Maximo JSON API-Query

~~V1 - Updated: 11/25/2015~~
~~V2 - Updated: 12/08/2015~~
~~V3 - Updated: 12/17/2015~~
~~V4 - Updated: 03/02/2016~~
~~V5 - Updated: 04/26/2016~~
V6 - Updated 12/01/2016

Contents

Introduction 3

Querying Resources (HTTP GET) 3

  Resource Collection 4

  Collectioncount 4

  JSON Resource Data 5

OSLC.SELECT 7

  Dynamic Relationship 9

  oslc.pageSize 10

OSLC.ORDERBY 11

OSLC.WHERE 12

  Using Child Objects and related objects 12

  opmodeor 13

  Operators 13

  Null Value 14

Saved Queries 14

  Saved Queries in 7.6.0.3 16

OSLC.PROPERTIES 16

  Dynamic Relationship 18

User Session 19

Text Searching 20

Hierarchical Resources (descendent) 20

Stable Paging 21

Attachments 22

    Attachments in a Resource Collection 25

Relative URIs 25

Introduction

This document covers Querying Maximo data using the JSON API. You should review the Maximo JSON API Overview document prior to using this document.

Querying Resources (HTTP GET)

In the APIMETA for MXASSET, the queryCapability listed multiple URLs that support querying. The URLs are defined as Saved Queries for the related application (Asset Application) as well as the primary query URL for the Asset Resource, with the name All.

```
   "queryCapability":
[
         {
            "ispublic": true,
            "name": "All",
            "href": "http://host:port/maximo/oslc/os/mxasset"
         },
         {
            "ispublic": true,
            "name": "publicAssets",
            "javaMethod": true,
            "href": "http://host:port/maximo/oslc/os/mxasset?savedQuery=publicAssets"
         },
         {
            "title": "IT Stock in Stock Locations (non-Storeroom)",
            "ispublic": true,
            "name": "ITSTOCK",
            "href": "http://host:port/maximo/oslc/os/mxasset?savedQuery=ITSTOCK"
         },
            {
               "title": "X",
               "ispublic": true,
               "name": "LINKED-ASSETS",
               "href": "http://host:port/maximo/oslc/os/mxasset?savedQuery=LINKED-
         ASSETS"
            }
],
```

We will start by using the primary query URL for retrieve the asset resource data.

Resource Collection

Using this URL, *http://host:port*/maximo/oslc/os/mxasset , will retrieve a collection of links for individual assets. Each URL is a link to a Maximo Asset record.

```
{
   "member":
[
{
   "href": "http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--"
},
{
   "href": "http://host:port/maximo/oslc/os/mxasset/_MjYwMDAvQkVERk9SRA--"
},
{
   "href": "http://host:port/maximo/oslc/os/mxasset/_QTc4MDEvQkVERk9SRA--"
```

The unique Resource ID of the asset is appended to the end of the URL, thus that URL is not for a collection of assets but a single instance of the asset resource. Using that URL will return a complete list of the fields as configured in the object structure (mxasset).

Collectioncount

You can add a query parameter called collectioncount with a value of 1 (true) to the URL to have a count of resources retrieved. Using this URL, *http://host:port*/maximo/oslc/os/mxasset?collectioncount=1 ,it will include the count in the responseinfo as show below:

```
"responseInfo":


   {
      "totalCount": ?683,
```

There is a related System Property, mxe.oslc.collectioncount, when set to 1 (true) it will always return the collection count (no need to provide the query parameter). When the system property is set to 1 (true), then the query parameter can be passed with a value of 0 (false) to override the property and not return the collection count. When the system property is set to 0 (false), then the query parameter can be passed with a value of 1 (true) to override the property and return the collection count.

