# Integrating with IBM Maximo Manage

 You can integrate IBM Maximo Assist with IBM Maximo Manage. When the products are integrated, you can import work orders from IBM Maximo Manage into IBM Maximo Assist, and you can start IBM Maximo Assist from a IBM Maximo Manage work order.

# Creating a Maximo work order collection.

Before you add Maximo work orders to IBM Maximo Assist, you must create a collection for the work orders.

1. In IBM Maximo Application Suite, launch Assist and launch the Manage documents.
2. Select **Create** to create a collection.
3. In the Create Collection dialog box, enter the collection information. For collection type, specify Maximo Work Orders.
4. Select an Watson™ Discovery project and associated Watson™ Discovery data collection. If there is no associated Watson Discovery data collection, create one.
5. Select **Create** to complete the collection creation.

# Using historical Maximo work orders

 You can download work orders from Maximo to a JSON file. You can then upload the work orders to a Maximo work order collection in Maximo® Equipment Maintenance Assistant.

# Required work order fields

Certain fields are required to use Maximo work orders with IBM Maximo Assist.

The following work order fields are required:

| **Field** | **Description** |
| --- | --- |
| wonum | Work order number. |
| description | Description of the work order. |
| description\_longdescription | Long description of the work order. |
| status | Work order status. |
| status\_description: | Description of the work order status. |
| owner | Owner of the work order. |
| reportdate | Reported date. |
| actstart | Actual start date. |
| targstartdate | Target start date. |
| schedstart | Scheduled start date. |
| rel.failure{failurecode,description} | Failure class of the work order. Related attributes. |
| failurecode | Failure code. |
| description | Failure description. |
| rel.location{location,description,siteid} | Location of the work order. Related attributes. |
| location | Location code. |
| description | Location description. |
| rel.classstructure{classificationid} | Classification of the work order. Related attribute. |
| rel.assignment{laborcode} | Assigned labor code of the work order. Related attribute. |
| rel.asset{assetnum,assettype,description, siteid,rel.classstructure{classificationid}} | Related attributes of the work order asset. |
| assetnum | Asset number. |
| assettype | Asset type. |
| description | Description of the asset. |
| rel.classstructure{classificationid} | Classification of the asset. Related attribute. |
| href | The href link to the work order details in Maximo. |
| siteid | Site id. |
| rel.woactivity{tasked,description} | Related attributes of the work order activity. |

# Downloading work orders by using the Maximo MXAPIWODETAIL API

Use the Maximo MXAPIWODETAIL API to download work orders. You can set filter, query, and sort parameters to download specific sets of work orders.

**User name and password parameters**

You must specify the Maximo® user name and password with the **\_lid** and **\_lpwd** parameters. For example, using \_lid=*id*&\_lpwd=*password*. The user must have an administrator user account.

**Site ID parameter**

By default, work orders from all sites are downloaded. You can use the **siteid** query parameter to select work orders from a specified site.

**Href parameter**

The **href** field is selected by default according to the mxe.oslc.webappurl Maximo system property which indicates the public URI provider of Maximo. The mxe.oslc.webappurl property is in the following format http://*maximo\_server*:*maximo\_server\_port*/*maximo\_context\_root*/oslc. To ensure that this field value is a publicly accessible URI, use one of the following options.

1. Log in Maximo work center with an administrator user account, and change value of the mxe.oslc.webappurl system property to be a publicly accessible URI.
2. Add **--header x-public-uri:** when calling the Maximo REST API MXAPIWODETAIL to download work orders from Maximo server. For example, use the following command.
3. curl --header "x-public-uri:http://*maximo\_server\_port*/
4. *maximo\_server*/oslc" "http://*maximo\_server\_port*/
5. *maximo\_server*/oslc/os/MXAPIWODETAIL?\_lid=*id*&\_lpwd=
6. *password*&lean=1&ignorecollectionref=1&oslc.select=wonum,description,
7. description\_longdescription,status,owner,rel.failure\{failurecode,description\}
8. ,rel.location\{location,description\},reportdate,actstart,targstartdate,schedstart,
9. rel.classstructure\{classificationid\},rel.assignment\{laborcode\},rel.asset\

{assetnum,assettype,description,rel.classstructure\{classificationid\}\}"

**Field parameters**

By using the MXAPIWODETAIL API, you can use the oslc.select query parameter to select the required fields of a work order and to download work logs in addition to the required fields.

