

# MDChange REST API Documentation

---

Configure, manage and use MDChange Rest APIs

*Midrange Dynamics*

Copyright © 2023 Midrange Dynamics

## Table of contents

---

1. MDCMS REST API	4
1.1 MDChange REST API	4
1.2 REST API Catalogue	6
1.2.1 Applications REST API	6
1.2.2 Attributes REST API	7
1.2.3 Levels REST API	8
1.2.4 Level Wildcard REST API	9
1.2.5 MDSEC User REST API	11
1.2.6 Object Request REST API	13
1.2.7 Open Projects REST API	19
1.2.8 Open RFPs REST API	21
1.2.9 Open Subtasks REST API	23
1.2.10 Open Tasks REST API	24
1.2.11 Pipeline Request Trigger REST API	25
1.2.12 Project REST API	27
1.2.13 RFP Acceptance REST API	30
1.2.14 RFP Approval REST API	34
1.2.15 RFP Installation REST API	37
1.2.16 RFP Rollback REST API	40
1.2.17 RFP Send REST API	42
1.2.18 RFP Submission REST API	46
1.2.19 Task REST API	49
1.3 Configure the MDCMS REST API Server	53
1.3.1 Overview	53
1.3.2 Configure the MDCMS REST Server	53
1.4 Authenticate Requests to the MDCMS REST API Server	57
1.4.1 Overview	57
1.4.2 Generate a Token	57
1.4.3 Manage Existing Tokens	57
1.4.4 Example Request Header	57
1.5 Setting Up SSL on IBM i	58
1.5.1 Setup SSL on IBM i	58
1.5.2 Setup SSL Certificate Store on IBM i	59
1.5.3 Installing SSL Certificate Authorities on IBM i	65
1.5.4 Setting SSL Store Permissions	79

1.5.5 Create a TLS-SSL DCM Application	81
1.5.6 Enable HTTPS for an HTTP Server Instance	89

# 1. MDCMS REST API

## 1.1 MDChange REST API

MDCMS REST API Documentation - Midrange Dynamics.

### Version 8.6

Published May 9, 2024

#### Overview

In the [API Catalogue](#) below, there is a list of all REST API Resources, currently published in an active version of MDCMS.

For each resource, each valid method (GET, POST, PUT, DELETE) for the resource is described along with a list of parameters, example request, and example response.

All payloads for MDCMS APIs must be in JSON format.

#### URI Formats

The URIs for resources always have the following structure:

**https://** [Defined URL Endpoint in MDCMS] (1)/ [resource-name] (2)

1. Defined URL Endpoint in MDCMS from Interfaces setup. e.g. `devbox.mycompany.com/mdcmstest`

```
MDLREST          MD Production BC          30.05.24
SCRN1            MDCMS HTTP Server Configuration    14:42:12

Defined HTTP Server URL.: https://demo.mdcms.ch/mdcms

Defined HTTP Server Name: MDCMS
Defined Addresses/Ports.: *:1901
```

2. Resource-name from [API Catalogue](#) below. e.g. `/applications`

**For example:** `https://devbox.mycompany.com/mdcmstest/applications`

Select from the following sections:

<a href="#">MDChange REST API Catalogue</a>	A catalogue of MDChange REST APIs and their attributes.
<a href="#">Configure the MDChange REST API Server</a>	Configure the HTTP server for REST APIs
<a href="#">Authenticate Requests to the MDChange REST API Server</a>	How to create and use an authentication token used by the MDChange REST APIs
<a href="#">Setup SSL on IBM i</a>	Before an HTTP Server Instance can be HTTPS enabled, TLS/SSL must first be configured for the whole IBM i server.



Info

You can download a PDF version of this guide [here](#).

<b>REST API Catalogue</b>
<a href="#">Applications REST API</a>
<a href="#">Attributes REST API</a>
<a href="#">Levels REST API</a>
<a href="#">Level Wildcard REST API</a>
<a href="#">MDSEC User REST API</a>
<a href="#">Object Request REST API</a>
<a href="#">Open Projects REST API</a>
<a href="#">Open RFPs REST API</a>
<a href="#">Open Subtasks REST API</a>
<a href="#">Open Tasks REST API</a>
<a href="#">Pipeline Request Trigger REST API</a>
<a href="#">Project REST API</a>
<a href="#">RFP Acceptance REST API</a>
<a href="#">RFP Approval REST API</a>
<a href="#">RFP Installation REST API</a>
<a href="#">RFP Rollback REST API</a>
<a href="#">RFP Send REST API</a>
<a href="#">RFP Submission REST API</a>
<a href="#">Task REST API</a>

## 1.2 REST API Catalogue

---

### 1.2.1 Applications REST API

---

*Published: 2024-05-15*

#### RESOURCE NAME

/applications

#### GET

Returns all applications defined in MDCMS

#### Request

There are no parameters for this request.

#### Response

Example:

```
{
  "applications": [
    {
      "appl": "CSSB",
      "desc": "CSSB"
    },
    {
      "appl": "MD",
      "desc": "MD Settings and Updates"
    },
    {
      "appl": "TEST",
      "desc": "Test Primary App"
    },
    {
      "appl": "TSTB",
      "desc": "Test Primary application$"
    },
    {
      "appl": "WS",
      "desc": "WebSmart"
    }
  ]
}
```

## 1.2.2 Attributes REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/attributes

### GET

Returns all attributes defined in MDCMS for the given query parameters

### Request

### Path parameters

**appl** Required

string

The application code containing the attributes to list

**lvl** Required

integer

The application level containing the attributes to list

**objt**

string

The object type of the attributes to list

Example:

endpoint/mdcms/attributes?appl=TEST&lvl=10&objt=\*SQLALS

### Response

Example:

```
{
  "attributes": [
    {
      "appl": "TEST",
      "lvl": "10",
      "objt": "*SQLALS",
      "attr": "SQLALS",
      "desc": "alias",
      "objlib": "TEST80_10"
    }
  ]
}
```

## 1.2.3 Levels REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/levels

### GET

Returns all levels defined in MDCMS or all level for a specific application

### Request

### Path parameters

#### **appl**

string

The application code containing the levels to list

Example:

endpoint/mdcms/levels?appl=TEST

### Response

Example:

```
{
  "levels": [
    {
      "appl": "TEST",
      "lvl": "10",
      "desc": "T8 Dev"
    },
    {
      "appl": "TEST",
      "lvl": "11",
      "desc": "T8 Dev 2"
    },
    {
      "appl": "TEST",
      "lvl": "30",
      "desc": "T8 QA (local)"
    },
    {
      "appl": "TEST",
      "lvl": "50",
      "desc": "T8 Prod (local)"
    }
  ]
}
```



## 1.2.4 Level Wildcard REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/level/wildcard

### POST

Update the value for an existing Level Wildcard in MDCMS.

QTMHHTTP must be granted the right to invoke command MDUPDLWC in MDSEC

### Request

#### Path parameters

none

#### Body parameters

**wild** required

string

An existing 6 character wildcard identifier for application level based variables

**appl**

string

\*ALL - Apply the update to existing entries for all applications otherwise, provide a specific application ID to apply the update only to existing entries for that application

**lvl**

integer

0 - Apply the update to existing entries for all application levels otherwise, provide a specific level number to apply the update only to existing entries for that level number among 1 or more applications depending on the value for appl

**fldv**

string

the new value to apply to the wildcard, up to 160 characters in length

Example:

```
{ "wild": "CONXT", "appl": "TSTAPP", "lvl": 100, "fldv": "ABCxyz123" }
```

### Response

#### Body parameters

The response provides a structure with the following parameters:

**sev**

string

message severity with 10=ok, 20=warning and 30=error

**msg**

string

A description of the result of the attempt to update the wildcard value

Example:

```
{  
  "sev": "10",  
  "msg": "1 Wildcard entry has been updated"  
}
```

## 1.2.5 MDSEC User REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/mdsec/user

### POST

Create, update or delete a user in MDSEC. In the case of an update, the value for any parameter that isn't included in the request payload will be left as is.

QTMHHTTP must be granted the right to invoke command MDUPDUSR in MDSEC

### Request

#### Path parameters

none

#### Body parameters

**user** required

string

The user ID of the user to register in MDSEC

**opt**

string

\*UPDADD - If the user already exists in MDSEC, it will be updated with the provided information. Otherwise, the user will be added.

\*ADD - Add the provided information only if the user isn't already defined in MDSEC. \*REMOVE - Remove the user from MDSEC

**desc**

string

a description of the user.

**addr**

string

the email address of the user.

**extu**

string

The external user ID, if mapping from LDAP for the usage of MDWorkflow.

**act**

string

\*NO - set the user to deactivated in MDSEC \*YES - set the user to activated in MDSEC.

**grpa**

string

Specifies if authority for the user should be based on the authority permitted for the primary group that the user belongs to.

\*NO - The user's authority is based solely on the authority permitted for the user themselves. \*YES - The user's authority is based solely on the authority permitted for the primary group that the user belongs to in MDSEC. \*BOTH - The user's authority is based on the combination of the authority for the group and the specific user.

### **grpa**

string

The CCSID to use for the user, instead of the CCSID defined for the MDCMS instance. This value is used for character translation between the MDCMS database and MDOpen/MDWorkflow. Not all values are accepted in MDSEC - check the permitted values first by listing them from the MDCMS system settings.

### **addroles**

array(string)

The list of up to 20 MDSEC authority roles to apply to the user

### **rmvroles**

array(string)

The list of up to 20 MDSEC authority roles to revoke for the user

Example:

```
{ "user": "SOMEUSER", "desc": "some user to be registered in MDSEC", "addr": "someuser@mdcms.ch", "ccsid": "37", "addroles": ["MD_ADMCMS", "MD_RFP_SBM"], "rmvroles": ["MD_PGMR"] }
```

Response

### Body parameters

The response provides a structure with the following parameters:

#### **msg**

string

A description of the result of the attempt to create/update/delete the user

#### **sev**

string

message severity with 10=ok, 20=warning and 30=error

Example:

```
{
  "msg": "User updated",
  "sev": "10"
}
```

## 1.2.6 Object Request REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/object-request

### POST

Create an Object Request (check-out) in MDCMS.

### Request

### Path parameters

none

### Body parameters

**appl** Required

string

The ID of the target application for the object

**lvl** Required

integer

A level number of the application that allows check-outs

**objt** Required

string

The System or MDCMS Object Type code for the Object

**attr** Required

string

The MDCMS Attribute code that identifies the behavior and target locations for the requested object

**objn** Required

string

The name of the object to be requested

**rpth**

string

Specifies the relative portion of an IFS path that will be deployed with the object. For example, if the \*IFS attribute has a target fixed directory defined as /srv/dev and this object should be deployed to srv/dev/app1/dist, then the value of RPTH should be /app1/dist.

**srcn**

string

Specifies the name of the Source Member or IFS Source File to be requested. If a value isn't passed for the parameter, then the source name will be set to the same as the object name. This parameter is ignored if the attribute is defined as having no source.

**rsn**

string

The reason (or purpose) of the object request. If a value isn't passed for the parameter, then \*MIGRATE will be used.

\*MIGRATE - source and/or object will be migrated into the target application level \*DELETE - an existing object will be deleted  
 \*RECOMPILE - an object will be recompiled based on the source already in the environment \*UPDATE - update an ILE program or Service Program to re-bind the current modules and service programs to the program.

**user**

string

The user (developer) to assign to the object request If a value isn't passed for the parameter, then the current user of the job (QTMHHTTP) will be used.

**folb**

string

Specifies the library or IFS path that contains the object to be migrated to the specified level. The library should be a developer or team library that is not managed by MDCMS. If the attribute contains source only, then enter the library/path containing the source. If the object resides in IFS, provide the entire directory path starting with /. If a value isn't passed for the parameter, then the library will be set to the name of the user.

If the object is of type \*IFS or \*REMOTE and the POST request includes an attached file of the same name as the objn parameter, the attached file will be automatically copied to the path indicated on the folb parameter for eventual deployment on an RFP. A CURL example of attaching a file to the REST request is provided at the bottom of this page.

**fslb**

string

Specifies the library or IFS path that contains the source to be migrated to the specified level. The library should be a developer or team library that is not managed by MDCMS. If a value isn't passed for the parameter, then the object library (folb) will be used.

If the source file is of type \*IFS and the POST request includes an attached file of the same name as the srcn parameter, the attached file will be automatically copied to the path indicated on the fslb parameter for eventual deployment on an RFP. A CURL example of attaching a file to the REST request is provided at the bottom of this page.

**fsfl**

string

Specifies the source file containing the source member to be migrated. This parameter is ignored if the attribute is defined as having no source. If a value isn't passed for the parameter, then the name of the source file will be set to the source file defined on the attribute.

**copy**

string

Specifies if the source or object should be copied from the target environment to the library from which to be migrated from. This parameter is only considered for Reason \*MIGRATE. If the attribute defines a source and object location, only the source will be copied. If the source or object already exists in the From Library, it will not be replaced by the source or object in the target environment.

\*NO (default) - the source or object will be otherwise placed in the From Library prior to migration. \*YES - MDCMS will copy the Source or Object from the Target Environment to the From Library, if it doesn't already exist in the From Library.

**proj**

string

Specifies the Project to assign to the Request. The project, if entered, must already exist and be in an open status. If the project is not yet authorized, then the user must have MDSEC authority to authorize the Project and then MDCMS will do so automatically.

**task**

integer

Specifies an existing, active Task number. Omit or set to 0 if the request should be assigned directly to the Project.

**stsk**

integer

Specifies an existing, active Subtask number. Omit or set to 0 if the request should be assigned directly to the Project or Task

**arfp**

string

Specifies if the request should be immediately assigned to an RFP and the method of determining the RFP.

\*NO (default) - the request will be created without being assigned to an RFP. \*YES - The request will be assigned to the RFP number based on parameter RFP. \*AUTO - MDCMS searches for an open RFP matching the Application, Level, User and Description parameter values. If an RFP is found, the Request will be assigned to that RFP. If an RFP is not found, a new RFP will be created.

\*NEW - A new RFP will be created for the Application, Level, User and Description parameter values.

**rfp**

integer

Specifies the RFP to assign to the Request. Will only be used if parameter ARFP is set to \*YES.

**rfpd**

string

The description to be used for a new RFP or to search for an existing RFP. Will only be used if parameter ARFP is \*AUTO or \*NEW.

**creq**

string

If a level exists to migrate after this target level, this parameter specifies if the object requests should be generated for that level once this level's RFP is complete. This parameter will only be applied to the RFP if it is created during the processing of this command.

\*YES (default) or \*NO

**areq**

string

If a level exists to migrate after this target level, this parameter specifies if the generated object requests should be assigned to an RFP. This parameter will only be applied to the RFP if it is created during the processing of this command.

\*YES (default)

\*NO \*MANUAL - Requests for the next level will be assigned to a new RFP, but the RFP will not be automatically submitted, even if the next level is set to automatically submit RFPs by default.

**sreq**

string

If distribution levels are defined for this level, this parameter specifies if the RFP should be placed in the Send List. This parameter will only be applied to the RFP if it is created during the processing of this command.

\*YES (default) or \*NO

**lock**

string

Specifies whether or not the Request will be placed in Locked status

\*YES (default) or \*NO

#### **csqo**

integer

Specifies the sequence for compiling (lowest first) for objects in same RFP that have the same primary sort sequence in order to handle potential dependency issues.

#### **data**

string

Specifies the origin of the data that should be copied into a new or modified physical file/SQL Table.

\*SAME (default) - The data is mapped from the old format of the modified file to the new format of the file of the same name/target library. \*MIGRATE - The data is migrated with the file from the check-out location to the target library. \*NONE - The data is not migrated. The new file format will be empty. \*NONE is required for a logical file if it is replacing a physical file. character value - specify the system or SQL name of the file to copy from when deploying the requested file

#### **dmbr**

string

Specifies the member(s) to copy to the new version of a physical file/SQL Table or to migrate from the prior environment.

\*ALL (default) - all existing members are included for the copy \*FIRST - The first member in the originating file is copied. Any other members are omitted. character value - specify the specific name of the member to be copied from the originating file. Any other members are omitted.

#### **rpgm**

string

Specifies if MDRapid should be used to map the data from the old version of a file to the new version.

\*DEFAULT (default) - MDRapid will be used if the number of records in the file is at least the number in the MDRapid template for the attribute. Otherwise not. \*NO - MDRapid will not be used \*YES - MDRapid will be used

#### **rjrn**

string

Specifies if the new version of a table or access path should have the journaling attributes applied to it that belonged to the file that it replaced.

\*DFT (default) - The default option defined for the Application is used \*NO - Journaling will not be reapplied automatically \*YES - Journaling will be reapplied automatically

#### **rcst**

string

Specifies if the new version of a table should have the constraints applied to it that belonged to the file that it replaced.

\*DFT (default) - The default option defined for the Application is used \*NO - constraints will not be reapplied automatically \*YES - constraints will be reapplied automatically

#### **rtrg**

string



Specifies if the new version of a table should have the triggers applied to it that belonged to the file that it replaced.