JSON Resource Data

Below is a snippet of an asset JSON data. A few key points:

- Only fields with a value (not null) are included in the response
- Fields included are configured as 'included' (not excluded) within the object structure definition
- Child objects, when data exists, will include the data along with a link to the record. (see assetspec near the bottom of the snippet)
- Boolean fields are returned with a value of true or false - these are known JSON types and are not wrapped in double quotes (")

http://host:port/maximo/oslc/os/mxasset/_MjYwMDAvQkVERk9SRA--

```
{
   "assetnum": "A8002",
   "changedate": "2004-10-06T22:45:22-04:00",
   "location": "HWSTOCK",
   "autowogen": false,
   "assetusercust_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--
/assetusercust",
   "_rowstamp": "26659",
   "orgid": "EAGLENA",
   "description": "Standard Desktop Computer",
   "serialnum": "GF43445T676",
   "itemnum": "D700",
   "siteid": "BEDFORD",
   "href": "http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--",
   "islinear": false,
    "isrunning": true,

    "status": "NOT READY",

    "statusdate": "2004-10-06T22:43:34-04:00",

    "hierarchypath": "ITAM_ALT \\ COMP_EQU \\ COMPUTER \\ DESKTOP",

    "newsite": "BEDFORD",

    "budgetcost": 0.0,

    "children": false,

    "assetspec_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--
/assetspecclass",

    "pluscisinhousecal": false,

    "assettype": "IT",

    "itemtype": "ITEM",

    "changeby": "WILSON",

    "vendor": "COMPDEP",

    "assetmntskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--
/assetmntskd",

    "moved": false,

    "assetopskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--
/assetopskd",

    "assetmeter_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--
/int_assetmeter",

    "assetspec":

[

        {

            "changedate": "2004-10-06T22:44:12-04:00",

            "continuous": false,

            "measureunitid": "GHZ",

            "classstructureid": "1035",

            "changeby": "WILSON",
                    "inheritedfromitem": true,

                    "mandatory": false,

                    "localref":
                  "http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--
                  /assetspecclass/0-557",

                    "numvalue": 2.4,

                    "itemspecvalchanged": false,

                    "displaysequence": 1,

                    "_rowstamp": "29225",

                    "assetattrid": "PROSPEED",

                    "orgid": "EAGLENA",

                    "href":
                  "http://host:port/maximo/oslc/os/mxasset/_QTgwMDIvQkVERk9SRA--
                  #QVNTRVQvQVNTRVRTUEVDL1BST1NQRUVEL0E4MDAyLzAvfk5VT
                  Ex_L0JFREZPUkQ-",

                    "assetspecid": 557,

                    "linearassetspecid": 0

                  },

          ],

            "disabled": false

          }
```

When querying a collection of assets you may want to see more than just a URI to determine which asset you want to retrieve. You can use the "oslc.select" query parameter to include additional attributes (columns) of the resource along with the URL.

OSLC.SELECT

The URL below will include the asset number, site id, status and description returned along with the URLs for each asset in the collection.

http://*host:port*/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,siteid,status,description

Below is one of the assets returned using the URL above. When a request is made to include additional fields of the resource, processing will also return URLs for each of the child objects in the resource. The following objects are a child to Asset in MXASSET: ASSETUSERCUST, ASSETOPSKD, ASSETMETER, ASSETMNTSKD, ASSETSPEC

```
{
   "assetusercust_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetusercust",

   "assetnum": "7506",

   "_rowstamp": "26655",

   "assetopskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetopskd",

   "assetmeter_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/int_assetmeter",

   "status": "NOT READY",

   "description": "Standard Laptop Computer",

   "assetmntskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetmntskd",

   "siteid": "BEDFORD",

   "assetspec_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetspecclass",

   "href": "http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE"

},
```

NOTE: you can specify * in the oslc.select to retrieve all columns, keeping in mind that the volume of data returned may be very large.

The following URL shows how you can select a field from a child object, in this case the select is getting the Asset number from ASSET and Metername from ASSETMETER

http://*host:port*/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum, assetmeter{metername}

You can see the Metername field displayed in the JSON response below:

```
{
   "assetmeter":
[
       {
           "_rowstamp": "678588",
           "localref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/int_assetmeter/0-27",
           "metername": "TEMP-F",
           "href":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE#QVNTRVQvQVNTRV
RNRVRFUi83NTA2LzAvVEVNUC1GL0JFREZPUkQ-"
       }
   ],
   "assetusercust_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetusercust",
   "assetnum": "7506",
   "_rowstamp": "26655",
   "assetopskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetopskd",
   "assetmeter_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/int_assetmeter",
   "assetmntskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetmntskd",
   "assetspec_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE/assetspecclass",
   "href": "http://host:port/maximo/oslc/os/mxasset/_NzUwNi9CRURGT1JE"
},
```

Within the brackets for assetmeter can be multiple columns separated by a comma or you can specify an * that would provide all the columns of the object.