Use the following command to select the required fields of a work order.

oslc.select=wonum,description,description\_longdescription,

status,owner,rel.failure{failurecode,description},rel.location{location,

description},reportdate,actstart,targstartdate,schedstart,

rel.classstructure{classificationid},rel.assignment{laborcode},

rel.asset{assetnum,assettype,description,

rel.classstructure{classificationid}}

Use the following command to download work logs in addition to the required fields.

oslc.select=wonum,description,description\_longdescription,status,owner,

rel.failure{failurecode,description},rel.location{location,description},

reportdate,actstart,targstartdate,schedstart,rel.classstructure{classificationid},

rel.assignment{laborcode},rel.asset{assetnum,assettype,description,

rel.classstructure{classificationid}},rel.worklog{\*}

**Additional parameters**

By using the MXAPIWODETAIL API, you can download Maximo work orders and save them to a JSON file. For example, by using the following command, you can download the 10th of 1000 work orders from Maximo server and save it to the test.json file.

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?\_lid=wilson&\_lpwd=wilson&lean

=1&ignorecollectionref=1&pageno=10&oslc.pageSize=1000&oslc.select=wonum,description,

description\_longdescription,status,owner,rel.failure\{failurecode,description\},

rel.location\{location,description\},reportdate,actstart,targstartdate,schedstart,

rel.classstructure\{classificationid\},rel.assignment\{laborcode\},rel.asset\

{assetnum,assettype,description,rel.classstructure\{classificationid\}\},

rel.worklog\{\*\}" -o test.json

You can filter work orders by time range. In the following example, "oslc.where=reportdate>="*date*" and reportdate<"*date*"" are used to filter work orders in a certain date range. This example downloads the first 100 work orders that are reported during the time period from "2009-01-06T00:00:00.000" to "2018-02-06T00:00:00.000".

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?\_lid=wilson&\_lpwd=wilson&lean=1

&ignorecollectionref=1&pageno=1&oslc.pageSize=100&oslc.where=reportdate>=\"2009-01-06T00:

00:00.000\"%20and%20reportdate<\"2018-02-06T00:00:00.000\"&oslc.select=wonum,description,

description\_longdescription,status,owner,rel.failure\{failurecode,description\},

rel.location\{location,description\},reportdate,actstart,targstartdate,schedstart,

rel.classstructure\{classificationid\},rel.assignment\{laborcode\},rel.asset\

{assetnum,assettype,description,rel.classstructure\{classificationid\}\},

rel.worklog\{\*\}" -o test.json

You can also filter work orders by time range by using the **tlrange** and **tlattribute** parameters to filter work orders with time range based query. For example, you can find work orders that are reported in the past 3 months with the query parameter **tlrange=-3M&tlattribute=reportdate**. And using the query parameter **tlrange=+-3Y&tlattribute=schedstart**, you can filter work orders that are scheduled to start 3 years in the past and 3 years in the future indexed on the current date. For example, using the following command, you can download the first 100 work orders that are reported in the past 3 months.

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?\_lid=wilson&\_lpwd=

wilson&lean=1&ignorecollectionref=1&pageno=1&oslc.pageSize=100&tlrange=-3M&tlattribute

=reportdate&oslc.select=wonum,description,description\_longdescription,status,owner,

rel.failure\{failurecode,description\},rel.location\{location,description\},reportdate,

actstart,targstartdate,schedstart,rel.classstructure\{classificationid\},rel.assignment

\{laborcode\},rel.asset\{assetnum,assettype,description,rel.classstructure\

{classificationid\}\},rel.worklog\{\*\}" -o test.json

Using the following command, you can download the first 1000 work orders that are scheduled to start between the last 3 years and the next 3 years from the current date. Note that you must use **%2B** instead of **+** in the curl request.

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?\_lid=wilson&\_lpwd=

wilson&lean=1&ignorecollectionref=1&pageno=1&oslc.pageSize=1000&tlrange=%2B-

3Y&tlattribute=schedstart&oslc.select=wonum,description,description\_longdescription,

status,owner,rel.failure\{failurecode,description\},rel.location\{location,description\}

,reportdate,actstart,targstartdate,schedstart,rel.classstructure\{classificationid\},

rel.assignment\{laborcode\},rel.asset\{assetnum,assettype,description,rel.classstructure\

{classificationid\}\},rel.worklog\{\*\}" -o test.json

To download a specific work order, you can add the **oslc.where=wonum=***workorder\_number* parameter to get the work order. For example, using the following command, you can download work order 43041.