\*DFT (default) - The default option defined for the Application is used \*NO - triggers will not be reapplied automatically \*YES - triggers will be reapplied automatically

#### **rlfm**

string

Specifies if the new version of a logical file should have the members added to it that belonged to the logical file that it replaced.

\*DFT (default) - The default option defined for the Application is used \*NO - members will not be reapplied to the logical file automatically \*YES - members will be reapplied to the logical file automatically

#### **dir**

string

Specifies whether or not the Request of an object of type \*IFS is a directory.

\*NO - not a directory (is treated as a file) \*YES - is a directory

#### **vref**

string

Optional Vendor Generated Identifier in order for an external process to easily identify the transaction record in the MDDAREQ table where each object-request transaction is logged.

#### **pipe**

string

If a pipeline server job should be notified of RFP activity for the RFP containing this object request, the Pipeline Server ID defined in MDOpen can be passed. The pipa and tkey parameters will also need to be passed.

#### **pipa**

string

If a pipeline server job should be notified of RFP activity for the RFP containing this object request, the MDCMS attribute defined for a \*PIPE object should be passed. MDCMS will then generate a \*PIPE object request and assign it to the same RFP as the primary object. The pipe and tkey parameters will also need to be passed.

#### **tkey**

string

The key that uniquely identifies the pipeline job that should be updated. This is typically a one-time build number and is used to notify the Pipeline server as the RFP proceeds. The pipe and pipa parameters will also need to be passed.

Example request body:

```
{ "appl":"TEST", "lvl":"10", "objt": "*PGM", "attr": "SQLRPGLE", "objn": "MYNEWPGM", "user": "MMORGAN", "rsn": "*migrate", "folb": "MMORGAN13", "fsfl": "QRPGLESRC", "proj": "DEMOCH", "task": "2", "stsk": "3", "arfp": "*YES", "rfp": "1709", "vref": "ep1234" }
```

Example CURL request including an attached file to be included with the object request. Variable MD\_REQ in this example is the json body containing the object-request parameters and variable MDCMS\_URI would be the URL string of the object-request resource.

```
sh "curl -v POST -H 'Expect:' -H 'Content-Type: multipart/mixed' -F 'payload=${MD_REQ};type=application/json' -F file=@sample.war ${MDCMS_URI}"
```

Response

Body parameters

The response provides a rtn structure with the following parameters:

**sev**

string

The message severity

10 - processed without errors or warnings 20 - processed, but warnings occurred 30 - did not process successfully due to errors

**msg**

string

The message text

Example:

```
{ "rtn": {  
  "sev": "10",  
  "msg": "Request created for MYNEWPGM into RFP 1709"  
}}
```

## 1.2.7 Open Projects REST API

---

*Published: 2024-05-15*

RESOURCE NAME

/open-projects

GET

Returns all active Projects in MDCMS

Request

Path parameters

**startsWith**

string

Limit the list to Projects that start with the given value

Example:

endpoint/mdcms/open-projects?startsWith=dem

## Response

## Example:

```
{
  "projects": [
    {
      "proj": "DEMOCH",
      "desc": "DEMOCH"
    },
    {
      "proj": "DEMOIS01",
      "desc": "Demo 01 in Israel"
    },
    {
      "proj": "DEMOPRO",
      "desc": "Demo Prosystem"
    },
    {
      "proj": "DEMOUK1",
      "desc": "demo of mdcms"
    },
    {
      "proj": "DEM003",
      "desc": "Demo project number three"
    },
    {
      "proj": "DEM003RT4",
      "desc": "Demo project number 3.test 4"
    },
    {
      "proj": "DEM0033",
      "desc": "Demo project 03"
    },
    {
      "proj": "DEM004",
      "desc": "test end-to-end"
    },
    {
      "proj": "DEM0044",
      "desc": "Demo project 44"
    },
    {
      "proj": "DEM00801",
      "desc": "Demo Project"
    },
    {
      "proj": "DEM0123",
      "desc": "demo 123 of dev flow"
    },
    {
      "proj": "DEM013RT",
      "desc": "Demo project number 13 with rich text"
    },
    {
      "proj": "DEM0180116",
      "desc": "demo of mdcms"
    }
  ]
}
```

## 1.2.8 Open RFPs REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/open-rfps

### GET

Returns all active RFPs in MDCMS for a specific application and level

### Request

#### Path parameters

**appl** Required

string

The application code containing the RFPs to list

**lvl** Required

integer

The application level containing the RFPs to list

**user**

string

The User that owns the RFP

Example:

endpoint/mdcms/open-rfps?appl=TEST&lvl=10&user=MMORGAN

## Response

## Example:

```
{ "rfps": [
  {
    "appl": "TEST",
    "lvl": "10",
    "rfp": "1340",
    "user": "MMORGAN",
    "desc": "Git contents of a Tag"
  },
  {
    "appl": "TEST",
    "lvl": "10",
    "rfp": "1605",
    "user": "MMORGAN",
    "desc": "test mdlmövo"
  },
  {
    "appl": "TEST",
    "lvl": "10",
    "rfp": "1630",
    "user": "MMORGAN",
    "desc": "parts of the git tree"
  },
  {
    "appl": "TEST",
    "lvl": "10",
    "rfp": "1655",
    "user": "MMORGAN",
    "desc": "test remote script"
  },
  {
    "appl": "TEST",
    "lvl": "10",
    "rfp": "1697",
    "user": "MMORGAN",
    "desc": "commit message for contents / start log entry"
  },
  {
    "appl": "TEST",
    "lvl": "10",
    "rfp": "1698",
    "user": "MMORGAN",
    "desc": "test jira workflow"
  },
  {
    "appl": "TEST",
    "lvl": "10",
    "rfp": "1731",
    "user": "MMORGAN",
    "desc": "8.2 demo"
  }
]
}
```

## 1.2.9 Open Subtasks REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/open-subtasks

### GET

Returns all active subtasks in MDCMS for a specific project and task

### Request

### Path parameters

**proj** Required

string

The Project ID

**task** Required

integer

The Project Task Number

Example:

endpoint/mdcms/open-subtasks?proj=DEMO044&task=1

### Response

Example:

```
{
  "subtasks": [
    {
      "proj": "DEMO044",
      "task": "1",
      "stsk": "1",
      "type": "ADMIN",
      "summary": "test with 1 subtask"
    }
  ]
}
```

## 1.2.10 Open Tasks REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/open-tasks

### GET

Returns all active tasks in MDCMS for a specific project

### Request

### Path parameters

**proj** Required

string

The Project ID

Example:

endpoint/mdcms/open-tasks?proj=DEMO044

### Response

Example:

```
{ "tasks": [ {  
  "proj": "DEMO044",  
  "task": "1",  
  "type": "ADMIN",  
  "summary": "test"  
} ] }
```



## 1.2.11 Pipeline Request Trigger REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/git/checkout

### POST

Checkout source or objects from a Git repository when not triggered directly by a repository push Webhook.

Optionally pass a unique tracing key for the pipeline, such as the build ID, to be able to invoke downstream RFP activity and get status updates from MDCMS back to the Pipeline.

Status updates can be performed using the MDUPDPIPE command on \*RFP exit point commands.

### Request

#### Path parameters

none

#### Body parameters

##### **href** Required

string

The URL of the Git repository in html or SSH format. This repository must be already defined in MDOpen and have at least one Continuous Integration or Cross-Referencing element defined for it.

##### **branch**

string

The branch to pull from, if not specified in the URL

##### **userName**

string

The name of the user that committed the changes to Git. This user will then be mapped to a MDCMS user based on the mapping definitions for the Git repository in MDOpen.

##### **oldHash** Required

string

the prior commit hash in Git

##### **newHash** Required

string

the new commit hash in Git

##### **commitMsg**

string

the commit message. If including the task reference in the message, it should be at the very beginning of the message with format:

PROJECT-TASK.SUBTASK:

##### **taskRef**

string

the reference of the MDCMS, Jira or Azure task to apply the checkouts to. This can be omitted if the commitMsg starts with the reference id.

**pipeline**

string

The 10-character ID of a Pipeline server defined in MDOpen. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**traceKey**

string

A unique key to identify the pipeline build that MDCMS should communicate with. This will be used for downstream update messages to the server using the MDUPDPIPE command.

Example:

```
{ "href": "ssh://git@bitbucket.org/yourcompany/yourproject", "branch": "main", "userName": "john@doe.com", "oldHash": "a2c8933436a99a4ea90bc0972e7728a4e8bea519", "newHash": "1a0c808bb1aa1886f896d3989645fb6574fc23a8", "commitMsg": "DEMOCH-3: flow demo", "pipeline": "BAMBOO", "traceKey": "YOURPROJECT-37" }
```

Response

Body parameters

The response provides a structure with the following parameters:

**transaction**

string

the transaction number which can be used to find the log record in file MDXREF/MDDFREP

**msg**

string

a detailed message of the error or successful transaction

Example:

```
{
  "transaction": "149",
  "msg": "MDFREP submitted to batch"
}
```

## 1.2.12 Project REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/project

### GET

Returns information about a specific project

### Request

### Path parameters

**proj** Required

string

The Project ID

Example:

endpoint/mdcms/project?proj=DEMO044

### Response

Body parameters

See POST method

Example:

```
{
  "proj": "DEMO044",
  "prjt": "ONLYTASKS",
  "agp": "TEST",
  "agrp": "",
  "ausr": "",
  "pri": "3",
  "edat": "20170430",
  "sts": "1",
  "dsc": "Demo project 44",
  "requester": "REN",
  "requestDate": "20170320",
  "requestTime": "115201",
  "closer": "",
  "closeDate": "0",
  "closeTime": "0",
  "hrse": "11.00",
  "cste": "1999.99",
  "hrsa": "10.00",
  "csta": "1300.00"
}
```

---

### POST

Create or update a Project in MDCMS.

When for update, parameters only need to be included in the body if a new value should be set for the parameter. For any parameters that aren't included, the existing value remains in place.

### Request

### Path parameters

none

### Body parameters

**proj** Required

string

The Project ID. If the ID already exists in MDCMS, then an update will be performed, otherwise the Project will be added to MDCMS.

**prjt**

string

A valid Project Type to apply to the project. If not included for a new Project, the default type will be used.

**agp**

string

An optional application code to apply to the project

**agrp**

string

The User Group to assign the Project to

**ausr**

string

A specific user to assign the Project to

**pri**

integer

The priority of the Project. If not included for a new Project, the priority will be set to 3=Medium.

1 - Critical 2 - High 3 - Medium 4 - Low 5 - Optional

**sts**

string

The Status of the Project. If not included for a new Project, the status will be set to 1=Open

**dsc** Required when new

string

A brief description or title for the Project

**hrse**

decimal

The number of hours expected to complete the project

**cste**

decimal

The expected cost to complete the project

**musr**

string

The user to register as the creator or modifier of the project

**edsc**

string

The extended description of the Project

Example:

```
{ "proj": "RESTPROJ3", "agp": "TEST", "prjt": "onlytasks", "ausr": "mmorgan", "pri": "2", "musr": "mmorgan", "edat": "20190415",  
"dsc": "Project created via REST API", "hrse": "40.56789", "cste": "4567.89", "edsc": "this used mdwsproj to invoke program  
mdupdproj für Erzeugung von Überall" }
```

Response

Body parameters

The response provides a rtn structure with the following parameters:

**sev**

string

The message severity

10 - processed without errors or warnings 20 - processed, but warnings occurred 30 - did not process successfully due to errors

**msg**

string

The message text

Example:

```
{ "rtn": {  
  "sev": "10",  
  "msg": "Project RESTPROJ3 updated"  
}}
```

## 1.2.13 RFP Acceptance REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/rfp/acceptance

### POST

Update the MDWorkflow Test Status for Project(s) in installed MDCMS RFPs.

Optionally confirm the final Test Status for the RFPs too, if the status has provisionally been set for all projects in the RFP.

An RFP will only be updated if:

- it's application, level, rfp number, and impacted projects or tasks match the filter criteria passed in the API request
- it is already completely installed and has an "ongoing" test status
- QTMHHTTP has been granted the right to invoke command MDWFARFP in MDSEC

### Request

#### Path parameters

none

#### Body parameters

**grp** Required

string

The User Group ID of the user to update the Workflow test status for RFPs. The group must be an involved acceptance group for the project(s) impacted by the RFPs to accept/reject.

**user**

string

The specific user in the Group to attribute the acceptance status to

\*USER - the user id of the job will be used (typically QTMHHTTP)

**act**

string

The action to carry out for each qualified project in each qualified RFP, based on the passed parameter values.

\*ACCEPT - Set the test status to accepted \*REJECT - Set the test status to rejected \*INPROG - Set the test status to In Progress

\*UNDO - Reset the provisional test status

**cmnt**

string

A comment to apply to the status update for informational purposes

**conf**

string

Whether or not to set the RFP test status to confirmed, if all impacted projects for the RFP have been provisionally accepted or if at least one of the impacted projects for the RFP has been provisionally rejected.

\*NO - Only set the provisional status. Final confirmation will occur later. \*YES - Set the RFP test status to confirmed and carry out any post-confirmation steps.

**appl**

string

Filter the RFP candidate list to those for the provided MDCMS application

**rfp**

integer

Update a specific RFP of the given number

**rfpt**

string

The type of specific RFP number passed in parameter RFP

\*CURRENT - the current number for an RFP \*FROM - the RFP number refers to the RFP installed into the prior level \*ORIG - each RFP to be updated originated from the specified RFP number, which was installed into the initial level for a chain of migrations across systems.

**flvl**

integer

The minimum application level for the test status update

**tlvl**

integer

The maximum application level for the test status update

**frfp**

integer

The minimum RFP number for the test status update

**trfp**

integer

The maximum RFP number for the test status update

**proj**

string

Specifies the Project that must be assigned to one or more of the objects in an RFP for that RFP to be considered. If a value isn't passed for this parameter, then the RFPs won't be filtered by a project and the test status for all projects in the RFP will be updated. If included, the RFP must impact the Project and the test status will only be applied to that project.

**task**

integer

Specifies the Project Task that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**stsk**

integer

Specifies the Project Subtask that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**pipeline**

string

The 10-character ID of a Pipeline server defined in MDOpen. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**traceKey**

string

A unique key to identify the pipeline build that MDCMS should communicate with. This will be used for downstream update messages to the server using the MDUPDPIPE command.

Example:

```
{ "grp": "TESTER1", "user": "USER1", "act": "*ACCEPT", "conf": "*YES", "proj": "demouk1", "task": "1", "stsk": "1" }
```

Response

Body parameters

The response provides a structure with the following parameters:

**transactionId**

integer

The transaction ID for the API call. All processed RFPs for the transaction ID are written to table MDCMS/MDDWFAR and can be queried with the ID value in column MDTRN.

**transactions**

array

A list of transaction message objects. Each object contains:

- sev - severity 10=ok, 20=warning and 30=error
- msg - the message text
- appl - the application of the rfp, if message for a specific rfp
- lvl - the application level of the rfp, if message for a specific rfp
- rfp - the rfp number, if message for a specific rfp
- proj - the project for which the test status was updated, if message for a specific project impacted by the rfp



## Example:

```
{
  "transactionId": "149",
  "transactions": [
    {
      "appl": "TEST",
      "lvl": "30",
      "rfp": "1765",
      "proj": "DEMOUK1",
      "sev": "10",
      "msg": "Test Status for Project in RFP updated"
    },
    {
      "appl": "TEST",
      "lvl": "30",
      "rfp": "1765",
      "sev": "10",
      "msg": "RFP Confirmed as Accepted"
    },
    {
      "appl": "TEST",
      "lvl": "30",
      "rfp": "1766",
      "proj": "DEMOUK1",
      "sev": "10",
      "msg": "Test Status for Project in RFP updated"
    },
    {
      "appl": "TEST",
      "lvl": "30",
      "rfp": "1766",
      "sev": "10",
      "msg": "RFP Confirmed as Accepted"
    }
  ]
}
```

## 1.2.14 RFP Approval REST API

---

Published: 2024-05-15

### RESOURCE NAME

/rfp/approve

### POST

Approve an MDCMS RFP for Installation

An RFP will only be approved if:

- it's application, level, rfp number, and impacted projects or tasks match the filter criteria passed in the API request
- it is currently in status 02 - Waiting for Approval
- QTMHHTTP has been granted the right to invoke command MDAPRRFP in MDSEC

### Request

#### Path parameters

none

#### Body parameters

##### **appl**

string

Filter the RFP candidate list to those for the provided MDCMS application

##### **rfp**

integer

Update a specific RFP of the given number

##### **rfpt**

string

The type of specific RFP number passed in parameter RFP

\*CURRENT - the current number for an RFP \*FROM - the RFP number refers to the RFP installed into the prior level \*ORIG - each RFP to be updated originated from the specified RFP number, which was installed into the initial level for a chain of migrations across systems.