Dynamic Relationship

You can also retrieve data from other objects dynamically by providing the relationship name. In the prior example, assetmeter is configured as a child object to asset in the object structure. If you wanted to see work orders for that asset, you can make that request dynamically since the workorder object is not configured in the object structure. The URL would look like this where the relationship from Asset to Workorder is called ALLWO.

http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,assetmeter{metername},rel.allwo{*}

oslc.pageSize

When you are querying a collection of resources (assets), the volume of data may be large, such that you don't want to pull all the data in a single request. One way to control the volume of data returned is to use the pageSize query parameter

http://*host:port*/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum&oslc.pageSize=3

The above URL will select assets in Maximo using a page size of 3, meaning only 3 assets will be returned at a time. At the end of the response will be a responseInfo section where there is a link to the next page (next 3 assets). When you progress to the second page (or beyond) there will also be a previous page link provided as well.

```
"responseInfo":
{
   "nextPage":
      {
         "href": "http://host:port/maximo/oslc/os/mxasset?pageno=2&oslc.pageSize=3&lean=1&oslc.select=assetnum%2Cassetmeter%7Bmetername%7D%2Crel.allwo%7B*%7D"
      },
      "href": "http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,assetmeter{metername},rel.allwo{*}&oslc.pageSize=3"
   },
   "href": "http://host:port/maximo/oslc/os/mxasset"


}
```

OSLC.ORDERBY

When querying a collection of resources, the oslc.orderBy provides the ability to sort the result set. This URL will sort the list of asset by Location in descending order.

http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,location&oslc.orderBy=-location

<u>NOTE: orderBy is CASE SENSITIVE! The B must be 'B' not 'b'</u>

-location is order by location descending

+location is order by location ascending

NOTE: if you are testing in a browser, we have seen Websphere return an error because it expects the URL to be fully encoded (the + is problematic). You may see an error like this:

```
{
   "Error":
{
   "message": "BMXAA8744E - The OSLC query was not parsed. Ensure that the query in the HTTP request follows the correct syntax of the OSLC query specification.\n\tEncountered \" <IDENTIFIER> \"location \"\" at line 1, column 2.\nWas expecting:\n    <SORT_ORDER_SIGN> ...\n    ",
   "statusCode": "400",
   "reasonCode": "BMXAA8744E",
   "extendedError":
{
   "moreInfo":
         {
            "href": "http://host:port/maximo/oslc/error/messages/BMXAA8744E"
         }
      }
   }
}
```

To work around this problem, go to an online site that supports encoding, such as **http://meyerweb.com/eric/tools/dencoder/** . Copy the value, +location, from your URL and encode it. Bring the encoded value back to your URL and execute.

OSLC.WHERE

When querying a collection of resources, the oslc.where provides the ability to filter the response of the query to a smaller set.

The URL below will return assets that reside in location MTP100

http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,location&oslc.where=location="MTP100"

Note that you can use multiple oslc query parameters (select & where) in a single request.

When a request has multiple conditions in the where clause, they are treated as 'and', meaning both conditions must be true to satisfy the where clause. This url selects assets in location MTP100 that have a status of OPERATING.

http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,location&oslc.where=location="MTP100" and status="OPERATING"

Using Child Objects and related objects

If you want the query to be based on data in a child object you can do so using this format (assetmeter is the child object of asset in mxasset)

http://*host:port*/maximo/oslc/os/mxasset?lean=1&oslc.where=assetmeter{metername="TEMP-F"}

The above URL will return all assets that have a meter named TEMP-F.

