



CISCO CONFIDENTIAL

## CHAPTER 27

# Using Product Instance Device Mapping

Product Instance Device Mapping (PIDM) is a registry that stores mapping information between devices and applications.

Applications need to register with PIDM the information about the devices managed by the applications. Applications can do this using PIDM APIs (which are also available via CSTM from Common Services 3.0 Service Pack 2).

Device Center depends on this information for showing the devices in Device Selector and also for showing the information under Tools , Management Tasks , Reports and Summary sections.

Device Center usually shows only the devices managed by the applications in the Device Selector. On selecting a device, the links corresponding to the applications that manage the device are shown in Device Center.

The following topics describe how to integrate DCR with your application:

- [Using the PIDM APIs](#)
- [Using the PIDM North-bound APIs](#)

## Using the PIDM APIs

The PIDM APIs are as follows:

**Table 27-1** PIDM APIs

APIs	Attributes
<code>com.cisco.nm.dcr.AppId[]</code> <code>getMappedAppIDs(com.cisco.nm.dcr.DeviceId deviceID)</code>	Returns the managing application's IDs for the given device.
<code>com.cisco.nm.dcr.AppId[]</code> <code>getMappedAppIDs(com.cisco.nm.dcr.DeviceId deviceID)</code>	Returns the deviceIDs for the devices managed by the given application.
<code>void mapDevicesToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId[] deviceIDs)</code>	Marks given devices as 'managed by' given application.
<code>void mapDeviceToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId deviceID)</code>	Marks given device as 'managed by' given application.

**CISCO CONFIDENTIAL****Table 27-1** PIDM APIs

APIs	Attributes
void unmapDevicesToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId[] deviceIDs)	Removes the mapping between given application and devices.
void unmapDeviceToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId deviceID)	Removes the mapping between given application and device.
void close()	Closes the pidm proxy.

## Creating the ProductToDeviceMapProxy Object

```
ProductToDeviceMapProxy pidm = new ProductToDeviceMapProxy();
```

## Mapping a Device or Marking a Device(s) as Managed

```
DeviceId devId = new DeviceId("101");
AppId Myapp = new AppId("Campus Manager", "4.0", "bundle-pc10");
try {
    pidm.mapDeviceToProduct(Myapp, devId);
} catch (PDMEException pdmExp) {
    System.out.println("Error in mapping a Device " + pdmExp.getMessage());
}
```

## Unmapping a Device or Marking a Device(s) as Not Managed

```
DeviceId devId = new DeviceId("101");
AppId Myapp = new AppId("Campus Manager", "4.0", "bundle-pc10");
try {
    pidm.unmapDeviceToProduct(Myapp, devId);
} catch (PDMEException pdmExp) {
    System.out.println("Error in un-mapping a Device " + pdmExp.getMessage());
}
```

## Retrieving PIDM Information

To get the applications managing a particular device:

```
DeviceId devId = new DeviceId("101");
AppId appIds[] = pdm.getMappedAppIDs(devId);
// To get devices managed by a particular application.
DeviceId devIds[] = pdm.getMappedDeviceIDs(Myapp);
```

**CISCO CONFIDENTIAL**

# Using the PIDM North-bound APIs

PIDM APIs are also available via CSTM from Common Services 3.0 Service Pack 2.

## PIDM North-bound APIs

The PIDM North-bound APIs are as follows:

**Table 27-2** PIDM North-bound APIs

APIs	Attributes
com.cisco.nm.dcr.AppId[] getMappedAppIDs(com.cisco.nm.dcr.DeviceId deviceID, com.cisco.nm.dcr.APIExtraInfo apiExtraInfo)	Returns the managing application IDs for the given device.
com.cisco.nm.dcr.DeviceId[] getMappedDeviceIDs(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.APIExtraInfo apiExtraInfo)	Returns the device IDs for the devices managed by the given application.
void mapDevicesToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId[] deviceIDs, com.cisco.nm.dcr.APIExtraInfo apiExtraInfo)	Marks given devices as managed by given application.
void mapDeviceToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId deviceID, com.cisco.nm.dcr.APIExtraInfo apiExtraInfo)	Marks given device as managed by given application.
void unmapDevicesToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId[] deviceIDs, com.cisco.nm.dcr.APIExtraInfo apiExtraInfo)	Removes the mapping between given application and devices.
void unmapDeviceToProduct(com.cisco.nm.dcr.AppId appID, com.cisco.nm.dcr.DeviceId deviceID, com.cisco.nm.dcr.APIExtraInfo apiExtraInfo)	Removes the mapping between given application and device.
void close()	Closes the PIDM proxy.

## PIDM NB APIs and Associated Tasks

The PIDM NB APIs and associated tasks are as follows:

**Table 27-3** PIDM NB APIs and Associated Tasks

API	DCR Task	ACS Task
getMappedDeviceIDs	VIEW_DEVICE_TASK	View
getMappedAppIDs	VIEW_DEVICE_TASK	View
mapDeviceToProduct	ADD_DEVICE_TASK	Add
mapDevicesToProduct	ADD_DEVICE_TASK	Add

**CISCO CONFIDENTIAL****Table 27-3** *PIDM NB APIs and Associated Tasks*

API	DCR Task	ACS Task
unmapDeviceToProduct	DELETE_DEVICE_TASK	Delete
unmapDevicesToProduct	DELETE_DEVICE_TASK	Delete

## Creating the APIExtraInfo Object

ProductToDeviceMapNBProxy API accepts objects of class APIExtraInfo in addition to their regular arguments. APIExtraInfo encapsulates:

- The AppID - This object is the unique ID of your application, and establishes your application name, version number, and host information for use in the SourceContext object.
- The SourceContext - This object establishes your application and its context as the source of the ProductToDeviceMapNBProxy API call.
- The SecurityContext - This object contains the information needed to authenticate or authorize the ProductToDeviceMapNBProxy API request.

```
AppId Myapp = new AppId("Device Manager", //unique AppID and application name
    "1.3.2", //application version number
    "192.168.1.15"); // host on which the application is running

SourceContext source = new SourceContext(Myapp) ;
// For Local API calls
SecurityContext security = new SecurityContext("username");

// For North-Bound API calls
SecurityContext security = new SecurityContext("username",
    "password",
    "secretKey")
//Wrapper for Source and Security information:
APIExtraInfo extraInfo = new APIExtraInfo(security, source);
```

## Creating the ProductToDeviceMapNBProxy Object

```
ProductToDeviceMapNBProxy pidmNB = new ProductToDeviceMapNBProxy();
```

## Mapping a Device or Marking a Device(s) as Managed

```
DeviceId devId = new DeviceId("101");
AppId Myapp = new AppId("Campus Manager", "4.0", "bundle-pc10");

try {
    pidmNB.mapDeviceToProduct(Myapp, devId, extraInfo);
}
catch (PDMEException pdmExp) {
    System.out.println("Error in mapping a Device " + pdmExp.getMessage());
}
```

**CISCO CONFIDENTIAL**

## Unmapping a Device or Marking a Device(s) as Not Managed

```
DeviceId devId = new DeviceId("101");
AppId Myapp = new AppId("Campus Manager", "4.0", "bundle-pc10");

try {
    pidmNB.unmapDeviceToProduct(Myapp, devId, extraInfo);
}
catch (PDMEException pdmExp) {
    System.out.println("Error in un-mapping a Device " + devId + " : " + pdmExp.getMessage());
}
```

## Retrieving PIDM Information

To get the applications managing a particular device:

```
DeviceId devId = new DeviceId("101");
AppId appIds[] = pidmNB.getMappedAppIDs(devId, extraInfo);
```

To get devices managed by a particular application.

```
DeviceId devIds[] = pidmNB.getMappedDeviceIDs(Myapp, extraInfo);
```

***CISCO CONFIDENTIAL***