##### **flvl**

integer

The minimum application level for the approval

##### **tlvl**

integer

The maximum application level for the approval

##### **frfp**

integer

The minimum RFP number for the approval

**trfp**

integer

The maximum RFP number for the approval

**proj**

string

Specifies the Project that must be assigned to one or more of the objects in an RFP for that RFP to be considered. **task**

integer

Specifies the Project Task that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**stsk**

integer

Specifies the Project Subtask that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**pipeline**

string

The 10-character ID of a Pipeline server defined in MDOpen. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**traceKey**

string

A unique key to identify the pipeline build that MDCMS should communicate with. This will be used for downstream update messages to the server using the MDUPDPIPE command.

Example:

```
{ "proj": "demouk1", "task": "1", "stsk": "1" }
```

Response

Body parameters

The response provides a structure with the following parameters:

**transactionId**

integer

The transaction ID for the API call. All processed RFPs for the transaction ID are written to table MDCMS/MDDARFP and can be queried with the ID value in column MDTRN.

**transactions**

array

A list of transaction message objects. Each object contains:

- sev - severity 10=ok, 20=warning and 30=error
- msg - the message text
- appl - the application of the rfp, if message for a specific rfp
- lvl - the application level of the rfp, if message for a specific rfp
- rfp - the rfp number, if message for a specific rfp

**Example:**

```
{
  "transactionId": "162",
  "transactions": [ {
    "appl": "TEST",
    "lvl": "30",
    "rfp": "1769",
    "sev": "10",
    "msg": "RFP TEST/1769 successfully approved"
  } ]
}
```

## 1.2.15 RFP Installation REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/rfp/install

### POST

Perform the installation (application update) phase of MDCMS RFPs.

An RFP will only be installed if:

- it's application, level, rfp number, and impacted projects or tasks match the filter criteria passed in the API request
- the submit, approve and MDRapid phases for the RFP are complete so that installation is pending.
- QTMHHTTP has been granted the right to invoke command MDINSRFP in MDSEC

### Request

#### Path parameters

none

#### Body parameters

##### **appl**

string

Filter the RFP candidate list to those for the provided MDCMS application

##### **flvl**

integer

The minimum application level that the RFP is targeting

##### **tlvl**

integer

The maximum application level that the RFP is targeting

##### **frfp**

integer

The minimum RFP number for installation

##### **trfp**

integer

The maximum RFP number for installation

##### **pend**

string

\*YES - RFPs in status 03 or IP will be considered \*NO - Only RFPs in status 03 will be considered \*ONLY - Only RFPs in status IP will be considered

##### **schdt**

string

If pending RFPs are considered, only process those with a maximum scheduled install date of a given value. Specify the date in format YYYYMMDD. If parameter isn't passed, the current date will be considered the maximum.

**proj**

string

Specifies the Project that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**task**

integer

Specifies the Project Task that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**stsk**

integer

Specifies the Project Subtask that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**pipeline**

string

The 10-character ID of a Pipeline server defined in MDOpen. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**traceKey**

string

A unique key to identify the pipeline build that MDCMS should communicate with. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**user**

string

The user to apply to the RFP as the installer of the RFP.

\*APPROVER - The user that is registered as having approved the RFP for installation is also registered as the installer \*USER - the current user of the job (typically QTMHHTTP)

or enter a valid user profile that is defined in MDSEC

Example:

```
{ "appl": "TEST", "flvl": "30", "tlvl": "30", "proj": "DEMOUK1", "task": "1" }
```

Response

Body parameters

The response provides a structure with the following parameters:

**transactionId**

integer

The transaction ID for the API call. All processed RFPs for the transaction ID are written to table MDCMS/MDDIRFP and can be queried with the ID value in column MDTRN.

**transactions**

array

A list of transaction message objects. Each object contains:

- sev - severity 10=ok, 20=warning and 30=error
- msg - the message text
- appl - the application of the rfp, if message for a specific rfp
- lvl - the application level of the rfp, if message for a specific rfp
- rfp - the rfp number, if message for a specific rfp

Example:

```
{
  "transactionId": "155",
  "transactions": [{
    "appl": "TEST",
    "lvl": "30",
    "rfp": "1767",
    "sev": "10",
    "msg": "RFP TEST/1767 installed"
  }]
}
```

## 1.2.16 RFP Rollback REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/rfp/rollback

### POST

Perform the rollback of an installation of an entire RFP.

An RFP will only be rolled back if:

- the passed application and rfp number are for an installed RFP
- QTMHHTTP has been granted the right to invoke command MDRBRFP in MDSEC

### Request

#### Path parameters

none

#### Body parameters

**appl** required

string

The application code of the RFP

**rfp** required

integer

The number of the previously installed RFP to be rolled back

**user**

string

The user to apply to the RFP as the installer of the rollback RFP.

\*INSTALLER - The user that originally installed the RFP \*USER - the current user of the job (typically QTMHHTTP)

or enter a valid user profile that is defined in MDSEC

Example:

```
{ "appl": "TEST", "rfp": "1767", "user": "mmorgan" }
```

### Response

#### Body parameters

The response provides a structure with the following parameters:

**transactionId**

integer

The transaction ID for the API call. All processed RFPs for the transaction ID are written to table MDCMS/MDDRRFP and can be queried with the ID value in column MDTRN.

**transactions**

array



A list of transaction message objects. Each object contains:

- sev - severity 10=ok, 20=warning and 30=error
- msg - the message text
- appl - the application of the rfp, if message for a specific rfp
- rfp - the rfp number, if message for a specific rfp

Example:

```
{
  "transactionId": "156",
  "transactions": [ {
    "appl": "TEST",
    "rfp": "1767",
    "sev": "10",
    "msg": "Rollback of TEST/1767 completed using RFP TEST/1768"
  } ]
}
```

## 1.2.17 RFP Send REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/rfp/send

### POST

Perform the send of MDCMS RFPs to target locations.

An RFP will only be sent if:

- it's application, level, rfp number, target location and impacted projects or tasks match the filter criteria passed in the API request
- the RFP must be open in the RFP Send list
- QTMHHTTP has been granted the right to invoke command MDSNDRFP in MDSEC

### Request

#### Path parameters

none

#### Body parameters

##### **appl**

string

Filter the RFP candidate list to those for the provided MDCMS application

##### **flvl**

integer

The minimum application level of the RFP to be sent

##### **tlvl**

integer

The maximum application level of the RFP to be sent

##### **frfp**

integer

The minimum RFP number

##### **trfp**

integer

The maximum RFP number

##### **loc**

string

the location to send the RFP to. Either a specific location or:

\*ALLDFT - All target locations defined for the RFP's level that have the Default to Send property set to Y=Yes or M=Yes for Manual Send. \*ALL - All target locations defined for the RFP's level

**locgrp**

string

The Location Group to send the RFP to. MDCMS will only send to locations that meet the value for parameter loc and locgrp. Either a specific location group or:

\*ALL - The locations to send to aren't limited to a specific group

**ftlvl**

integer

The minimum target application level

**ttlvl**

integer

The maximum target application level

**merge**

string

Specifies if, in the case of multiple RFPs for the same local level, they should be merged into a single RFP before sending. The merge will occur for each level and result in the lowest RFP number for the level being the container for all of the impacted objects.

\*NO - each RFP will be sent separately \*YES - All RFPs for the same application level will be merged into the lowest RFP so that a single RFP is sent to the target locations for that level.

**resend**

string

Specifies if an RFP should be sent to a target level, if it has already been sent to that level. If the installation completed for the target level, then the RFP won't be resent.

\*NO - a target level will be omitted, if the RFP has already been sent to it \*YES - the RFP will be resent to a target level, even if it had been sent before, as long as the installation hasn't completed on the target level.

**insdt**

string

The date to install the RFP on the target system. Specify the date in format YYYYMMDD. If parameter isn't passed, the current date will be considered the install date.

**instm**

string

The time of day to install the RFP on the target system. Specify the time in format HHMMSS. If parameter isn't passed, the current time will be considered the install time.

**tzone**

string

The time zone of the time to install the RFP on the target system.

\*TARGET - the installation date/time is based on the time zone of the target system. \*LOCAL - the installation date/time is based on the time zone of the local sending system.

**proj**

string

Specifies the Project that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**task**

integer

Specifies the Project Task that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**stsk**

integer

Specifies the Project Subtask that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**pipeline**

string

The 10-character ID of a Pipeline server defined in MDOpen. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**traceKey**

string

A unique key to identify the pipeline build that MDCMS should communicate with. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**batch**

string

Specifies if the target levels are part of a batch send, so that a single process will send to several different targets at the same time rather than each one individually. This provides a way to avoid conflicts when trying to send to multiple targets for the same RFP using multiple calls to this resource.

\*ONLY - The send should be invoked immediately without batching multiple calls to this command together \*FIRST - Any selected targets in this call to MDSNDRFP are to be placed in an initialized batch list per RFP. This should be the value used for the first MDSNDRFP call for a batch. \*ADD - Any selected targets in this call to MDSNDRFP will be appended to an existing list per RFP. If the list for a specific RFP doesn't exist yet, it will be started. \*LAST - Any selected targets in this call to MDSNDRFP will be appended to an existing list per RFP. If the list for a specific RFP doesn't exist yet, it will be started. The send for the entire batch list per RFP will then be immediately processed. This must be the value used for the last MDSNDRFP call for a batch.

Example:

```
{ "appl": "TEST", "flvl": "30", "tlvl": "30", "frfp": "1730", "trfp": "1770", "merge": "*yes", "resend": "*yes", "proj": "DEMOUK1" }
```

Response

Body parameters

The response provides a structure with the following parameters:

**transactionId**

integer

The transaction ID for the API call. All processed RFPs for the transaction ID are written to table MDCMS/MDDRSND and can be queried with the ID value in column MDTRN.

**transactions**

array

A list of transaction message objects. Each object contains:

- sev - severity 10=ok, 20=warning and 30=error
- msg - the message text
- appl - the application of the rfp, if message for a specific rfp
- lvl - the application level of the rfp, if message for a specific rfp
- rfp - the rfp number, if message for a specific rfp
- rloc - the target location ID
- tlvl - the target level number

Example:

```
{
  "transactionId": "169",
  "transactions": [
    {
      "appl": "TEST",
      "lvl": "30",
      "rfp": "1765",
      "rloc": "MDDEMO",
      "tlvl": "30",
      "sev": "10",
      "msg": "RFP selected for Send to Target Level"
    },
    {
      "appl": "TEST",
      "lvl": "30",
      "rfp": "1765",
      "rloc": "SF0",
      "tlvl": "30",
      "sev": "10",
      "msg": "RFP selected for Send to Target Level"
    }
  ]
}
```

## 1.2.18 RFP Submission REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/rfp/submit

### POST

Perform the initial submit (verification and compile) phase of MDCMS RFPs.

An RFP will only be submitted if:

- it's application, level, rfp number, and impacted projects or tasks match the filter criteria passed in the API request
- the RFP status is currently 01 - Requests Assigned or SP - Submission Pending
- QTMHHTTP has been granted the right to invoke command MDSBMRFP in MDSEC

### Request

#### Path parameters

none

#### Body parameters

##### **appl**

string

Filter the RFP candidate list to those for the provided MDCMS application

##### **flvl**

integer

The minimum application level that the RFP is targeting

##### **tlvl**

integer

The maximum application level that the RFP is targeting

##### **frfp**

integer

The minimum RFP number for submission

##### **trfp**

integer

The maximum RFP number for submission

##### **pend**

string

\*YES - RFPs in status 01 or SP will be considered \*NO - Only RFPs in status 01 will be considered \*ONLY - Only RFPs in status SP will be considered

##### **schdt**

string

If pending RFPs are considered, only process those with a maximum scheduled submit date of a given value. Specify the date in format YYYYMMDD. If parameter isn't passed, the current date will be considered the maximum.

**proj**

string

Specifies the Project that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**task**

integer

Specifies the Project Task that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**stsk**

integer

Specifies the Project Subtask that must be assigned to one or more of the objects in an RFP for that RFP to be considered.

**pipeline**

string

The 10-character ID of a Pipeline server defined in MDOpen. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**traceKey**

string

A unique key to identify the pipeline build that MDCMS should communicate with. This will be used for downstream update messages to the server using the MDUPDPIPE command.

**user**

string

The user to apply to the RFP as the submitter of the RFP.

\*CREATOR - The user that is the owner of the RFP is also registered as the submitter \*USER - the current user of the job (typically QTMHHTTP)

or enter a valid user profile that is defined in MDSEC

Example:

```
{ "appl": "TEST", "flvl": "30", "tlvl": "30", "proj": "DEMOUK1", "task": "1" }
```

Response

Body parameters

The response provides a structure with the following parameters:

**transactionId**

integer

The transaction ID for the API call. All processed RFPs for the transaction ID are written to table MDCMS/MDDSRFP and can be queried with the ID value in column MDTRN.

**transactions**

array

A list of transaction message objects. Each object contains:

- sev - severity 10=ok, 20=warning and 30=error
- msg - the message text
- appl - the application of the rfp, if message for a specific rfp
- lvl - the application level of the rfp, if message for a specific rfp
- rfp - the rfp number, if message for a specific rfp

Example:

```
{
  "transactionId": "163",
  "transactions": [ {
    "appl": "TEST",
    "lvl": "30",
    "rfp": "1769",
    "sev": "10",
    "msg": "RFP TEST/1769 submitted"
  } ]
}
```



## 1.2.19 Task REST API

---

*Published: 2024-05-15*

### RESOURCE NAME

/task

### GET

Returns information about a specific task or subtask

### Request

### Path parameters

**proj** Required

string

The Project ID

**task** Required

integer

A Task Number within the Project

**stsk** Required

integer

A Subtask within the Task. Pass the value of 0 if retrieving information for the Task itself.

Example:

endpoint/mdcms/task?proj=MDS&task=141&stsk=0

### Response

#### Body parameters

See POST method

Example:

```
{
  "proj": "MDS",
  "task": "141",
  "stsk": "0",
  "tskt": "SD_ENHANCE",
  "sum": "Default Project and Task Types",
  "agp": "",
  "iref": "",
  "pri": "3",
  "sts": "7",
  "dued": "0",
  "duet": "0",
  "agrp": "",
  "ausr": "MMORGAN",
  "tstg": "",
  "tstu": "",
  "requester": "JIRA",
  "requestDate": "20190211",
  "requestTime": "191136",
  "closer": "JIRA",
  "closeDate": "20190416",
  "closeTime": "121142",
  "hrse": ".00",
  "cste": ".00",
  "hrsa": ".00",
  "csta": ".00"
}
```

## POST

Create or update a Task or Subtask in MDCMS.

When for update, parameters only need to be included in the body if a new value should be set for the parameter. For any parameters that aren't included, the existing value remains in place.

## Request

## Path parameters

none

## Body parameters

**proj** Required

string

The ID of an existing, open Project

**task**

integer

Specifies an existing Task number, or 0 if information is for a new Task Number. MDCMS will automatically generate the number for a new Task.

**stsk**

integer

Specifies an existing Subtask number. If 0: when nsts (new subtask) = \*NO, then the API will process at the Task level when nsts (new subtask) = \*YES, then the API will create a new Subtask

**nsts**

string

Specifies if the values should be saved to a new Subtask.

\*NO If parameter task = 0, then a new task will be created. If task > 0 and stsk = 0, then the existing task will be updated. If task > 0 and stsk > 0, then the existing subtask will be updated.

\*YES If parameter task = 0, then a new task will be created. If parameter task > 0, then a new subtask for the task will be created

\*REF If the value of iref is found in the database for the given project, then the referenced task or subtask will be updated. If parameter task = 0, and the value of IREF isn't found in the database for the given project, then a new task will be created. If parameter task > 0, and the value of IREF isn't found in the database for the given project, then a new subtask will be created for the task.

\*NOREF If the value of IREF is found in the database for the given project, then the referenced task or subtask will be updated. If the value of IREF isn't found in the database for the given project, then nothing will occur.

**tskt**

string

A valid Task Type to apply to the Task. If not included for a new Task, the default type will be used.

**sum** Required when new

string

A brief description of the task

**agp**

string

An optional application code to apply to the task

**iref**

string

An optional internal reference code for the task

**pri**

integer

The priority of the Task. If not included for a new Task, the priority will be set to 3=Medium.

1 - Critical 2 - High 3 - Medium 4 - Low 5 - Optional

**sts**

string

The Status of the Task. If not included for a new Task, the status will be set to 1=Open

**dued**

integer

The Date when the Task or Subtask is expected to be completed. The format of the date is YYYYMMDD with Y=Year, M=Month and D=Day of Month.