<u>Dynamic Relationship</u>

As with the oslc.select, you can use a related object dynamically, meaning the child object is specified using a relationship when that object is <u>not</u> configured in the object structure.

http://*host:port*/maximo/oslc/os/mxasset?lean=1&oslc.where=allwo.worktype="EM" }The URL uses the ALLWO relationship to query for assets tied to Emergency WOs.

opmodeor

If your conditions in the where clause need to be evaluated with an OR, meaning only one or the other needs to be true, you can leave the where clause as is, and provide another query parameter, opmodeor=1 (with a value of 1) and that will direct the request to retrieve data when either of the conditions is true (treating the 'and' as an 'or')

http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,location&oslc.where=location="MTP100" and status="OPERATING"&opmodeor=1

So all your queries are formatted with 'and' but you use the opmodeor parameter for those requests where the 'and' should be evaluated as an 'or'.

NOTE: there is no support currently for a mix of 'and' and 'or' in a oslc.where clause.

<u>Operators</u>

Other operators in addition to = :

!=	not equal
<	less than
>	greater than
<=	less than or equal
>=	greater than or equal
IN	value in a list
"like"	value like

Support for 'like' is done using the = operator and providing a partial value with the %. For example all locations that begin with 'ABC' would be done this way:

http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.select=assetnum,location&oslc.where=location="ABC%"

"%ABC" for locations ending with ABC

"%ABC%" for locations containing ABC

NOTE: if you are testing in a browser, we have seen Websphere return an error because it expects the URL to be fully encoded (the % is problematic). You may see an error like this:

```
{
   "Error":
   {
      "message": "BMXAA1649E - A programming error resulted in a null pointer exception.",
      "statusCode": "500",
      "reasonCode": null
   }
}
```

To work around this problem, go to an online site that supports encoding, such as **http://meyerweb.com/eric/tools/dencoder/** . Copy the value, "ABC%", from your URL and encode it. Bring the encoded value back to your URL and execute.

Support for IN will let you compare a value to a list and meet the condition when the value exists in the list. This URL will retrieve an asset if the asset number is 1001 or 1002.

```
http://host:port/maximo/oslc/os/mxasset?lean=1&oslc.where=assetnum in["1001","1002"]
```

Null Value

A where clause can check a field for a value of null or if the field is not null. This URL will check if the field is a null value:

```
description!=”*”
```

A check for a not null value would be

```
description="*"
```

.

Saved Queries

We reviewed the apimeta document for mxasset and saw pre-defined queries in the queryCapability section as show below.

```
    "queryCapability":
 [
        {
          "ispublic": true,
          "name": "All",
          "href": "http://host:port/maximo/oslc/os/mxasset"
        },
        {
          "ispublic": true,
          "name": "publicAssets",
          "javaMethod": true,
          "href":
        "http://host:port/maximo/oslc/os/mxasset?savedQuery=publicAssets"
        },
        {
          "title": "IT Stock in Stock Locations (non-Storeroom)",
          "ispublic": true,
          "name": "ITSTOCK",
          "href":
        "http://host:port/maximo/oslc/os/mxasset?savedQuery=ITSTOCK"
        },
          {
            "title": "X",
            "ispublic": true,
            "name": "LINKED-ASSETS",
            "href":
        "http://host:port/maximo/oslc/os/mxasset?savedQuery=LINKED-
        ASSETS"
```

The one named All is the default query that retrieves all records for a resource. Every resource (Object Structure) has this query and using this does not require the savedQuery parameter on the URL.

Maximo applications have the capability to define a query for common requests user might make, such as all assets in a specific location or all assets of a certain type or status. When these types of saved queries are defined for the application they can also be made available to users of the API. To support this:

the Object Structure must be configured with the application name as the Authorized Application

the saved query must be defined as Public or the user of the api request must be the 'owner' of the query for it to be available.

Using the links provided in the apimeta allows you to execute saved queries such as the one below

http://host:port/maximo/oslc/os/mxasset?savedQuery=ITSTOCK

Note: the parameter savedQuery is case sensitive, if you provide 'savedquery' this is not a valid parameter and will be ignored (no error).

