MyID

Version 11.7

Device Management API

Lutterworth Hall, St Mary's Road, Lutterworth, Leicestershire, LE17 4PS, UK
www.intercede.com | info@intercede.com | @intercedemyid | +44 (0)1455 558111

Document reference: IMP1950-08 September 2020

Copyright

Whilst Intercede Limited has made every effort in the preparation of this manual to ensure the accuracy of the information, the information contained in this manual is delivered without warranty, either express or implied. Intercede Limited will not be held liable for any damages caused, or alleged to be caused, either directly or indirectly by this manual.

Licenses and Trademarks

Conventions Used in this Document

- Lists:
  - Numbered lists are used to show the steps involved in completing a task when the order is important
  - Bulleted lists are used when the order is unimportant or to show alternatives

- **Bold** is used for menu items and for labels.

  For example:

  - "Record a valid email address in **'From' email address**"
  - Select **Save** from the **File** menu

- *Italic* is used for emphasis and to indicate references to other sections within the current document:

  For example:

  - "Copy the file *before* starting the installation"
  - "See *Issuing a Card* for further information"

- ***Bold and italic*** are used to identify the titles of other documents.

  For example: "See the ***Release Notes*** for further information."

  Unless otherwise explicitly stated, all referenced documentation is available on the product media.

- A `fixed width` font is used where the identification of spaces is important, including filenames, example SQL queries and any entries made directly into configuration files or the database.

- **Notes** are used to provide further information, including any prerequisites or configuration additional to the standard specifications.

  For example:

  Note: This issue only occurs if updating from a previous version.

- Warnings are used to indicate where failure to follow a particular instruction may result in either loss of data or the need to manually configure elements of the system.

  For example:

Warning: You must take a backup of your database before making any changes to it.

Contents

1 Introduction

The Device Management API is a mechanism that allows server to server management of devices from external sources within MyID®. These methods contain no MyID authentication and rely instead on platform authentication such as mutual SSL on IIS.

1.1 WSDL

You can obtain the WSDL for the web service by browsing to:

```
http://myserver.example.com/DeviceManagementAPI/DeviceAPI.svc?singleWsdl
```

where `myserver.example.com` is the name of the server on which you have installed the Device Management API.

1.2 System architecture

The Device Management API is written as a WCF service in C#, with the intention that it is to be hosted on an IIS server with very restrictive access.

To enable this API, select the **Device Management API** option when installing MyID.

You can install the web service on any server that has DCOM proxies that link it to the MyID application server (for example, the MyID web server, or the MyID web services server) – for information on setting up DCOM proxies, see the MyID ***Installation and Configuration Guide***.

1.3 Web server security

See the ***System Security Checklist*** for details of setting up security on MyID web services. You must set up security on the Device Management API web service in the same way as the other MyID server-to-server web services.

1.4 Change history

Version	Description
`IMP1950-01`	Released with MyID 11.0.
`IMP1950-02`	Released with MyID 11.1.
`IMP1950-03`	Released with MyID 11.2.
`IMP1950-04`	Released with MyID 11.3.
`IMP1950-05`	Released with MyID 11.4.
`IMP1950-06`	Released with MyID 11.5.
`IMP1950-07`	Released with MyID 11.6.
`IMP1950-08`	Released with MyID 11.7.

2 Naming

2.1 Terminology

Name	Description	Examples
Person	A user account within MyID, that will require issuance of credentials to represent their identity.	Employees, system administrators, MyID operators.
Device	A physical entity (typically with some form of computer processor) that will require issuance of credentials to represent its identity. It may comprise of one or more Device Elements. These items are held within the `Carriers` database table in MyID.	Computers, mobile phones, tablets, routers, firewalls.
Device Identity	A collection of credentials that represent the identity of a device.	Authentication certificates used to prove the identity of a computer accessing a network.
Person Identity	A collection of credentials that represent the identity of a person.	Digital certificates used for: • authentication of the person. • signing or encryption/decryption of data by the person. A printed Identity badge. One time passwords generated for authentication to a VPN.
Device Element	A distinct part of a device that may hold credentials. The type of device element to be used is generally managed within a credential profile. These items are held within the `Devices` database table in MyID.	Smart card. OTP Token. Software Certificates. Microsoft Virtual Smart Card. Trusted Platform Module. Secure Elements, such as a UICC SIM card in a mobile phone.
Profile	A definition within MyID of the credentials to be issued, the device element to be used, the lifetime of the credentials, issuance process to be used and access permissions to the profile. These are managed within the **Credential Profiles** workflow in MyID.	See the MyID ***Administration Guide*** for further details about configuring profiles.
Credentials	Information to help prove the identity or provide access to the holder of the credential.	Digital certificates, Authentication services (such as One Time Password), printed identity badges.
Job	An action within MyID that can be executed at a later date. Jobs may require validation before they are allowed to be actioned.	Card Issuance Job, Card Cancellation Job, Replacement Card Job.