**duet**

integer

The Time when the Task or Subtask is expected to be completed. The format of the time is HHMMSS with H=Hour, M=Minute and S=Second

**agrp**

string

The User Group to assign the task to

**ausr**

string

A specific user to assign the task to

**tstg**

string

The User Group responsible for testing the results of the task

**tstu**

string

A specific user responsible for testing the results of the task

**hrse**

decimal

The number of hours expected to complete the task

**cste**

decimal

The expected cost to complete the task

**musr**

string

The user to register as the creator or modifier of the task

**edsc**

string

The extended description of the task

Example:

```
{ "proj": "demopro", "task": "1", "stsk": "1", "tskt": "admin", "sum": "a subtask created directly from the rest api", "nsts": "*no",
"agp": "TEST", "agrp": "pgmr 1", "pri": "2", "sts": "3", "musr": "mmorgan", "dued": "20190415", "duet": "110000", }
```

Response

Body parameters

The response provides a rtn structure with the following parameters:

**sev**

string

The message severity

10 - processed without errors or warnings 20 - processed, but warnings occurred 30 - did not process successfully due to errors

**msg**

string

The message text

Example:

```
{ "rtn": {
  "sev": "10",
  "msg": "Subtask DEMOPRO 1.1 updated"
}}
```

## 1.3 Configure the MDCMS REST API Server

---

Published: 2024-05-15

Available from MDCMS Version 8.2

### 1.3.1 Overview

---

#### What is REST?

From Wikipedia:

**Representational State Transfer (REST)** is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, termed *RESTful* Web services (RWS), provide interoperability between computer systems on the Internet. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations. Further information can be found in [Wikipedia](#).

#### The MDCMS REST API Server

MDCMS Version 8.2 and higher provides a collection of REST APIs (Web services) that can be used to share information between MDCMS and external tools via HTTP.

The server itself is installed as an instance of the native IBM http apache server, which is automatically available as part of the core OS/400 licensed program stack.

The APIs themselves are standard ILE RPG programs, which also run on any OS/400 operating system without further prerequisites. They are invoked by the http server using the native IBM CGI framework.

#### Prerequisites

- MDCMS with a minimum version of 8.5.2 must be installed on each IBM i partition that this extension will connect to.
- A valid MDOpen license must be active on each IBM i partition that this extension will connect to.
- You have a user profile and are registered in MDSEC on each IBM i partition that this extension will connect to.
- The MDCMS REST API and Diagramming Server, which is an Apache http server, must already be generated, active and available through any firewall on each IBM i partition that this extension will connect to.

### 1.3.2 Configure the MDCMS REST Server

---

#### Generate the Server

To create the server for an instance of MDCMS on a partition:

1. command MDCMS within a 5250 session
2. Option 1 - MDCMS Setup Menu
3. Option 10 - Interface Settings
4. Option 9 - MD REST API Server
5. Option 2 - Generate Server

```
MDLGRAS                MD Dev                16.05.24
SCRN1                  Generate MDCMS HTTP Server             17:42:04

Server Name . . . MDCMS

Port Number . . . _____
```

F3=Exit F6=Messages F8=Submitted Jobs F11=View Output F21=Sys Command

<b>Server Name</b>	The name of the Server to create. The server can't already exist. Only one server should be active for a given instance and partition of MDCMS. When created, the server configuration will be placed in IFS folder /www/<server name>
<b>Port Number</b>	The port number that the server should listen to for incoming http requests. The port number is mandatory and should not be attributable to any other server on the partition.

When Enter is pressed, the server is created and automatically started.



It is highly recommended to add the following command to the QSTRUP program so that the server will automatically restart after an IPL:

```
STRTCPSVR SERVER(\*HTTP) HTTPSVR(<server name>)
```

### Verify Server is Running

To check if the server is running, use command : `WRKSBSJOB QHTTPSVR .`

If the server is running, several jobs that are named the same as the Server (usually MDCMS) will be in ACTIVE status.

If not running, use `STRTCPSVR SERVER(\*HTTP) HTTPSVR(<server name>)` to start.

If still not running, then `WRKLNK '/www/<server name>/logs'` to troubleshoot the problem.



### Additional Requirement for MDOpen VSCode

- The OpenSSH server must be active and available through any firewall on each IBM i partition that this extension will connect to.
- The Code for IBM i extension from Halcyon Tech should also be installed, though connections do not have to be created in the Code for IBM i extension for MDOpen to function, as MDOpen creates its own connections.

### Set up MDCMS HTTP Server to use HTTPS (TLS/SSL)

HTTPS connectivity for the MDCMS REST API's can be achieved in one of two ways:

1. Directly configuring the MDCMS HTTP server instance to use HTTPS(TLS/SSL).
2. Via a Proxy server instance (described below) where the Proxy server instance is HTTPS enabled. This approach will enable multiple HTTP server instances to use a single HTTPS configuration

HTTPS requires TLS/SSL to be set up on the IBM i server, before an HTTP Server instance can be HTTPS enabled. If SSL is already in use on your IBM i, you can jump to the second step below:

1. [Setup TLS/SSL on the IBM i](#)
2. [Enable an HTTP server instance for HTTPS](#)

### Set up Reverse-Proxy to Forward REST Service Requests from Default HTTP Server

It is recommended to proxy all requests to the MDCMS server through the default http server on the partition because:

- no need to indicate the port number in the URL
- no need to add another port number to the firewall rules
- potentially take advantage of existing SSL configuration in the default http server

To configure the proxy:

1. use WRKLNK to navigate to the /www location of the default server (usually named APACHEDFT)
2. view the conf directory
3. edit the httpd.conf file
4. ensure the following modules are listed in the configuration. If missing then insert them into the configuration:

```
LoadModule proxy_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM
LoadModule proxy_http_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM
LoadModule proxy_connect_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM
LoadModule proxy_ftp_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM
LoadModule proxy_balancer_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM
```

5. add the following 3 lines to the configuration:

```
ProxyPreserveHost On
ProxyPass /mdcms/ http://localhost:1901/mdcms/
ProxyPassReverse /mdcms/ http://localhost:1901/mdcms/
```

Replace 1901 with the port number you specified for the server mdcms is correct if the instance of MDCMS is MDCMS. If your instance is TEST for example, then replace all 4 occurrences of mdcms with mdcmsTest. NOTE: mdcms in the URL is always the instance of MDCMS, regardless of the name of the server.

6. save the configuration and restart the default server

#### Define the URL to reach the MDCMS Server

MDCMS needs to be aware of the URL to the server in order to set up WebHooks for certain external servers such as JIRA. To define the URL:

1. command MDCMS within a 5250 session
2. Option 1 - MDCMS Setup Menu
3. Option 10 - Interface Settings
4. Option 9 - MD REST API Server
5. Option 1 - Set/Change URL

The Endpoint is the address of the partition, including the transport method http or https. Don't include the context path in the endpoint.

Example:

```
https://devbox.mycompany.com
```

When a REST request is sent to the endpoint, it will then be followed by the name of the mdcms instance (usually mdcms) and then the resource name of the API to be invoked.

Example to get a list of all applications defined in MDCMS:

```
https://devbox.mycompany.com/mdcms/applications
```

#### Test Connection to the MDCMS REST APIs

The simplest test, if your PC has access to the IBM i, is to open a browser and enter the following in the address prompt:

```
https:///mdcms/applications
```

replacing with the endpoint defined in your MDCMS settings.

If not using the default instance of MDCMS, replace mdcms with the name of the instance in lowercase.

If not using https, replace https with http.

If the test works, you'll see a list of your applications in JSON format. NOTE: From MDCMS 8.5 onwards, an API Token is required to authenticate a request to the REST APIs. If the connection test works, you will get a Status 401 with message "API Token not found" in the response body.

For more elaborate tests of the various services, including being able to include an API Token, we recommend the use of Postman or SoapUI.



**Tip**

If there are still connectivity problems, try this [MDCMS Connectivity Troubleshooting](#) knowledge guide.



## 1.4 Authenticate Requests to the MDCMS REST API Server

---

*Published: 2024-05-15*

Relevant from MDCMS Version 8.5

### 1.4.1 Overview

---

In order to protect MDCMS information from unauthorized access, a **bearer token** is expected to be included in the API request header. If the token is not present, MDCMS will return a 401-Unauthorized status.

If the token is present, it will be checked against the list of unexpired tokens. If not found, MDCMS will return a 401-Unauthorized status. If found, MDCMS will proceed further with carrying out the request based on the user that owns the token.

### 1.4.2 Generate a Token

---

Any user that is registered in MDSEC may generate a token for themselves. Any user that has MDSEC Administration rights may additionally generate tokens for other users. This can be useful when using a token applied to a service user rather than a human user.

To generate, do the following:

1. Within a 5250 session, type command MDSEC and press Enter
2. Select option 8 = API Tokens and press Enter
3. Press F6 = Add
4. Provide a description of the Token and a Valid Until Date and press Enter

The API Token will appear on the screen. **IMPORTANT:** Copy the token value and store in a secure location. It will not be possible to view the value of the token again

### 1.4.3 Manage Existing Tokens

---

Any user that is registered in MDSEC may manage their own tokens. Any user that has MDSEC Administration rights may additionally manage tokens for other users. This can be useful when using a token applied to a service user rather than a human user.

To manage, do the following:

1. Within a 5250 session, type command MDSEC and press Enter
2. Select option 8 = API Tokens and press Enter
3. Use option 2 to edit the description or Valid Until Date, use option 3 to copy the token or use option 4 to delete the token

### 1.4.4 Example Request Header

---

```
Authorization: Bearer MTgzNTg1NDIxMDA1MzkxNzIyOTYzMTA3Mjk3O3U2Nzg5ODAxN
```

## 1.5 Setting Up SSL on IBM i

---

### 1.5.1 Setup SSL on IBM i

---

*Published: 2022-03-14*

SSL/TLS uses digital certificates to establish the SSL tunnel which encrypts/decrypts traffic between the client and the server. These certificates are stored on the IBM i and managed with the Digital Certificate Manager (DCM).

Before an HTTP Server Instance can be HTTPS enabled, TLS/SSL must first be configured for the whole IBM i server.

#### SETUP TLS/SSL ON IBM I

If TLS/SSL is if not already configured on your IBM i, please follow these steps:

<a href="#">Setup SSL Certificate Store on IBM i</a>
<a href="#">Installing SSL Certificate Authorities on IBM i</a>
<a href="#">Setting SSL Store Permissions</a>
<a href="#">Create a DCM Application</a>

#### ENABLE HTTPS FOR AN HTTP SERVER INSTANCE ON IBM I

Follow [this article](#) if TLS/SSL is already configured on the IBM i, and the only requirement is to enable an HTTP server instance for HTTPS.

## 1.5.2 Setup SSL Certificate Store on IBM i

---

Published: 2024-05-15

This process will guide you through setting up the Digital Certificate Manager to enable your iSeries to interact as a client to other external servers requiring SSL connections. An example would be if you needed to send an XML credit request to Trans Union or other credit provider from your IBM i.

### Step 1: Enter Digital Certificate Manager

Please note that this process may be different for machines on V5R4. Additionally, you will need to verify that 5722AC3 (Crypto Access Provider 128-bit) is installed on your IBM i.

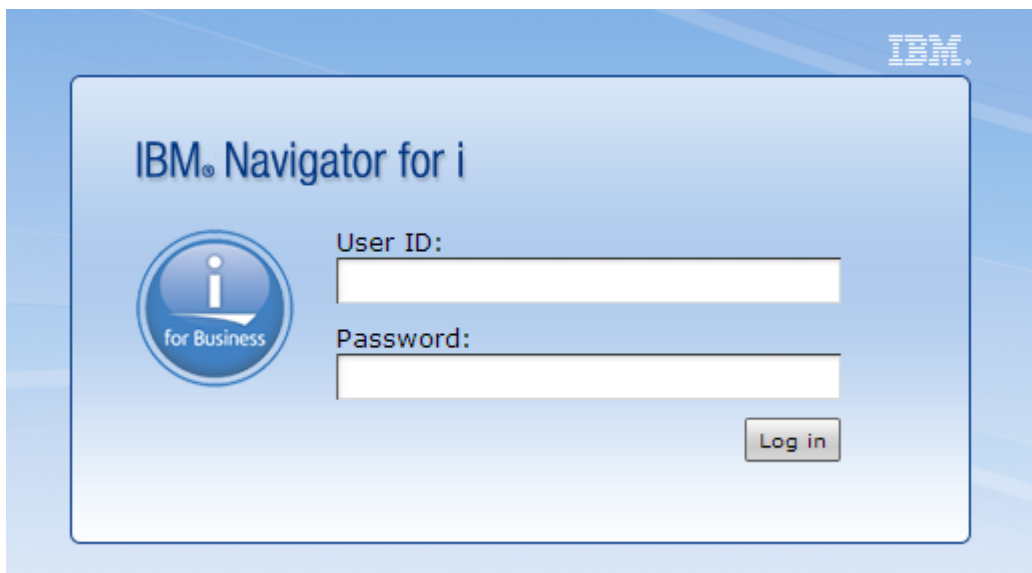
To begin, verify that the \*ADMIN HTTP server job is running with the following command:

```
WRKSBSJOB SBS(QHTTSPVR)
```

If you don't see \*ADMIN in the list, please run the following command to start it:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

After you've ensured that the \*ADMIN server is running, open a web browser (Internet Explorer is recommended), and go to `http://YourIBMIPAddress:2001` - you should see a login page as seen below:



Enter your IBM i username and password, and click "Log in". You should see a page split into two sections - a menu on the left, and a larger content area on the right that looks like the below image:


Welcome

## Welcome to the IBM Navigator for i

IBM Navigator for i provides an easy to use interface for the web-enabled IBM i management tasks.

Expand IBM i Management in the left-hand navigation area to get started.







To see the previous version of the 2001 port tasks and where they are located now, click below.

 [IBM i Tasks Page](#)

Click the “IBM i Tasks Page” link.

IBM i Tasks

IBM i Tasks allows you to access the tasks that were previously displayed on the IBM i Tasks web page.

-  IBM Web Administration for i: [http https](#)  
Allows you to manage and configure HTTP servers and application servers (Located in Internet Configurations)
-  New Digital Certificate Manager: [http https](#)  
A new user experience to allow you to create, distribute, and manage Digital Certificates
-  Digital Certificate Manager: [http https](#)  
Allows you to create, distribute, and manage Digital Certificates (Located in Internet Configurations)
-  IBM Tivoli Directory Server Web Administration Tool: [http https](#)  
Allows you to administer the IBM Tivoli Directory Server for i (Located in Network)
-  IBM IPP Server for i: [http https](#)  
Allows you to configure the IBM IPP Server (Located in Internet Configurations)
-  Cryptographic Coprocessor: [http https](#)  
Allows you to configure the cryptographic coprocessor (Located in Security)

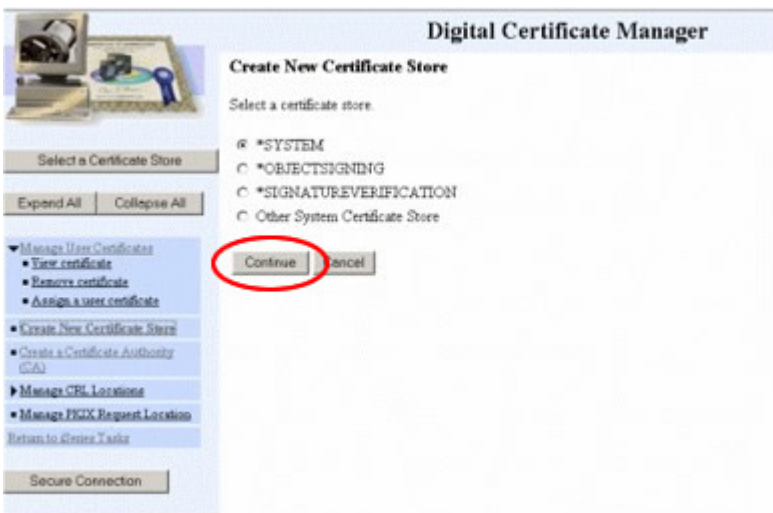
Now, click the “Digital Certificate Manager” link. You may be prompted to log in again - if you are, enter your IBM i username and password. It is recommended to log into the Digital Certificate Manager on a profile with elevated authority.

Step 2: Create New Certificate Store

Select the link “Create New Certificate Store”



Ensure \*SYSTEM is selected, and then select the “Continue” button. Note: if \*SYSTEM does not appear, this process has likely already been completed on your IBM i.



Step 3: Select Yes

Select “Yes”, and then press the “Continue” button.



Step 4: Finish Entering Data

Put anything you want in the "Certificate label" field. Then, specify a "Password", and record it for future reference. Fill out the remaining fields, populating them with whatever data is necessary, and then select the "Continue" button.

**Digital Certificate Manager**

**Create New Certificate Store with a Certificate**

Certificate type: Server or client  
 Certificate store: \*SYSTEM

Use this form to create a certificate and certificate store.

Key size: 1024 (bits)

Certificate label: Any Name (required)

Certificate store password: (required)

Confirm password: (required)

**Certificate Information**

Common name: Any Name (required)

Organization unit: (required)

Organization name: Any Name (required)

Locality or city: (required)

State or province: Minnesota (required-minimum of 3 characters)

Country or region: US (required)

Continue Cancel

Step 5: Store Certificate Key

Cut and paste the below certificate key into a text editor (like Notepad) and save it someplace secure. Select the "OK" button.

