>
    <Attribute name="LkT20Value">92111</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>
</ProvisionOperation>
```
MODIFY Operation

In a MODIFY operation, you need to specify the ParentFQDN, type, and RDN attribute to identify the network element to modify. You also need to specify all the necessary features for this NE.

You might use the MODIFY operation request to add/delete/modify features. When processing a MODIFY operation request, the MWTM 6.1.1 compares the list of features specified in the Northbound API (NBAPI) against the list of features on the target device:

- If a feature does not exist on the device, but exists in the NBAPI request, then the MWTM 6.1.1 adds this feature to the target device;
- If a feature exists on the device, but does not exist in NBAPI request, then the MWTM 6.1.1 deletes this feature from target device;
- If a feature exists in both places, then the MWTM 6.1.1 compares whether the attributes in these two features are the same. If they are different, then the MWTM 6.1.1 modifies this feature on target device.

Note

A “Basic” feature is required for any provisionable NE.

The following is an example XML script for a MODIFY operation:

```xml
<ProvisionOperation type="Link" operation="modify" 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">12501</Attribute>
    <Attribute name="MtT2Value">6001</Attribute>
    <Attribute name="MtT3Value">5001</Attribute>
    <Attribute name="MtT4nValue">2006</Attribute>
    <Attribute name="MtT4eValue">551</Attribute>
    <Attribute name="MtT5Value">91</Attribute>
    <Attribute name="MtT7Value">901</Attribute>
  </Feature>
  <Feature name="Description">
    <Attribute name="Description">New Link Description</Attribute>
    <Attribute name="DisplayName">New Link Display Name</Attribute>
  </Feature>
  <Feature name="CTParams">
    <Attribute name="CapacitySend">570000</Attribute>
    <Attribute name="ThresholdSend">1</Attribute>
    <Attribute name="ThresholdRcvd">1</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>
</ProvisionOperation>
```
DELETE Operation

In a DELETE operation, you need to specify the ParentFQDN, type, and RDN attribute to identify the network element to delete from the device. You do not need to specify any features on the NE.

The following is an example XML script for a DELETE operation:

```xml
<ProvisionOperation type="Link" operation="delete"
   parentFQDN="Node=sgm-26-96a.cisco.com,SP=ansi-config,Linkset=2696a_to_2696b">
   <Attribute name="RDN">0</Attribute>
</ProvisionOperation>
```

IOS Write to Startup

This operation requests the IOS device to save the running-config to the startup-config on the device. You must specify a node to perform this operation on.

The following is an example XML script for the IOS write to startup operation:

```xml
<IOSWriteToStartup FQDN="Node=sgm-26-96a"></IOSWriteToStartup>
```

Sync from Device

This operation syncs the IOS running-config from the device into the MWTM 6.1.1 repository. You must specify a node to perform this operation.

The following is an example XML script for the sync from the device operation:

```xml
<SyncFromDevice FQDN="Node=sgm-26-96a"></SyncFromDevice>
```

Provision Request Logging

The MWTM 6.1.1 provision request log is kept at:

```
$(MWTM_install_dir)/logs/provisionLogs/
```

The log file names are identified by the provision request ID.

The following is an example log file of /opt/CSCOsgm/logs/provisionLogs/Provision_1647152932250386401.log:

```
ADD [started]: Node=sgm-26-96b.cisco.com,SP=ansi-config,Linkset=TST0,Link=5
```
MWTM 6.1.1 Provision API Operations

This section lists the MWTM 6.1.1 Provision API operations.

All the operations are listed as pseudo-code with comments. Syntax for these operations is defined in Web Service Definition Language/XML Schema Definition (WSDL/XSD). See Appendix A, “MWTM NBAPI WSDL and XSD Definitions.”

All MWTM 6.1.1 Event API operations use the SOAP fault for error handling. The error codes are defined in Appendix B, “MWTM 6.1.1 NBAPI Error Codes.”

Process Provision Request

long provision(ProvisionRequest request)

This method retrieves all the events from MWTM 6.1.1 as traps.

Parameters:

ProvisionRequest request—You can specify the provision request (see Issuing Provision Requests, page 4-3 for a detailed description of the provision request).
Return Value:

Provision request ID.