2.2 Naming conventions

The naming convention for classes are:

- `[Name]` – Enough information to uniquely identify an entity of type `[Name]`.
- `[Name]Details` – Enough information to register a new entity of type `[Name]`.
- `[Name]Response` – Details returned from the server about the consequences of what just happened.
- `[Name]s` – a collection of entities of type `[Name]`.

3 Interface definition

3.1 AddDevice

```
Device AddDevice (
  DeviceDetails device,
  UserAccount deviceOwningUserAccount );
```

`AddDevice` is used to register a Device with MyID. This Device is likely to be a laptop or desktop computer, a mobile phone or tablet, or an appliance such as a router or firewall. You can optionally specify an owner for the Device.

If the Device already exists, attempting to add it again will fail.

3.1.1 Inputs

Class	Field	Data Type	Description	Allow Null?
`DeviceDetails`			Describes the Device being added.	No
	`SerialNumber`	String	A value to identify the Device uniquely. If not supplied, a GUID will be generated for the Device.	Yes
	`Type`	String	The category of the device; for example `Workstation`, `mobile`, `Appliance`. If you are importing mobile devices, you must use type `mobile`. If not supplied, it will default to `Asset`.	Yes
	`Description`	String	A text description of the Device.	Yes
	`DNS`	String	The DNS entry for the device on the network. Mandatory.	No
	`DN`	String	The DN for the device. If left blank this will be constructed from the DNS entry.	Yes
	`Active`	Boolean	Is the Device currently active? If blank defaults to false. Setting this to false will prevent it from being used in new requests.	Yes
	`Model`	String	The model of the device.	Yes
	`OS`	String	The operating system of the device.	Yes
	`Fields`		A collection of additional fields describing the Device. The Fields is a List of the type `ExtendedField`. `ExtendedField` contains two strings – `Name` and `Value`.	Yes

`UserAccount`			Identifies the virtual User Account that the Device will belong to. If null, the Device will be assigned to the default virtual device User Account,	Yes
	`LogonName`	String	The identifier for the system account that will own the Device.	No

Note: To add a device that can be used with a credential profile that is set for Known Mobiles only, you must provide the following information:

- `SerialNumber` – the serial number for the mobile device.
- `Type` – must be `mobile`.
- `DNS` – any.
- `UserAccount` – include a `LogonName` to specify the device owner.

3.1.2 Output

Class	Field	Data Type	Description	Allow Null?
`Device`			Describes the Device that has been added.	No
	`SerialNumber`	String	A value to identify the Device uniquely. If not supplied, a GUID will be generated for the Device.	Yes
	`Type`	String	The category of the device, for example "Workstation", "Tablet", "Phone".	Yes
	`DNS`	String	The DNS entry for the device on the network.	Yes

3.2 RequestDeviceIdentity

```
ProfileRequestResponse RequestDeviceIdentity(
  ProfileRequest profileRequest,
  Device device);
```

Creates a job to issue a Device Identity to a specified Device. The Device Identity is determined by the Device (which must already have been added).

3.2.1 Inputs

Class	Field	Data Type	Description	Allow Null?
`ProfileRequest`			Parameters defining the credentials to be requested.	No
	`ProfileName`	String	The name of the credential profile that the Device Identity is to receive. Profiles are defined in MyID using the **Credential Profiles** workflow. The latest version of the specified profile will be used.	No

Class	Field	Data Type	Description	Allow Null?
`ProfileRequest`			Parameters defining the credentials to be requested.	No
	`ExplicitExpiryDate`	DateTime	If present, the Device Identity will expire on the specified date. It is not possible for this to extend the life of a Device Identity beyond its profile value. This is currently not supported.	Yes
	`JobLabel`	String	If present, this will be passed through to the Job and can be used to search for the job.	Yes
`Device`			Identify the Device that will receive the Device Identity. To identify the device, you must specify either the `SerialNumber` and `Type`, or the `DNS`.	No
	`SerialNumber`	String	A value to identify the Device uniquely.	Yes
	`Type`	String	The category of the device; for example "Workstation", "Tablet", "Phone".	Yes
	`DNS`	String	The DNS entry for the device on the network.	Yes