**Digital Certificate Manager**

**Certificate Request Created**

The certificate request data is shown below. Copy and paste the request data, including both the Begin request and End request lines, into the form that the Certificate Authority (CA) provided.

**Warning:** If you exit this page, the certificate request data is lost. Therefore, make sure you carefully copy and paste the data into the Certificate Authority (CA) form or into a file for later use.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBhjCBBAIBADBNHQswCQYDVQQGEwJVUzESMBAGAEUECHJTV1ubwVzb3RlRkFk
DwYDVoQRKwhBbnkgTmFt2TERMA8GA1UEAxNIQW55IE5hbWUgD8wDQVJRoZThvcH
AQEBBQADgYUAMIGJAoGBANVqkomo1WGTREW4Uj8BEOzeFEBatOJYFeSInwJa6w8G
WQzo0tN8WxOdGef6twnE4NAr2wRXc2VlFf3g80wCPdeN5tGjc4Za7wO5w49dDvHs
LLJ4AYjh/kFPCEabw5KBY/HhzeBwL7xj88o+TJxgt/652x0LJ7Rdqp8eS1a1a25v
AgNBAAQgADANBgkqhkiG9w0BAQQAQAOBgQCFAv12KcbSyxrh1tn11KLSK0S4bSs2
3rwrEOK+2u+x3OEU2MDDH3dM1HXsvt5zav4NzB2GXUS+3Wz0D011kWFNPGoAMUK
KV/4Pqy1g0jdg7StCLp+2TtKHR4S6zsvURS5YbRk4B0ggJzJ2UYezp1d00051G5
HDS+08hAuQ0/1q==
-----END NEW CERTIFICATE REQUEST-----
```

**Note:** You must click on the Select a Certificate Store button in the left frame to refresh the Digital Certificate Manager (DCM) to work with this new certificate store.

OK

Step 6: Ensure Proper Configuration

Selecting the "Select a Certificate Store" button at the top of the left sidebar will place you at the below screen. Make sure \*SYSTEM is selected, and select the "Continue" button.



**Digital Certificate Manager**

**Select a Certificate Store**

Select the certificate store that you want to open.

\*SYSTEM

Other System Certificate Store

Continue Cancel

Select a Certificate Store

Expand All Collapse All

▼ Manage User Certificates

- View certificate
- Remove certificate
- Assign a user certificate

■ Create New Certificate Store

■ Create a Certificate Authority (CA)

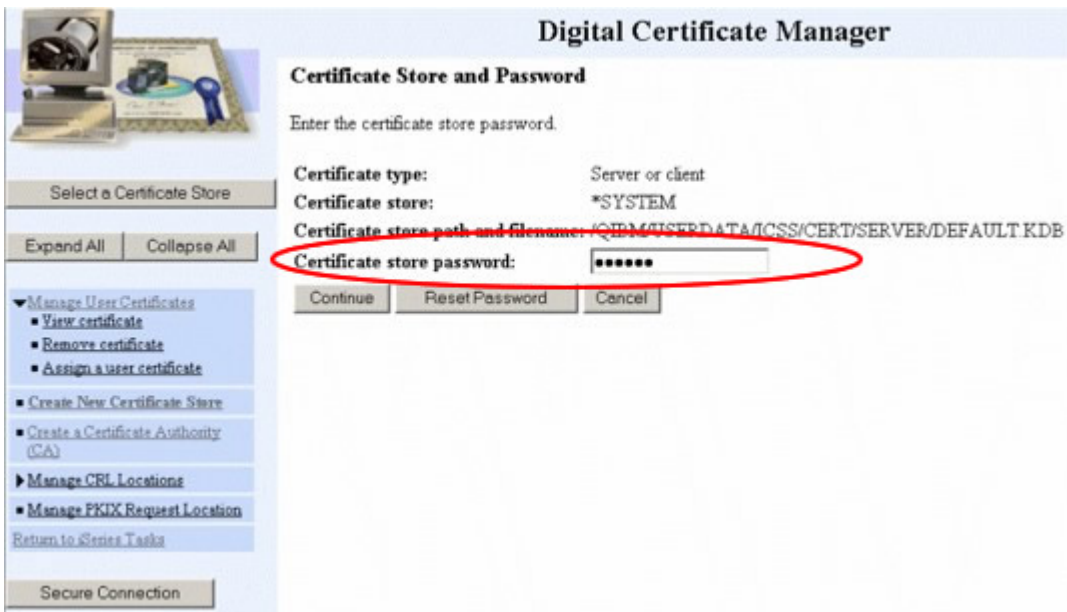
► Manage CRL Locations

■ Manage PKIX Request Location

[Return to iSeries Tasks](#)

Secure Connection

Enter the password you specified in Step 4, and select the Continue button. Note: If you ever forget the password, you can simply select "Reset Password" - you will be allowed to reset the password without knowing the previous password.



**Digital Certificate Manager**

**Certificate Store and Password**

Enter the certificate store password.

Certificate type: Server or client

Certificate store: \*SYSTEM

Certificate store path and filename: /QIBM/ISERDATA/ICSS/CERT/SERVER/DEFAULT.KDB

Certificate store password: [password field]

Continue Reset Password Cancel

Select a Certificate Store

Expand All Collapse All

▼ Manage User Certificates

- View certificate
- Remove certificate
- Assign a user certificate

■ Create New Certificate Store

■ Create a Certificate Authority (CA)

► Manage CRL Locations

■ Manage PKIX Request Location

[Return to iSeries Tasks](#)

Secure Connection

If your page looks like below, you have successfully set up SSL on your IBM i!





**Digital Certificate Manager** IBM

**Current Certificate Store**

You have selected to work with the certificate store listed below. The left frame is being refreshed to show the task list for this certificate store. Select a task from the left frame to begin working with this certificate store.

**Certificate type:** Server or client  
**Certificate store:** \*SYSTEM  
**Certificate store path and filename:** /QIBM/USERDATA/ICSS/CERT/SERVER/DEFAULT.KDB

Select a Certificate Store

Expand All Collapse All

- ▶ Fast Path
- Create Certificate
- Create New Certificate Store
- Create a Certificate Authority (CA)
- ▶ Manage Certificates
- ▶ Manage Applications
- ▶ Manage Certificate Stores
- ▶ Manage CRL Locations
- Manage PKIX Request Location
- Return to Home Tasks

Secure Connection



## 1.5.3 Installing SSL Certificate Authorities on IBM i

---

Published: 2024-05-15

### Retrieving the SSL CA Certificates

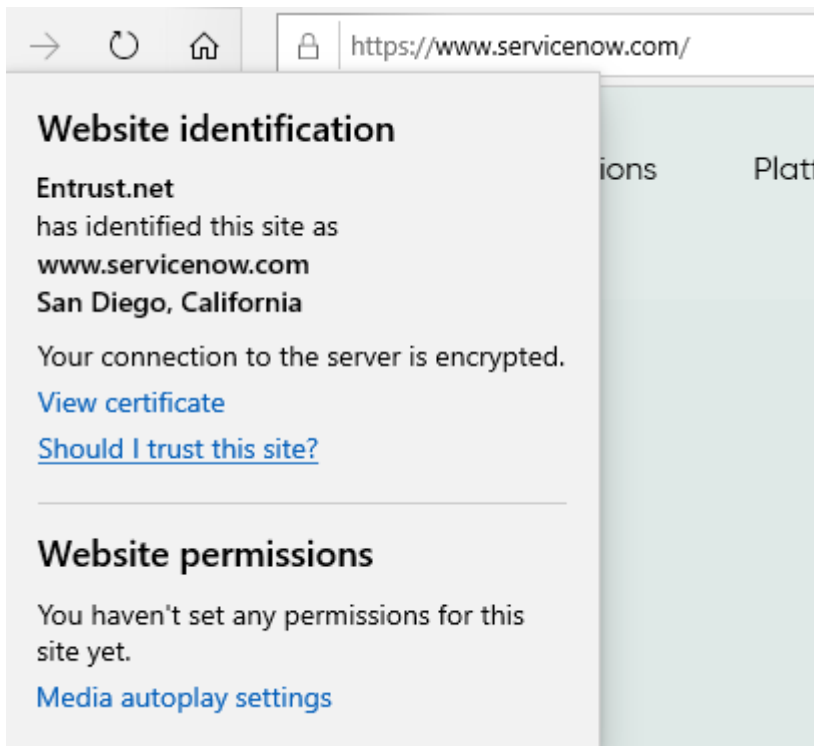
For each site accessed using a REST Consumer created with the MDRest4i SDK to connect to a REST or SOAP service over SSL, you may need to install Certificate Authority Certificates (CA's) that validate the server certificate returned by the end point you are connecting to.