Saved Queries in 7.6.0.3

In Maximo version 7.6.0.3 additional capability was provided to support additional sources of saved queries such as using an automation script. See the related tech note for more information:

http://www.ibm.com/support/docview.wss?uid=swg21972876

OSLC.PROPERTIES

The oslc.properties query parameter is different than those above in that it is used for single resource rather than a collection. The URL would need to include a resource id such as this example below:

http://host:port/maximo/oslc/os/mxasset/_MjYwMDAvQkVERk9SRA--

This URL would return the data for a single asset. The oslc.properties is comparable to the oslc.select except that it is used on a single resource. This allows you to control/limit the content returned when requesting data for an asset. The URL above would return all columns defined by the MXASSET object structure (Asset and its related objects).

This URL

http://host:port/maximo/oslc/os/mxasset/_MjYwMDAvQkVERk9SRA--
oslc.properties=assetnum, status, location

would return this data:

```
{

   "assetusercust_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetusercust",

   "assetnum": "26200",
    "_rowstamp": "26651",

    "assetopskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetopskd",

    "assetmeter_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/int_assetmeter",

    "status": "NOT READY",

    "assetmntskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetmntskd",

    "assetspec_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetspecclass",

    "href": "http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--"

}
```

Only the 3 fields identified in the query parameter are returned. As with the oslc.select, when you request for specific fields, you will also receive the urls for each child object (which may or may not contain data).

You can specify fields for child objects in the same way that it is done for the oslc.select parameter

http://host:port/maximo/oslc/os/mxasset/_MjYwMDAvQkVERk9SRA--
oslc.properties=assetnum, status, location, assetmeter{metername}

would return data that included the metername from the assetmeter object which is the child to asset in the object structure:

```
{

    "assetmeter":

[

      {

          "_rowstamp": "861244",

          "localref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/int_assetmeter/0-47",

          "metername": "TEMP-F",
            "href":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
#QVNTRVQvQVNTRVRNRVRFUi8yNjIwMC8wL1RFTVAtRi9CRURGT1JE"

        }

    ],

    "assetusercust_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetusercust",

    "assetnum": "26200",

    "_rowstamp": "26651",

    "assetopskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetopskd",

    "assetmeter_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/int_assetmeter",

    "status": "NOT READY",

    "assetmntskd_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetmntskd",

    "assetspec_collectionref":
"http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--
/assetspecclass",

    "href": "http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--"

}
```

Dynamic Relationship

Similar to oslc.select, you can also access data from related objects that are not configured in the object structure. This URL would return any related WOs for this asset as determined by the ALLWO relationship.

```
http://host:port/maximo/oslc/os/mxasset/_MjYwMDAvQkVERk9SRA--
oslc.properties=assetnum, status, location,
assetmeter{metername},rel.allwo{wonum,description}
```

This additional JSON data would be appended to the JSON data above when this URL is executed (in this example the asset had 1 related WO).

```
"allwo":

[

    {
            "description": "test",

            "wonum": "1203"

          }

        ],

        "href": "http://host:port/maximo/oslc/os/mxasset/_MjYyMDAvQkVERk9SRA--"

      }


      }
```

User Session

Invoking the APIs are independent, stateless calls that perform a Maximo Login for each call. This can create a significant amount of overhead for each request. A client can establish a session where they make initial login and then re-use that connection for subsequent calls.

This can be done by performing an initial login using this URL

    http://localhost:port/maximo/oslc/login

and passing in your user credentials (maxauth header when using Maximo authentication or in a test scenario you can provide parameters: _lid and _lpwd).

In the response will be the header Set-Cookie with a value similar to this:

```
JSESSIONID=0000M7euRrvVDTwtWRUIUNKmtBA:-1; Path=/;
HttpOnly
```

On subsequent requests, instead of providing the credentials (maxauth), you can provide the header, Cookie, with a value of

```
JSESSIONID=0000M7euRrvVDTwtWRUIUNKmtBA:-1
```

using the string provided in the Set-Cookie header up to the first semi colon (;). This will allow you to re-use the session rather performing another Maximo login. The JSessionid expires after 30 minutes of inactivity.