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?\_lid=wilson&\_lpwd=wilson&lean=1&

ignorecollectionref=1&oslc.where=wonum=43041&oslc.select=wonum,description,

description\_longdescription,status,owner,rel.failure\{failurecode,description\},

rel.location\{location,description\},reportdate,actstart,targstartdate,schedstart,

rel.classstructure\{classificationid\},rel.assignment\{laborcode\},rel.asset\

{assetnum,assettype,description,rel.classstructure\{classificationid\}\},

rel.worklog\{\*\}" -o test.json

To download a set of work orders, you can add the **oslc.where=wonum in [,,...]** parameter. For example, using the following command, you can download work orders 1158, 1161, and T1056.

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:

*maximo\_server\_port*/*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?

\_lid=wilson&\_lpwd=wilson&lean=1&ignorecollectionref=1&oslc.where=wonum%20in%20\

[1158,1161,\"T1056\"\]&oslc.select=wonum,description,description\_longdescription,

status,owner,rel.failure\{failurecode,description\},rel.location\{location,

description\},reportdate,actstart,targstartdate,schedstart,rel.classstructure\

{classificationid\},rel.assignment\{laborcode\},rel.asset\{assetnum,assettype,

description,rel.classstructure\{classificationid\}\},rel.worklog\{\*\}" -o test.json

To sort the work orders by attributes, you can add query parameter **oslc.orderBy=+attr1,-attr2**. Attributes prefixed with the plus sign (+) are sorted in ascending order, and the attributes prefixed with the minus sign (-) are sorted in descending order. Related attributes are not supported to be sorted. For example, using the following command, you can download 100 work orders that are reported most recently in the past 3 months.

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?\_lid=wilson&\_lpwd=wilson&lean=1&

ignorecollectionref=1&pageno=1&oslc.pageSize=100&oslc.orderBy=-reportdate&tlrange=

-3M&tlattribute=reportdate&oslc.select=wonum,description,description\_longdescription,

status,owner,rel.failure\{failurecode,description\},rel.location\{location,

description\},reportdate,actstart,targstartdate,schedstart,rel.classstructure\

{classificationid\},rel.assignment\{laborcode\},rel.asset\{assetnum,assettype,

description,rel.classstructure\{classificationid\}\},rel.worklog\{\*\}" -o test.json

Using the following command, you can download the first 1000 work orders after sorting the work orders in ascending order. Note that you must use **%2B** instead of **+** in the curl request.

curl --header "x-public-uri:http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc" "http://*maximo\_server*:*maximo\_server\_port*/

*maximo\_context\_root*/oslc/os/MXAPIWODETAIL?\_lid=wilson&\_lpwd=wilson&lean=

1&ignorecollectionref=1&pageno=1&oslc.pageSize=1000&oslc.orderBy=%2Bwonum&oslc.

select=wonum,description,description\_longdescription,status,owner,rel.failure\

{failurecode,description\},rel.location\{location,description\},reportdate,actstart,

targstartdate,schedstart,rel.classstructure\{classificationid\},rel.assignment\

{laborcode\},rel.asset\{assetnum,assettype,description,rel.classstructure\

{classificationid\}\},rel.worklog\{\*\}" -o test.json

Uploading historical Maximo work orders After you download Maximo work orders to a JSON file, you can upload the work orders into a Maximo work order collection. You can upload a maximum of 10,000 work orders at one time.

1. In IBM Maximo Assist Studio, launch the Document Manager.
2. Select the Maximo work order collection.
3. Select **Add New** and choose the JSON file that contains the downloaded work orders.
4. Input the information and then select **Upload**.
5. You can add new work orders or update the existing work orders by selecting **Add New**.

Opening IBM Maximo Assist from IBM Maximo Technician

After integration is enabled, you can open IBM Maximo Assist from IBM Maximo Technician. There are two ways to open Assist from Technician:

1. Open work order detail page of one maximo work order, at the bottom of the work order detail page, click on the Assist icon.

2. Open work order detail page of one maximo work order, at the bottom of the work order detail page, click the task icon. In the task page, expand one task, click on the Launch assist button icon.

# Automatically refreshing Maximo work orders

Use Maximo Integration Framework to configure IBM Maximo Assist so that work orders automatically refresh when they are updated in Maximo.