To obtain the certificate go to the URL/URI using your web browser. These examples were created using Microsoft Edge version 44 and Chrome version 78. For the example we used the ServiceNow.com website. From either Edge or Chrome navigate to this url:

<https://www.servicenow.com/>

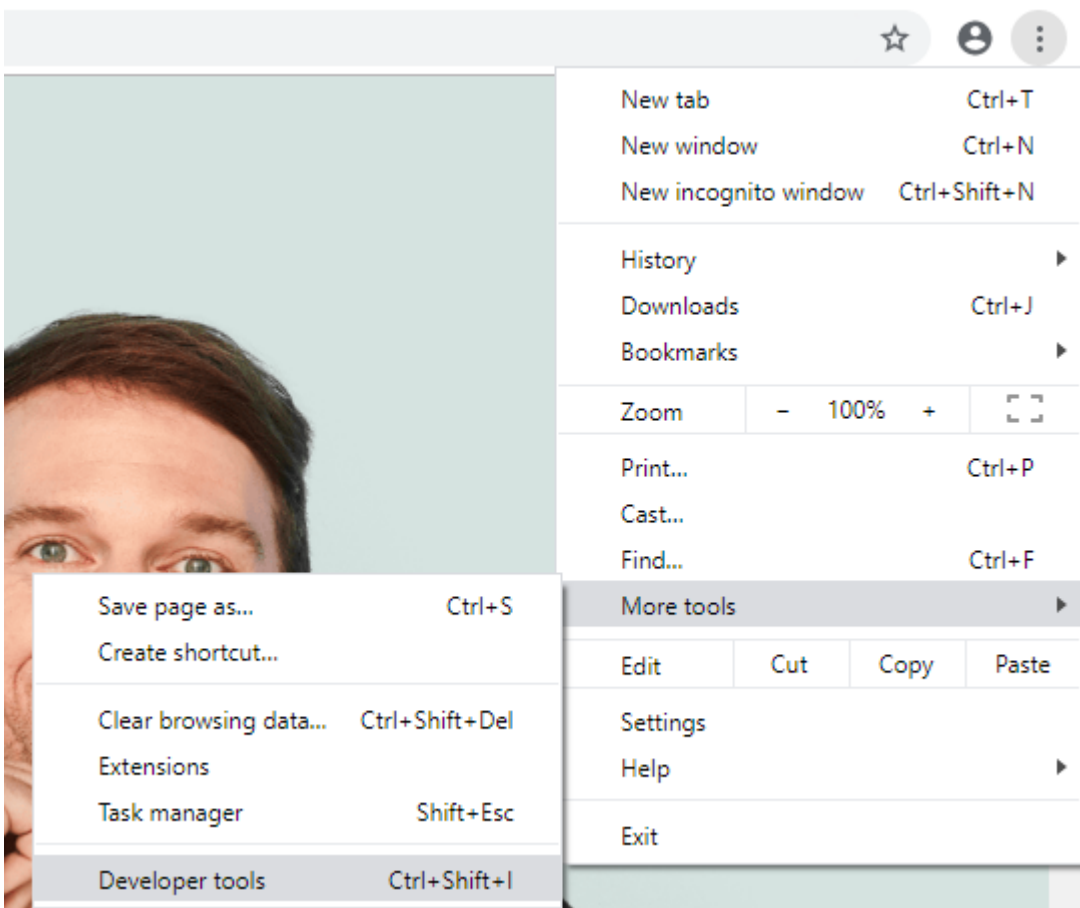
Microsoft Edge

Click the padlock at the right-hand side of the URL bar, then click on "View certificate":

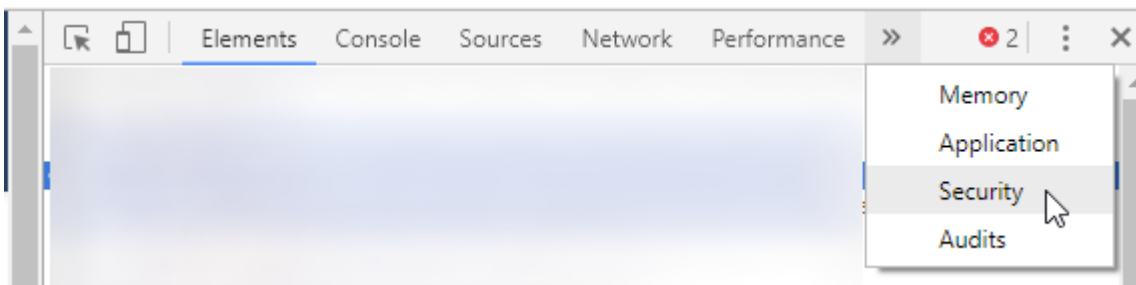


Chrome

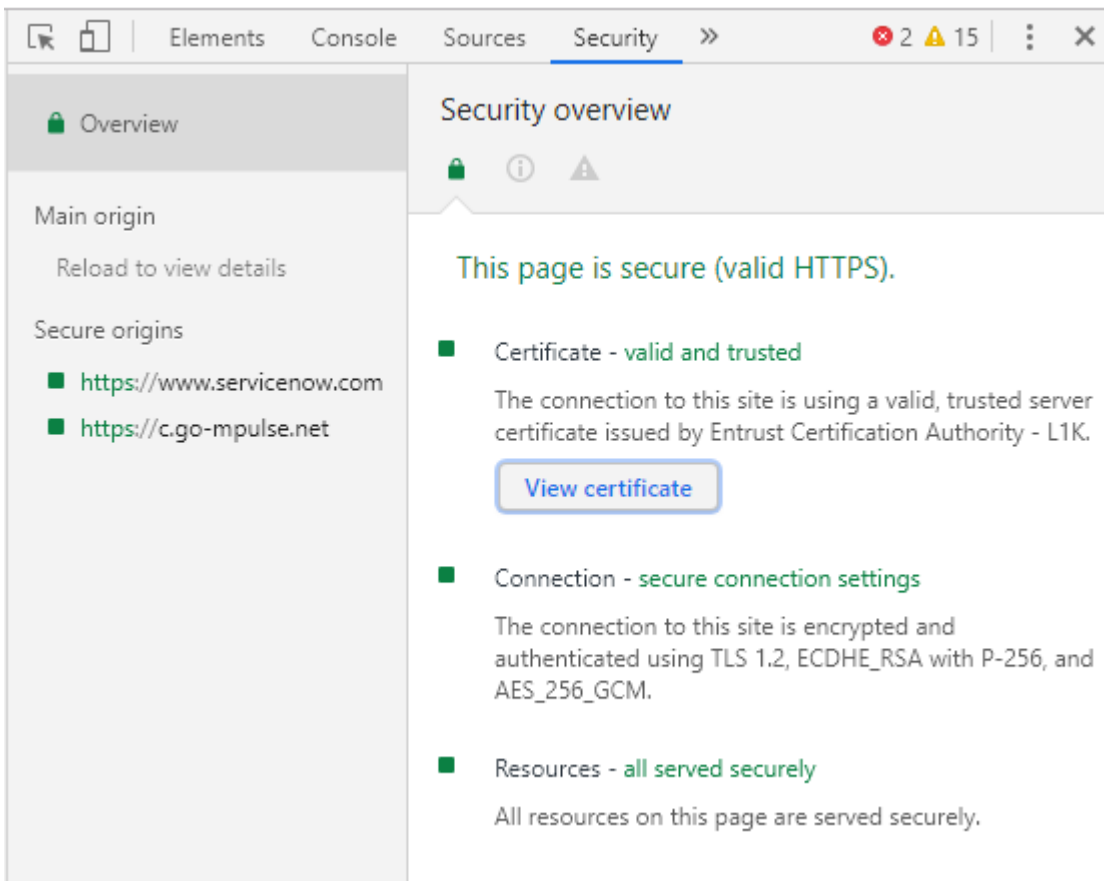
From the Menu, go to "More Tools" > "Developer Tools":



In the developer tools frame, to go the “Security” tab:

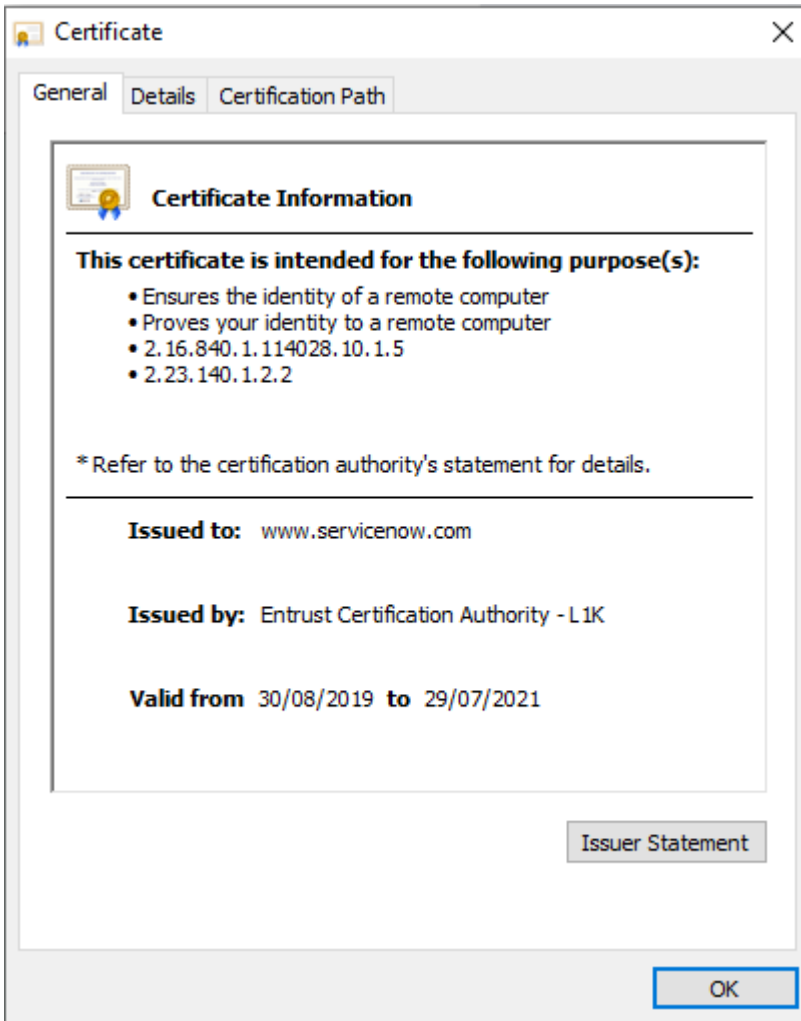


Then, click on “View certificate”:

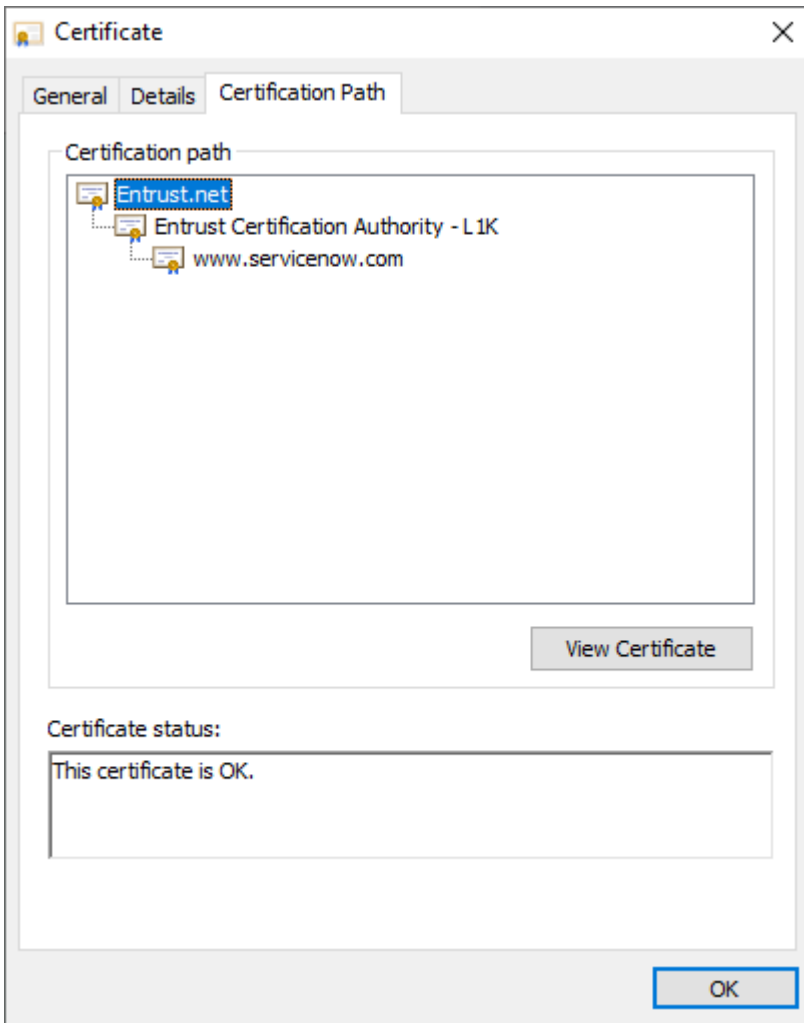


### Downloading the Certificates

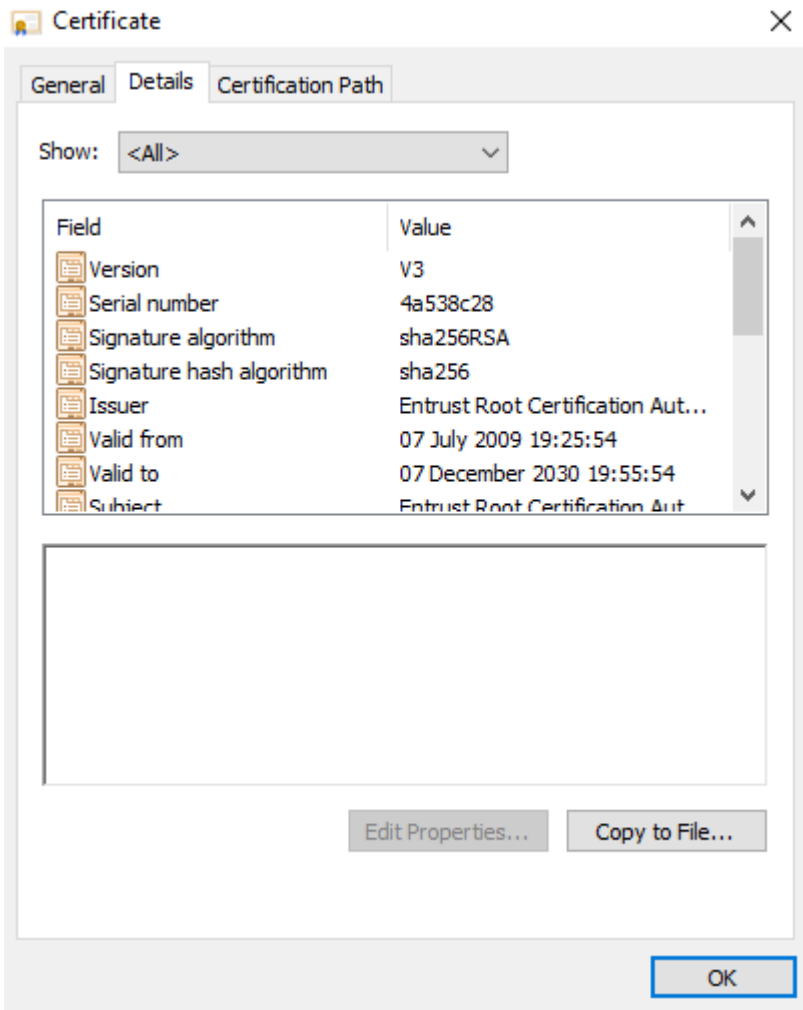
From the Certificate Information window:



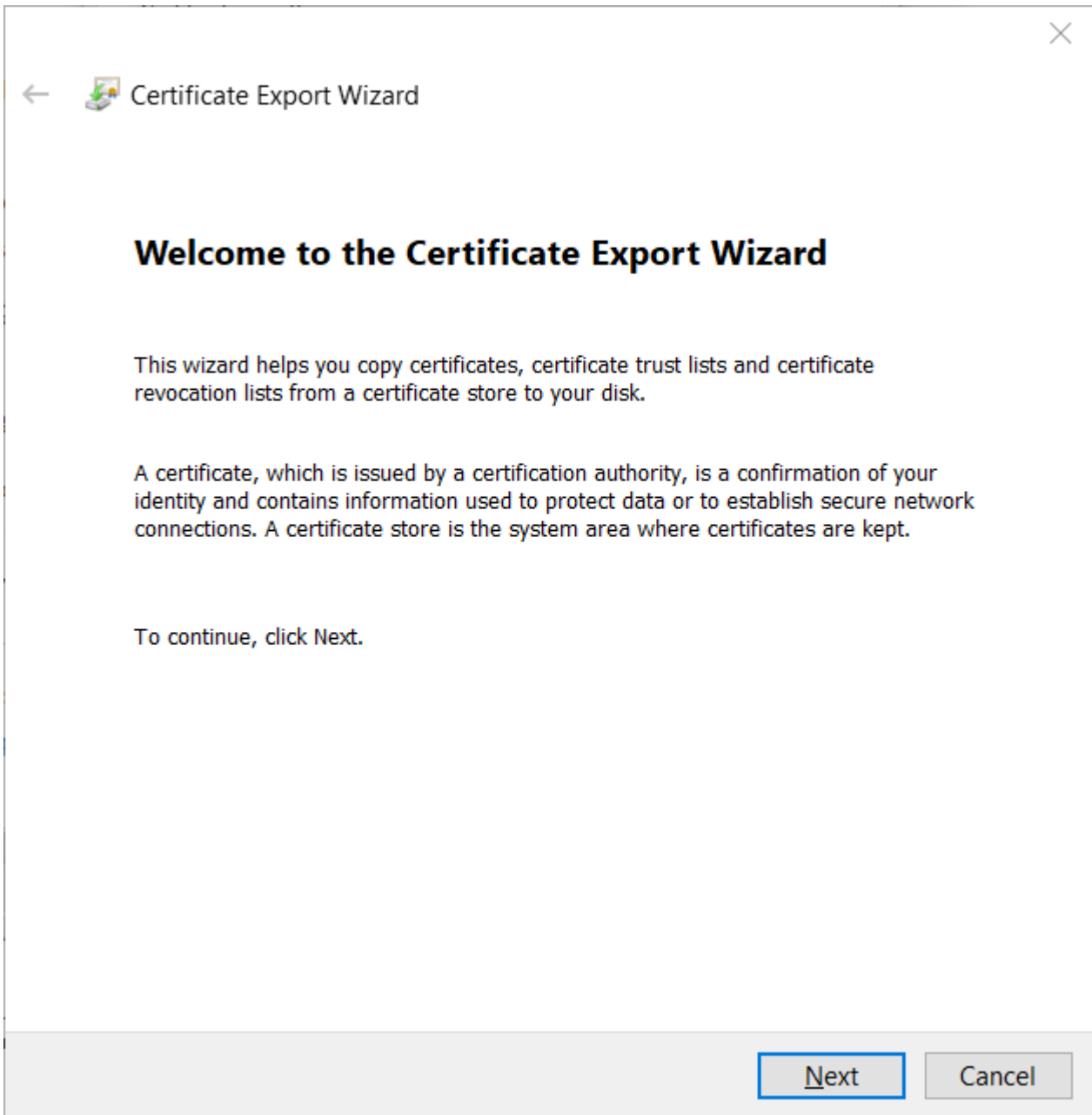
Click on the "Certification Path" tab to view the different signers of this certificate. In this case we have two signers, Entrust.net and Entrust Certificate Authority dow- L1k (also known as an intermediary CA). With Entrust.net highlighted, select the "View Certificate" button.



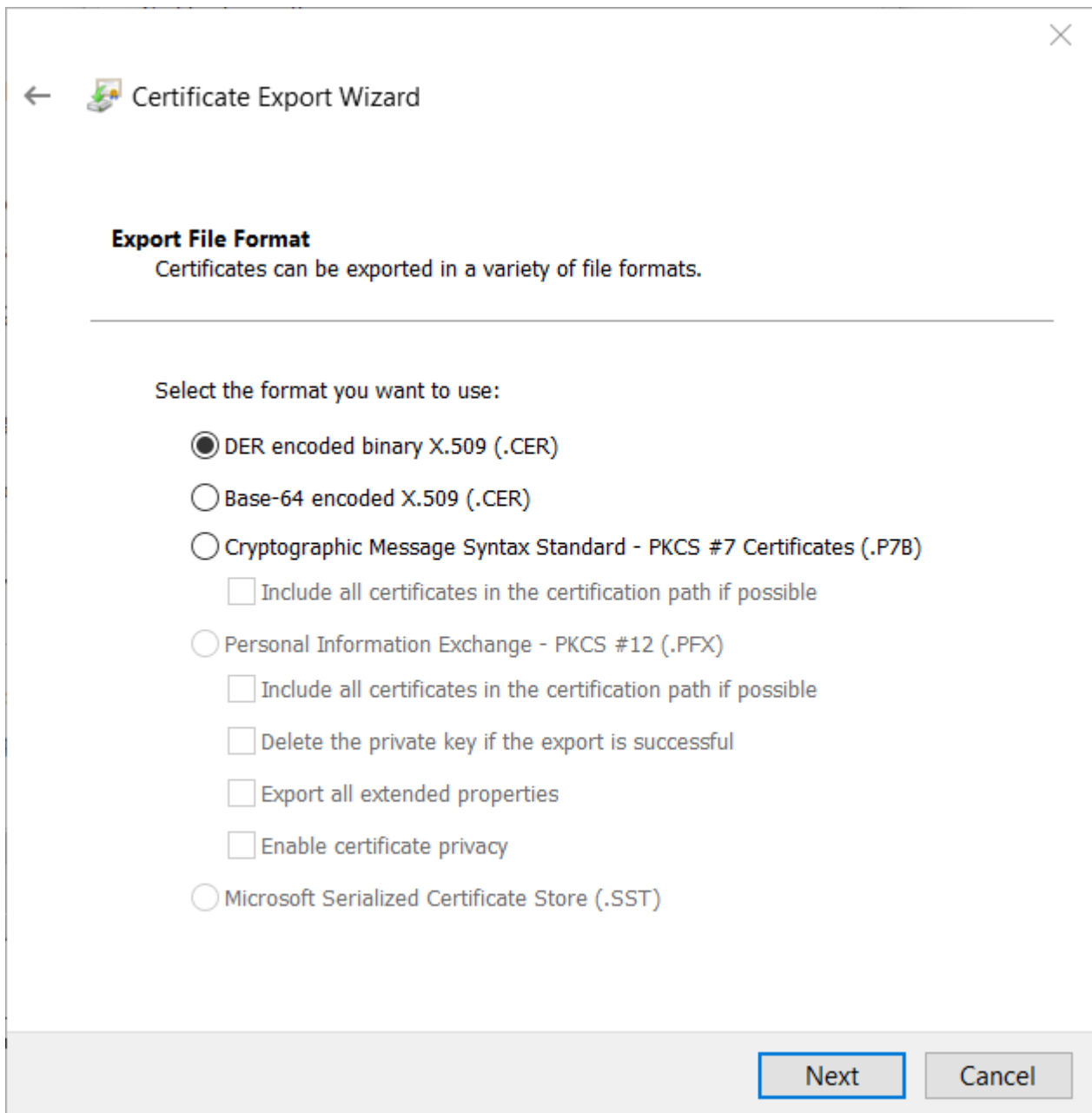
You should now be presented with a new Certificate window, and upon clicking the "Details" tab should see a window like the below image. Select the "Copy to File" button which will start you into a wizard process:



Select the "Next" button.



Leave "DER encoded binary X.509 (.CER)" selected, and select the "Next" button.



The image shows a Windows-style dialog box titled "Certificate Export Wizard". It has a back arrow on the left and a close 'X' button on the top right. The main content area is titled "Export File Format" and contains the text "Certificates can be exported in a variety of file formats." Below this is a horizontal line and the instruction "Select the format you want to use:". There are seven radio button options: "DER encoded binary X.509 (.CER)" (selected), "Base-64 encoded X.509 (.CER)", "Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)", "Personal Information Exchange - PKCS #12 (.PFX)", and "Microsoft Serialized Certificate Store (.SST)". Under the "Cryptographic Message Syntax Standard" and "Personal Information Exchange" options, there are three checkboxes: "Include all certificates in the certification path if possible", "Delete the private key if the export is successful", and "Export all extended properties". Under the "Personal Information Exchange" option, there is also a checkbox for "Enable certificate privacy". At the bottom right, there are two buttons: "Next" (highlighted with a blue border) and "Cancel".

← Certificate Export Wizard

**Export File Format**  
Certificates can be exported in a variety of file formats.

---

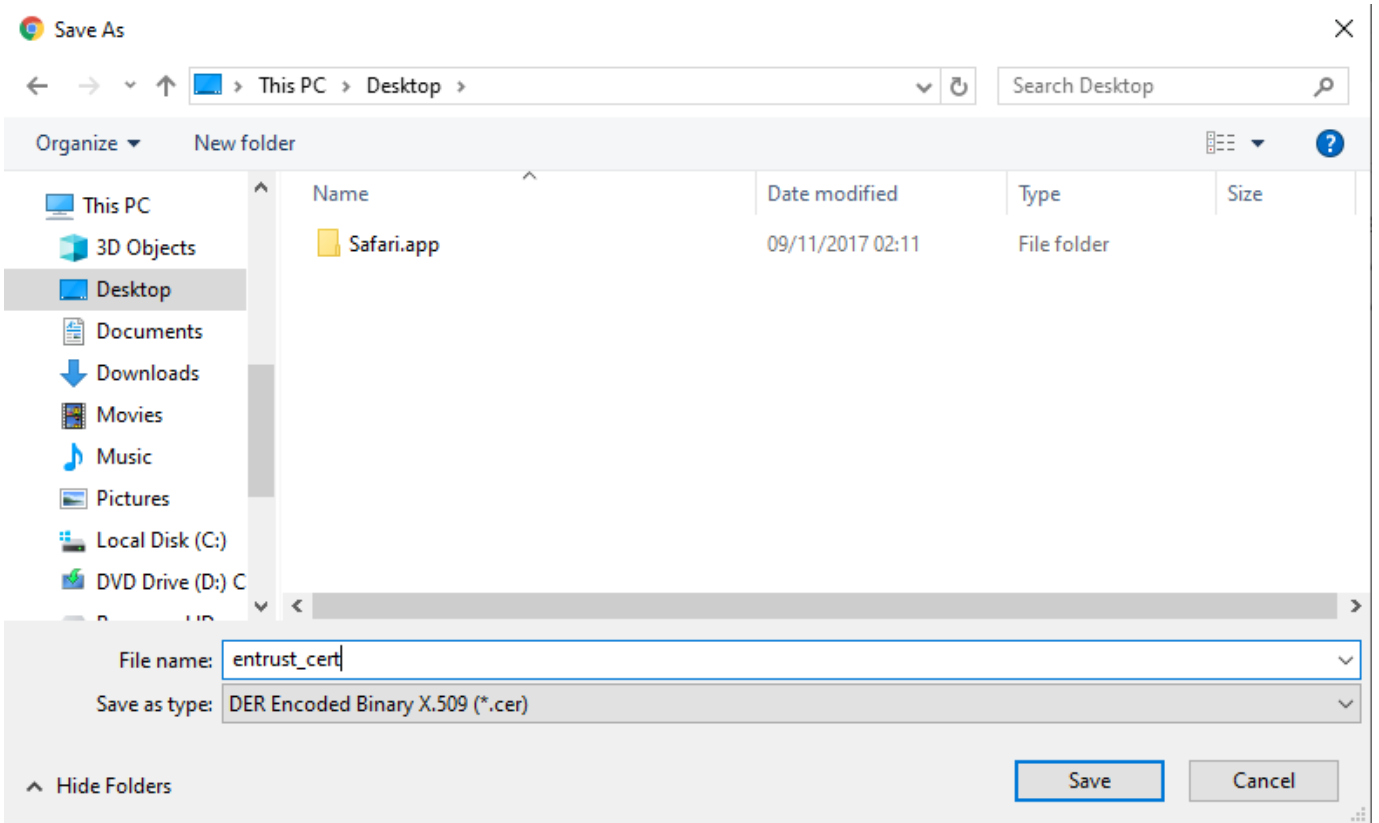
Select the format you want to use:

- DER encoded binary X.509 (.CER)
- Base-64 encoded X.509 (.CER)
- Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
  - Include all certificates in the certification path if possible
- Personal Information Exchange - PKCS #12 (.PFX)
  - Include all certificates in the certification path if possible
  - Delete the private key if the export is successful
  - Export all extended properties
  - Enable certificate privacy
- Microsoft Serialized Certificate Store (.SST)

Next Cancel

You will be prompted to save the file. Browse to your Desktop and save it with a recognizable name as shown in the following screen shot:





Now repeat this process by closing all dialogs and returning to the original browser window. This time after clicking on the padlock, certificate, certification path, select Entrust Certificate Authority - L1k and then the “View Certificate” button and then details. Save this certificate to the desktop as well.

Upload the .cer files to your IBM i via FTP (or other means) and place them in /home (or other folder of choice, just remember where you put it).

### Applying the Certificates

To begin, verify that the \*ADMIN HTTP server job is running with the following command:

```
WRKSBSJOB SBS(QHTTSPVR)
```

If you don't see \*ADMIN in the list, please run the following command to start it:

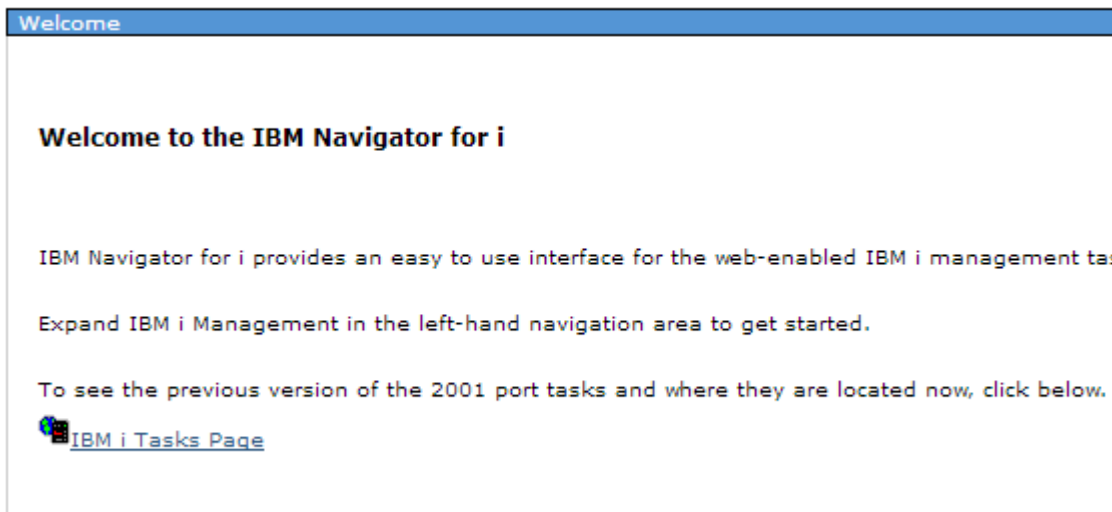
```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

After you've ensured that the \*ADMIN server is running, open a web browser (Internet Explorer is recommended), and go to [http://\[YourIBMIAddress\]:2001](http://[YourIBMIAddress]:2001) - you should see a login page as seen below:



The image shows the login screen for IBM Navigator for i. It features a blue header with the IBM logo in the top right corner. The main content area is a light blue box with a white border. Inside this box, the text "IBM® Navigator for i" is displayed at the top left. Below this text is a circular icon containing a lowercase 'i' and the words "for Business". To the right of the icon are two input fields: "User ID:" followed by a white text box, and "Password:" followed by another white text box. A "Log in" button is positioned to the right of the password field.

Enter your IBM i username and password, and click "Log in". You should see a page split into two sections - a menu on the left, and a larger content area on the right that looks like the below image:





The image shows a screenshot of the IBM Navigator for i welcome page. It has a blue header bar with the word "Welcome" in white. Below the header, the text "Welcome to the IBM Navigator for i" is displayed in bold. The main content area contains the following text: "IBM Navigator for i provides an easy to use interface for the web-enabled IBM i management tas", "Expand IBM i Management in the left-hand navigation area to get started.", and "To see the previous version of the 2001 port tasks and where they are located now, click below." Below this text is a link labeled "IBM i Tasks Page" with a small icon to its left.


Click the "IBM i Tasks Page" link.


IBM i Tasks - Green.sym-corp.com

IBM i Tasks allows you to access the tasks that were previously displayed on the IBM i Tasks web page.

 [IBM Web Administration for i](#)  
Allows you to manage and configure HTTP servers and application servers (Located in Internet Configurations)

 [Digital Certificate Manager](#)  
Allows you to create, distribute, and manage Digital Certificates (Located in Internet Configurations)

 [IBM Tivoli Directory Server Web Administration Tool](#)  
Allows you to administer the IBM Tivoli Directory Server for i (Located in Network)

 [IBM IPP Server for i](#)  
Allows you to configure the IBM IPP Server (Located in Internet Configurations)

Now, click the “Digital Certificate Manager” link. You may be prompted to log in again - if you are, enter your IBM i username and password. It is recommended to log into the Digital Certificate Manager on a profile with elevated authority.

After you are logged in, click on the “Select a Certificate Store” button in the far left of the page. Then, select the \*SYSTEM store and press the “Continue” button. If you do not see \*SYSTEM, you will need to [go set up SSL](#) on your IBM i.

## Select a Certificate Store

Select the certificate store that you want to open.

- \*SYSTEM  
 Other System Certificate Store

It will then prompt you for your \*SYSTEM store password. Enter your password and select the “Continue” button. Note: If you do not remember the password, you can simply select “Reset Password” - you will be allowed to reset the password without knowing the previous password.

## Certificate Store and Password

Enter the certificate store password.

Certificate type: Server or client  
 Certificate store: \*SYSTEM  
 Certificate store path and filename: /QIBM/USERDATA/ICSS/CERT/SERVER/DEFAULT.KDB  
 Certificate store password:

Next, select “Manage Certificates” on the left:



Click "Import Certificate"

Select "Certificate Authority", and then click "Continue":

## Import Certificate

**Certificate store:** \*SYSTEM

Select the type of certificate that you want to import.

- Server or client**  
 **Certificate Authority (CA)**

Enter the IFS file path of the certificate you are importing. It is very typical that there will be multiple levels of SSL certificates arranged in a "chain". If this is the case, you need to import the highest level first. In this case we need to import DST\_cert.cer, and then LetsEncrypt\_cert.cer. Below shows how to import the DST\_cert.cer certificate.

## Import Certificate Authority (CA) Certificate

**Certificate type:** Certificate Authority (CA)

**Certificate store:** \*SYSTEM

Specify the fully qualified path and file name of the certificate that you want to import.

Example path and file name: /MYDIRECTORY/MYFILE.EXT

**Import file:**

Continue

Cancel

You will be prompted to enter a label for the certificate. The label you choose doesn't matter, but it's recommended to choose a label that describes the certificate you're uploading. Then, click "Continue".

## Import Certificate Authority (CA) Certificate

**Certificate type:** Certificate Authority (CA)

**Certificate store:** \*SYSTEM

Specify a label for the certificate.

**CA certificate label:**

Continue

Cancel

At this point, you will likely receive one of two messages. The first possible message looks like the below image. This indicates that someone has already imported this certificate into your IBM i's \*SYSTEM store. In this case, your work is done for this certificate - move onto the next one.

## Import Certificate Authority (CA) Certificate

**Message** A duplicate key exists in the certificate store. The certificate or the label may already be in the certificate store. The label must be unique.

OK

Otherwise, you should receive a message indicating that the certificate has been successfully imported.

## Import Certificate Authority (CA) Certificate

Message The certificate has been imported.  
Use the Manage Applications task if you want to specify that applications trust this Certificate Authority (CA).

OK

Now, repeat the process for each certificate you uploaded to your IBM i.

## 1.5.4 Setting SSL Store Permissions

Published: 2024-05-15

When calling a remote web service that uses SSL (i.e. the URL starts with https) MDRest4i is using the SSL related components of the IBM i GSK API's underneath the covers, and in turn those API's are accessing system objects that are locked up fairly tight by default.

The most common errors are with SSL permission issues (you can see errors in the job log of the user trying to run the consumer program).

To correct this you need to provide the \*PUBLIC profile access to what are called the keyring SSL files. You can locate your keyring files in the IFS by running the following command:

```
WRKLNK '/QIBM/UserData/ICSS/Cert/Server'
```

You should then see files DEFAULT.KDB and DEFAULT.RDB as show in the below screen shot. If you don't see those files then you probably have not run [SSL setup on your IBM i](#) yet. If so, do this first.

```
Work with Object Links
Directory . . . . . : /QIBM/UserData/ICSS/Cert/Server
Type options, press Enter.
 2=Edit   3=Copy   4=Remove   5=Display   7=Rename   8
11=Change current directory ...

Opt  Object link      Type      Attribute  Text
---  -
 9   DEFAULT.KDB      STMF
 9   DEFAULT.RDB      STMF
```

Take an option 9 on each of the DEFAULT.\* files and give \*PUBLIC a Data Authority of \*R as shown in the below screen shot:

```
Work with Authority
Object . . . . . : /QIBM/UserData/ICSS/Cert/Server/DEFAULT.* >
Owner . . . . . : QSYS
Primary group . . . . . : *NONE
Authorization list . . . . . : *NONE

Type options, press Enter.
 1=Add user  2=Change user authority  4=Remove user

opt  User      Data Authority  --Object Authorities--
    User      Authority  Exist Mgt Alter Ref
---  -
 9   *PUBLIC  *R
 9   QSYS     *RW          X      X      X      X
```

The last step is to change the Data Authority on folder /QIBM/UserData/ICSS/Cert/Server for profile \*PUBLIC to be \*RX. Use the following command to view the .../Server folder:

```
WRKLNK '/QIBM/UserData/ICSS/Cert/Server'
```

Next take option 9 on the .../Server folder and give \*PUBLIC a Data Authority of \*RX as shown in the screen shot below:

```

work with Authority

object . . . . . : /QIBM/UserData/ICSS/Cert/Server
owner . . . . . : QSYS
Primary group . . . . . : *NONE
Authorization list . . . . . : *NONE

Type options, press Enter.
  1=Add user   2=Change user authority  4=Remove user

opt  User          Data          --Object Authorities--
     User          Authority  Exist  Mgt  Alter  Ref
-----
=   *PUBLIC       *RX
-   QSYS          *RWX

```

That's it! Now go back and re-run the program under the profile that was having permission issues to ensure it is running correctly.



## 1.5.5 Create a TLS-SSL DCM Application

---

Published: 2024-05-15

An SSL Application in the IBM i DCM is used to assign certificates, specify cyphers and encryption algorithms. These settings are then used by the HTTP server when negotiating the SSL tunnel with a remote SSL server or client for a request.

### Create the SSL Application

#### Warning

A user profile with \*IOSYSCFG authority is required for these setup tasks.

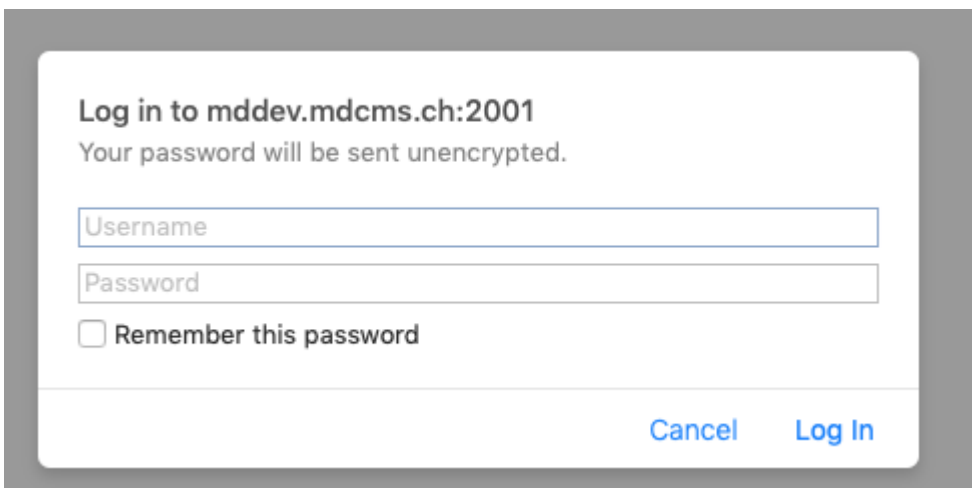
To begin, verify that the \*ADMIN HTTP server job is running with the following command:

```
WRKSBSJOB SBS(QHTTSPVR)
```