Text Searching

The API supports recordwide text search where you can search one or more fields for a value. For example, query all assets that have '12BTY24BD9' in either the description or serial number fields. This is accomplished using two query parameters. The parameter, oslc.searchTerms identifies the value that the search is for and searchAttributes identifies the resource fields to search.

> http://host:port/maximo/oslc/os/mxasset?lean=1&searchAttributes=description,serialnumber&oslc.searchTerms="12BTY24BD9"

Note that the query parameter names are <u>case sensitive</u>.

Hierarchical Resources (descendent)

An object structure (JSON Resource) can be configured as 'Self Referencing', meaning it supports the retrieval of a hierarchy of related objects, such as Work Order with all of its child Work Orders. The API provides an option to select a specified number of depths in the hierarchy or you can retrieve the full hierarchical structure (unlimited levels deep).

To retrieve a Workorder (using the MXWOHIER object structure) and return the root Work Order number and the work order number of each child Work Order immediately under the root WO, the URL would look like this:

> http://localhost:port/maximo/oslc/os/mxwohier/_QkVERk9SRC8xMDAw?oslc.properties=wonum,workorder{wonum}

When the object structure is hierarchical, specifying the object name as part of the oslc.properties infers selecting the WO in the next level down the hierarchy.

To query to the next level (2 levels deep) the URL would be

> http://localhost:port/maximo/oslc/os/mxwohier/_QkVERk9SRC8xMDAw?oslc.properties=wonum,workorder{wonum, workorder{wonum}

If you wanted all the child WOs for all levels below the root your URL would use 'descendent' in the URL:

> http://localhost:port/maximo/oslc/os/mxwohier/_QkVERk9SRC8xMDAw?oslc.properties=wonum,workorder{wonum, descendent}

Above would give the root WO, all of its child work orders and all their children and so on.

The above examples are queries for a single WO resource using oslc.properties. This same capability can be used when querying a collection using the oslc.select parameter.

Stable Paging

OSLC stable paging defines a paging pattern where the resource that is paged does not change when it is being paged to the client (a mobile device for example). Using stable paging provides improved performance by paging through in-memory data rather than making requests to the database. Stable paging is supported only for a resource collection and has some restrictions. A common use of stable paging would be the need to download master data to a client.

Stable paging is not the default paging format of the servers. To request stable paging, the client must provide the stablepaging query parameter as part of the collection request URI. The server loads the requested collection resource in memory and redirects the OSLC client to a new URI that points to the loaded collection resource. Subsequent requests for the next pages are served from this loaded resource.

Example of stable paging

The client requests an asset resource:
http://localhost:port/maximo/oslc/os/mxasset?stablepaging=true

The request produces the following response:
303 Redirecting
Location: http://host:port/../mxasset?stableid=1234

The client then performs a GET request on the redirect URI:
http://localhost:port/maximo/oslc/os /mxasset?stableid=1234

The response is the first page of the collection. The URI for the next page is embedded in the oslc:ResponseInfo object as part of the message. With stable paging, the subsequent requests always load from the in-memory resource and never from the database. Stable paging results in better performance as the client pages through the result.

As the client moves through the pages, they expire after they load. So if the client is at page 3, page 3 cannot be reloaded, nor can page 1 or page 2. Any attempt to reload those pages generates an HTTP 410 error. The loaded resource stays in memory until it expires based on an idle expiry time that is controlled by using the mxe.oslc.idleexpiry system property.

The idle expiry time indicates the period that the resource was not accessed by the client. The client can request member properties by using the oslc.select clause and order the resulting collection by using the oslc.orderBy query parameter.

In a clustered configuration, stable paging requires that the subsequent requests for next page is served by the same server process or cluster member that served the initial request for stable paging. The same server process is required because the resource is loaded in the memory of that server process, not in the other cluster members. np

Attachments

The API supports the retrieval of attachments that are associated to resources. For example if an asset is configured with a PDF attachment that describes the maintenance procedures for that asset, a client can query that asset and request the attachment be returned with the resource data.