3.2.2 Output

Class	Field	Data Type	Description	Allow Null?
`ProfileRequestResponse`			Reports the details of the Job created.	No
	`JobID`	Integer	The MyID identifier for the request.	No
	`JobStatus`	String	Either "Awaiting Issue" or "Awaiting Validation", depending on whether the requested credential profile requires a validation step.	No

3.3 CancelDevice

```
DeviceCancellationResponse CancelDevice(
  Device deviceToCancel,
  DeviceStatusChange deviceStatusChange);
```

Used to revoke a Device and all associated Device Elements, Device Identities, and Person Identities linked to that Device. Certificates on all associated devices will be revoked, and external systems notified of the action.

3.3.1 Inputs

Class	Field	Data Type	Description	Allow Null?
`Device`			Identify the Device to cancel. You can use either the `SerialNumber` and `Type`, or the DNS, to identify the Device	No
	`SerialNumber`	String	A value to identify the Asset uniquely.	Yes
	`Type`	String	Used to help identify Asset of different types but with identical serial numbers	Yes
	`DNS`	String	The DNS entry for the Asset on the network.	Yes
`DeviceStatusChange`				No
	`CancellationReasonID`	Integer	The ID for the reason the Asset is to be cancelled. These are available from the `StatusMapping` table within MyID. Some sample values are: `0` Unspecified or Automated Processes `1` Lost `2` Damaged `3` Stolen `4` Forgotten `5` Permanently Blocked `6` Compromised	No
	`Comment`	String	A free text field that is added to the MyID audit.	Yes
	`JobLabel`	String	If present, this will be passed through to the Job and can be used to search for the job.	Yes

Class	Field	Data Type	Description	Allow Null?
	DisposalStatus	String	If present, this will set the status of the device when it is cancelled; if not present, it will use the default value of Unassigned. Valid values are: None Collected Disposed Legacy Lost Not Disposed	Yes

Note: Not Disposed is the internal value. In MyID Desktop, this appears as **Not Collected**.

3.3.2 Output

Class	Field	Data Type	Description	Allow Null?
DeviceCancellation Response				No
	DeviceElement RevocationResponses		A list of DeviceElementRevocationResponse objects, one per Device that was revoked as a consequence of this action.	No

4 Error Messages

The following table lists the error messages that appear, and the requests that may cause them.

Message	Request
An unknown error has occurred	Any
Passing a UserAccount to AddDevice is not currently implemented	`AddDevice`
AddDevice failed	`AddDevice`

The device must specify a DNS or SerialNumber	`CancelDevice,`
	`RequestDeviceIdentity`

No device was found	`RequestDeviceIdentity`
More than one device was found, please make your criteria more specific	`RequestDeviceIdentity`
Card profile is incompatible for the device	`RequestDeviceIdentity`
Licence exceeded requesting device identity	`RequestDeviceIdentity`
The specified CancellationReasonID is not valid.	`CancelDevice`
The specified DisposalStatus is not valid.	`CancelDevice`

5 Data Types

When passing optional parameters into the API it is quite forgiving when the data type is a `String` but for other data types it is not so forgiving.

The following examples pass an empty string as the parameter value:

```
<JobLabel/>
```

```
<JobLabel></JobLabel>
```

If the intention is to pass a null value then the node must be omitted entirely.

The following example will generate an error as the `ExplicitExpiryDate` is a data type `DateTime` which cannot accept an empty string so must contain a valid date or not be included.

```
<RequestDeviceIdentity>
  <ProfileRequest>
          <ProfileName>My Credential Profile</ProfileName>
          <ExplicitExpiryDate/>
          <JobLabel/>
  </ProfileRequest>
  <Device>
          <DNS>my.dns.local.com</DNS>
          <SerialNumber></SerialNumber>
          <Type/>
  </Device>
</RequestDeviceIdentity>
```

The following is a valid example with the optional nodes missing entirely.

```
<RequestDeviceIdentity>
  <ProfileRequest>
          <ProfileName>My Credential Profile</ProfileName>
  </ProfileRequest>
  <Device>
          <DNS>my.dns.local.com</DNS>
  </Device>
</RequestDeviceIdentity>
```