If you don't see \*ADMIN in the list, please run the following command to start it:

```
STRTCPSVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

After you've ensured that the \*ADMIN server is running, open a web browser (Microsoft Edge or Chrome is recommended), and go to [http://\[youribmiserver\]:2001/HTTPAdmin](http://[youribmiserver]:2001/HTTPAdmin) - you should see a login page as seen below:



Log in to mddev.mdcms.ch:2001  
Your password will be sent unencrypted.

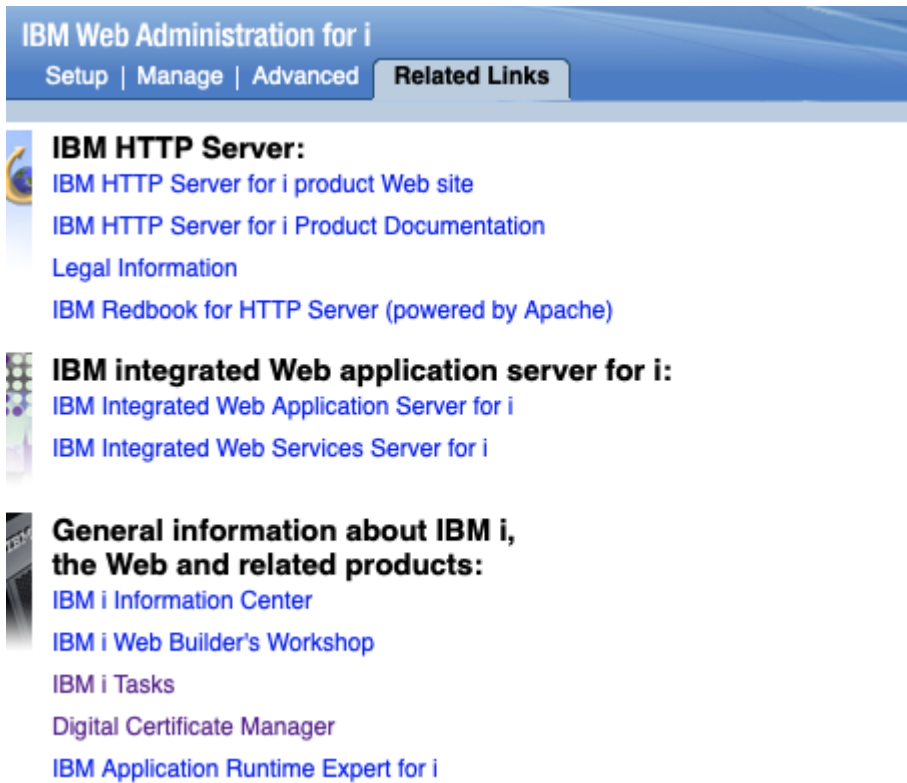
Username

Password

Remember this password

Cancel Log In

from the landing page elect (if not already selected) the "related links tab" which will bring up the page below:



IBM Web Administration for i

Setup | Manage | Advanced | **Related Links**

**IBM HTTP Server:**

- [IBM HTTP Server for i product Web site](#)
- [IBM HTTP Server for i Product Documentation](#)
- [Legal Information](#)
- [IBM Redbook for HTTP Server \(powered by Apache\)](#)

**IBM integrated Web application server for i:**

- [IBM Integrated Web Application Server for i](#)
- [IBM Integrated Web Services Server for i](#)

**General information about IBM i, the Web and related products:**

- [IBM i Information Center](#)
- [IBM i Web Builder's Workshop](#)
- [IBM i Tasks](#)
- [Digital Certificate Manager](#)
- [IBM Application Runtime Expert for i](#)

Now, click the "Digital Certificate Manager" link. You may be prompted to log in again - if you are, enter your IBM i username and password. It is recommended to log into the Digital Certificate Manager on a profile with elevated authority.

After you are logged in, click on the "Select a Certificate Store" button in the far left of the page. Then, select the \*SYSTEM store and press the "Continue" button. If you do not see \*SYSTEM, you will need to go [set up SSL on your IBM i](#).

## Select a Certificate Store

Select the certificate store that you want to open.

- \*SYSTEM
- Other System Certificate Store

Continue

Cancel

It will then prompt you for your \*SYSTEM store password. Enter your password and select the "Continue" button. Note: If you do not remember the password, you can simply select "Reset Password" - you will be allowed to reset the password without knowing the previous password.

## Certificate Store and Password

Enter the certificate store password.

**Certificate type:** Server or client

**Certificate store:** \*SYSTEM

**Certificate store path and filename:** /QIBM/USERDATA/ICSS/CERT/SERVER/DEFAULT.KDB

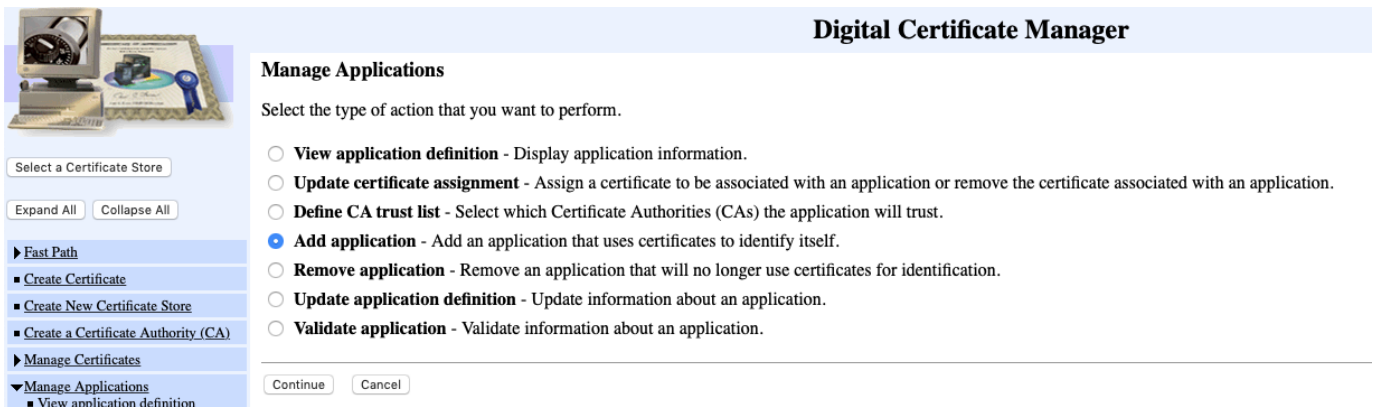
**Certificate store password:**

Continue

Reset Password

Cancel

Next, select "Manage Applications" on the left, and then select "Add Application" and the continue button:



**Digital Certificate Manager**

**Manage Applications**

Select the type of action that you want to perform.

- View application definition** - Display application information.
- Update certificate assignment** - Assign a certificate to be associated with an application or remove the certificate associated with an application.
- Define CA trust list** - Select which Certificate Authorities (CAs) the application will trust.
- Add application** - Add an application that uses certificates to identify itself.
- Remove application** - Remove an application that will no longer use certificates for identification.
- Update application definition** - Update information about an application.
- Validate application** - Validate information about an application.

Continue Cancel

Then select "Server" and the continue button:

### Add Application

Select the type of application that you want to add.

- Server** - Add a server application
- Client** - Add a client application

Continue Cancel

Now add a suitable name for the application and select Application description and provide an applicable one (Kindly consult your networking/infrastructure staff for an appropriate name):

**Add Application**

Application type: Server

Application ID:

Enter either the application description message information or an application description.

Application description message information

<input type="radio"/>	Message file:	<input type="text"/>
<input type="radio"/>	Message file library:	<input type="text"/>
<input type="radio"/>	Message ID:	<input type="text"/>
<input checked="" type="radio"/>	Application description:	<input type="text" value="QISM_HTTP_SERVER_DEF"/>

#### Warning

Please copy or make a note of the application ID (not description) used. This Application ID is a mandatory value when enabling an HTTP server for HTTPS!

Scroll down to "SSL protocols" and select the appropriate protocols and versions. (Kindly consult your networking/infrastructure staff for which to select):

SSL protocols	
<input type="radio"/>	*PGM
<input checked="" type="radio"/>	Define protocols supported:
<input checked="" type="checkbox"/>	TLS 1.2
<input checked="" type="checkbox"/>	TLS 1.1
<input type="checkbox"/>	TLS 1.0
<input checked="" type="checkbox"/>	SSL 3.0
<input type="checkbox"/>	SSL 2.0

Scroll down to "SSL cipher specification options" and select "Define cipher specification list:" and leave default sort order(Kindly consult your networking/infrastructure staff for which to select):

SSL cipher specification options	
<input type="radio"/>	*PGM
<input checked="" type="radio"/>	Define cipher specification list:
	<b>Order</b>
ECDHE_ECDSA_AES_128_CBC_SHA256	1
ECDHE_ECDSA_AES_256_CBC_SHA384	2
ECDHE_ECDSA_AES_128_GCM_SHA256	3
ECDHE_ECDSA_AES_256_GCM_SHA384	4
RSA_AES_128_CBC_SHA256	5
RSA_AES_128_CBC_SHA	6
RSA_AES_256_CBC_SHA256	7
RSA_AES_256_CBC_SHA	8
RSA_AES_128_GCM_SHA256	9
RSA_AES_256_GCM_SHA384	10
ECDHE_RSA_AES_128_CBC_SHA256	11
ECDHE_RSA_AES_256_CBC_SHA384	12
ECDHE_RSA_AES_128_GCM_SHA256	13
ECDHE_RSA_AES_256_GCM_SHA384	14
ECDHE_ECDSA_3DES_EDE_CBC_SHA	15

Scroll down to "Define CA Trust list:" and select "Yes":

Define the CA trust list:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Certificate Revocation List (CRL) processing:	<input checked="" type="radio"/> Yes <input type="radio"/> No

Scroll down to "SSL Signature Algorithms" and select "Define signature algorithms supported:" (Kindly consult your networking/infrastructure staff for which to select) :

SSL signature algorithms	
<input type="radio"/> *PGM	
<input checked="" type="radio"/> Define signature algorithms supported:	Order
ECDSA_SHA512	1 <input type="button" value="v"/>
ECDSA_SHA384	2 <input type="button" value="v"/>
ECDSA_SHA256	3 <input type="button" value="v"/>
ECDSA_SHA224	4 <input type="button" value="v"/>
ECDSA_SHA1	5 <input type="button" value="v"/>
RSA_SHA512	6 <input type="button" value="v"/>
RSA_SHA384	7 <input type="button" value="v"/>
RSA_SHA256	8 <input type="button" value="v"/>
RSA_SHA224	9 <input type="button" value="v"/>
RSA_SHA1	10 <input type="button" value="v"/>
RSA_MD5	11 <input type="button" value="v"/>

Finally elect the "Add" button. The following screen should appear:

## Add Application

Message **The application has been added.**

OK

### Update Certificate Assignment

Next a certificate from the system store must be allocated to the Application you have created above.

from the "Manage Applications" menu on the left, select "Update Certificate Assignment". from the screen that appears select "Server" and the continue button

#### Update Certificate Assignment

Select the type of application that you want to update.

- Server** - Add, change, or remove certificate assignment for a server application
- Client** - Add, change, or remove certificate assignment for a client application

Continue Cancel

from the next screen that appears, select the server application created above, and the "Update Certificate Assignment" button:

### Update Certificate Assignment

**Application type:** Client

Select the application that you want to update.

	Application	Certificate Assigned
<input type="radio"/>	IBM i TCP/IP Telnet Client	None assigned
<input type="radio"/>	IBM i DDM/DRDA Client - TCP/IP	None assigned
<input type="radio"/>	IBM Directory Server publishing	None assigned
<input type="radio"/>	IBM Directory Server client	None assigned
<input type="radio"/>	Enterprise Identity Mapping (EIM)	None assigned
<input type="radio"/>	IBM i Remote Journaling Source	None assigned
<input type="radio"/>	IBM i TCP/IP SMTP Client	None assigned
<input type="radio"/>	IBM i TCP/IP FTP Client	None assigned
<input checked="" type="radio"/>	Service Now Application	None assigned
<input type="radio"/>	Default Client App for MDRest4i	Comodo PositiveSSL 2019

**Note:** Anytime you change certificate selections, you may need to end your client and start it again to have the change take effect.

Update Certificate Assignment Cancel

from the next screen below, select an appropriate certificate and the "Update certificate Assignment" button. This assumes you have already [Setup SSL](#) on your IBM i:

### Update Certificate Assignment

**Application type:** Client  
**Application ID:** SERVICENOW  
**Application description:** Service Now Application

Select up to four certificates that you want to assign to the application.

**Warning:** When you assign a certificate to a client application and a server requests client authentication, then the server authenticates all users of the application based on that certificate. Consequently, the server does not authenticate users on an individual basis. To ensure that the server authenticates each user of a client application individually outside the SSL protocol, do not assign a certificate to the client application.

Certificate	Common name		
<input checked="" type="checkbox"/> Comodo PositiveSSL 2019	mddev.mdems.ch	View	Validate
<input type="checkbox"/> Comodo PositiveSSL 2017	mddev.mdems.ch	View	Validate

**Note:** Anytime you change certificate selections, you may need to end your client and start it again to have the change take effect.

**Note:** If you assign more than one certificate, the system determines which one to use during SSL session establishment based upon protocol information negotiated with the peer.

Update Certificate Assignment Cancel

From here you should receive this message:

### Update Certificate Assignment

**Message** The certificate was assigned to the application.

**Application type:** Client  
**Application ID:** SERVICENOW  
**Application description:** Service Now Application

**Define CA Trust List (optional)**

This step is optional. Kindly consult your networking/infrastructure staff for which to select.

Start by selecting "Define CA Trust List" from the "Manage Applications" menu on the left:

from the window that appears, select "Server" and the "Continue" button:



Now select the recently created application, and the "Define CA Trust list" button:



From the list that appears select the CA certificates recently imported (the Entrust ones for example from ServiceNow):

**Define CA Trust List**

**Application type:** Client

**Application ID:** SERVICENOW

**Application description:** Service Now Application

Only define a CA Trust List if the list will be a subset of the eligible CAs. See the online help for more information.

The Certificate Authorities (CAs) defined in the CA trust list for the application are checked. If you wish to change the trust list, click on the check box and select OK.

<input type="checkbox"/>	Certificate Authority (CA)		
<input type="checkbox"/>	SectigoRSADomainValidationSecureServerCA	View	Validate
<input type="checkbox"/>	Comodo User Trust cert	View	Validate
<input checked="" type="checkbox"/>	EntrustCertificationAuthorityL1K	View	Validate
<input checked="" type="checkbox"/>	EntrustRootCertificationAuthorityG2	View	Validate
<input checked="" type="checkbox"/>	EntrustRoot	View	Validate
<input type="checkbox"/>	Comoda CA DV	View	Validate

The following validation message should appear:

---

## Define CA Trust List

Message Certificate Authority (CA) changes applied.

**Application type:** Client

**Application ID:** SERVICENOW

**Application description:** Service Now Application

SSL for a client/consumer on IBM i is now complete.



## 1.5.6 Enable HTTPS for an HTTP Server Instance

Published: 2024-05-14

### REQUIREMENTS

- TLS/SSL already configured on the IBM i.
- User profile with **\*IOSYSCFG** and **\*SECADM** authority
- **\*ADMIN** HTTP Server instance to be started
- Web browser access to IBM i on port number **TCP port 2001**
- **DCM Application ID** used when [creating the DCM Application during TLS/SSL setup](#)

### Warning

A user profile with \*IOSYSCFG authority is required for these setup tasks

### OPEN HTTP SERVER INSTANCE EDITOR

To begin, verify that the \*ADMIN HTTP server job is running with the following command:

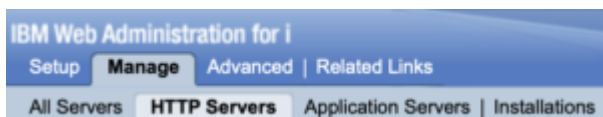
```
WRKSBSJOB SBS(QHTTSPVR)
```

If you don't see \*ADMIN in the list, please run the following command to start it:

```
STRTCPVR SERVER(*HTTP) HTTPSVR(*ADMIN)
```

After you've ensured that the \*ADMIN server is running, open a web browser (Microsoft Edge or Chrome is recommended), and go to [http://\[youribmiserver\]:2001/HTTPAdmin](http://[youribmiserver]:2001/HTTPAdmin) - you should see a login page as seen below:

Select the "Manage" and "HTTP Servers" tabs in the top left hand part of the window after logging in.



From the drop down box just below the "Manage" tab select the server you wish to enable for HTTPS



In the left hand part of the window, expand the "Tools" menu and select "Edit Configuration File" from the available options. An editor window will appear on the right hand side.



#### ADD TLS-SSL CONFIG ENTRIES

On the second line of the configuration, paste the following value:

```
LoadModule ibm_ssl_module /QSYS.LIB/QHTTPSVR.LIB/QZSRVSSL.SRVPGM
```

Scroll down the configuration data until the "**Listen \*...**" is reached. Edit this line to:

```
Listen *:443
```

#### Warning

Port number 443 is the default port number for HTTPS. If ANY other server on this IBM i uses port 443, this server instance will not start. Consult your networking/infrastructure staff to ensure this is the correct port number, and it is not in use on this IBM i server already.

Edit the following values and paste on the line AFTER the "Listen \*:443" entry above. **QIBM\_HTTP\_SERVER** should be replaced with the [DCM Application ID](#) used to setup SSL on the IBM i.

```
<VirtualHost *:443>
  SSLEngine On
  SSLAppName QIBM_HTTP_SERVER
  SSLProtocolDisable SSLv3 TLSv1 TLSv1.1
</VirtualHost>
```

Click "OK" or "Apply" at the bottom of the editor window to save these settings.

Restart the server and attempt to connect once again using HTTPS instead of HTTP in the address bar. For example:

```
https://[youribmiserver]/mdcms/applications
```

The above URL should display the following JSON in the browser window:

```
{ "Error": "No authorization token received" }
```

This indicates that the connection via HTTPS has been successful!

#### Warning

Please remember to update the MDCMS HTTP Server Configuration in MDCMS, and any external web hooks configured to access MDCMS via the MDCMS REST API's