Note: see the related API CRUD document regarding the API's support for creating/updating/deleting attachments associated to a resource.

For a resource, such as MXASSET, to support attachments:

- the Maximo attachments feature has to enabled
- the mxasset object structure must be configured with the DOCLINKS MBO as a child to the ASSET object.

You can access attachments when querying a resource collection or an individual resource. An attachment is made up of two components, the attachment file and the related metadata of the attachment.

When you query a specific resource (using its ID) that has an attachment, you will get a doclinks URL returned for the attachment:

Request URL:

```
http://localhost:port/maximo/oslc/os/mxasset/_MTAwMS9CRURGT1JE
```

Response

```
{


   "assetnum": "1001",

   "changedate": "1999-03-31T16:53:00-05:00",

   "location": "MOFLOOR1",

   "tloampartition": false,

   "totdowntime": 0.0,

   "autowogen": false,

   "assetusercust_collectionref":
"http://localhost:port/maximo/oslc/os/mxasset/_MTAwMS9CRURGT1JE/assetusercust",

   "_rowstamp": "26592",

   "orgid": "EAGLENA",

   "description": "Fire Extinguisher",

   "priority": 3,

   "itemnum": "FIRE-100",

   "doclinks":

{

   "href":
"http://localhost:port/maximo/oslc/os/mxasset/_MTAwMS9CRURGT1JE/doclinks"

},
```

Using the **doclinks** url from the JSON data above, you will be returned the metadata (with its URL) and a URL to retrieve the attachment file.

```
{

   "href":
"http://localhost:port/maximo/oslc/os/mxasset/_MTAwMS9CRURGT1JE/doclinks"
,

   "member":
```

```
[

{

   "href":
"http://localhost:port/maximo/oslc/os/mxasset/_MTAwMS9CRURGT1JE/docli
nks/76",

   "describedBy":

{

   "docinfoid": 58,

   "addinfo": false,

   "weburl": "http://localhost:port/ATTACHMENTS/Presentation1.ppt",

   "docType": "Attachments",

   "changeby": "WILSON",

   "createby": "WILSON",

   "copylinktowo": false,

   "show": false,

   "format":

            {

               "label": "application/vnd.ms-powerpoint",

               "href": "http://purl.org/NET/mediatypes/application/vnd.ms-
powerpoint"

            },

            "getlatestversion": true,

            "ownerid": 53,

            "printthrulink": false,

            "urlType": "FILE",

            "upload": false,

            "attachmentSize": 101376,

            "modified": "2015-12-04T09:54:24-05:00",

            "title": "ATTACH1",

            "created": "2015-12-04T09:53:43-05:00",

            "description": "Test Attachment 1",
                "fileName": "Presentation1.ppt",

                "ownertable": "ASSET",

                "href":
"http://localhost:port/maximo/oslc/os/mxasset/_MTAwMS9CRURGT1JE/doclinks/
meta/76",

                "identifier": "76"

              }

            }

          ]
```

The .../**doclinks/ID** url is the link to the actual attachment file.

Attachments in a Resource Collection

To retrieve the same attachment data (as shown above) while querying a resource collection (not a specific resource) you can request it using the oslc.select:

http://localhost:port/maximo/oslc/os/mxasset?oslc.select=assetnum,**doclinks{*}**

By specifying doclinks{*} in the oslc.select, this will return attachment data for each resource in the collection

Relative URIs

You can hide the hostname in the links provided by the API when querying (and updating) by providing the query parameter, relativeuri=1.

This URL

http://localhost:port/maximo/oslc/os/mxasset?relativeuri=1

will return a collection URLs with this format:

```
{

  "member": [

    {

      "href": "oslc/os/mxasset/_QkVERk9SRC82NzI3"

    },
    {
      "href": "oslc/os/mxasset/_QkVERk9SRC83NDky"
    },
    {
      "href": "oslc/os/mxassset/_QkVERk9SRC81MDEw"
    },
```

The client that uses these URLs must prefix them with 'http://hostname/maximo'.

Please send any corrections or suggestions to Tom Sarasin at tsarasin@us.ibm.com