### Add template to MXAPIWODetail Object Structure

1. Go to "Object Structures" application.

2. Search "MXAPIWODETAIL" Object Structure.

3. Click "Query Template" on the left menu.

4. Add "EMA" to Query Templates for MXAPIWODETAIL.

- Template Name: EMA

- Page Size: 1

5. Add template attributes to "Query Template Attributes for EMA".

- actstart

- description

- description\_longdescription

- owner

- rel\$asset.assetnum\\*

- rel\$asset.assettype\\*

- rel\$asset.description\\*

- rel$asset.rel$classstructure.classificationid\\*

- rel\$assignment.laborcode\\*

- rel\$classstructure.classificationid\\*

- rel\$failure.description\\*

- rel\$failure.failurecode\\*

- rel\$location.description\\*

- rel\$location.location\\*

- reportdate

- schedstart

- status

- targstartdate

- wonum

- siteid

- rel\$asset.siteid\\*

- rel\$location.siteid\\*

- rel\$woactivity.taskid\\*

- rel\$woactivity.description\\*

### Create Publish Channel

1. Go to "Publish Channel" application.

2. Click "New Publish Channel".

3. Fill the contents.

- Name: MXINTEMA

- Object Structure: MXAPIWODETAIL

- Template Name: EMA

- Publish JSON?: check

4. Click "Enable Event Listener" on the left menu.

5. Click "Save Publish Channel".

### Add new End Point

1. Go to "End Points" application.

2. Click "New End Point".

3. Fill the contents after changing the {} parts.

- Name: EMA

- Handler: HTTP

- HEADERS: X-TENANT-ID:{tenantId}, X-API-KEY:{API Key for document management}, Content-Type: application/json

- HTTPMETHOD: POST

- URL: https://ea-prod-cluster.us-south.containers.appdomain.cloud/api/v1/document-management/collections/{collectionId}/workorders/json

4. Click Test button on the bottom.

5. Fill in the Test Data with the below data.

{

 "member": [

 {

 "wonum": "1",

 "status": "OPEN"

 },

 {

 "wonum": "2",

 "status": "CLOSE"

 }

 ]

}

6. Click Test button.

(1) If you get response with {}, then the end point is OK.

(2) If you get response with java.security.cert.CertificateException, then you need to follow the section(SSL Configuration for IoT Platform and Cloudant Database)in the link(https://www.ibm.com/support/pages/node/1130841?lang=en) to add certificate for EMA. If you get the error: Received fatal alert: protocal\_version, then you can follow the step in the link(https://www.ibm.com/support/knowledgecenter/en/SSYMRC\_6.0.6.1/com.ibm.jazz.install.doc/topics/t\_enable\_tls1.2\_was.html)

(3) After complete step (2), restart the maximo server and then redo step (1) to check.

### Add new External System

1. Go to "External Systems" application.

2. Click "New External System".

3. Fill the contents.

- Name: EMA

- End Point: EMA

- Outbound Sequential Queue: emaivt13sqout(This queue is provided by Maximo Manage)

4. Check "Enabled?".

5. Click "Publish Channels" Tab.

- Click "Select Channel"

- Select "MXINTEMA"

- Click "Enabled?"

6. Click "Save External System" on left menu.

Notification: Message Provider need to be provided by Maximo Manage. And you can create your message queue by the following steps:

1. Click Add/Modify Queues
2. Click add button to add a new queue
3. Fill the contents and click OK
* Message Provider: ASTBDHTZ (This need to be provided by Maximo Manage)
* Queue JNDI Name: emaivt13sqout (This need to be provided by Maximo Manage)
* Inbound: disabled
* Retention Time： 24

### Add Cron Task Instance

1. Go to "Cron Task Setup" application.

2. Filter the "Kafka" Cron Task.

3. Select "Kafka".

4. Click "New Row" on Cron Task Instances.

5. Fill the contents.

- Cron Task Instance Name: emasqout

- Schedule: Set schedule or time interval. (ex. 1h,\\*,0,\\*,\\*,\\*,\\*,\\*,\\*,\\*)

- Run as User: MAXADMIN

- Check "Active"

5. Set parameters

- MESSAGEPROCESS : psdi.iface.jms.QueueToDestManagerProcessor

- QUEUENAME: emaivt13sqout

- MESSAGEHUB: ASTBDHTZ (This name will be provided by Maximo Manage)

6. Click "Save Cron Task Definition".

7. Click "Reload Request".