IBM Maximo Asset Performance Management for Energy & Utilities SaaS

User Guide

IBM

Note

Before using this information and the product it supports, read the information in Chapter 11, “Notices,” on page 295.

This edition applies to IoT for Energy and Utilities on Cloud.

Executive summary

This document contains the IBM® Maximo Asset Performance Management for Energy & Utilities SaaS product overview application overview, installation instruction, administrative guide, and application and user instructions for Asset Performance Management and Asset Investment, Connectivity Model, Asset 360 for Wind and Situational Awareness.

The installation instructions describe both a Docker install and the upgrade instructions for a non Docker installation.

The administering of the product describes the global settings for the product, the instructions for services, the system status monitoring instructions, and managing the standard operating procedures.

The Asset Performance Management application describes the custom data model that is used by the application, the customizing of the application, and the viewing and running of an analysis.

The Asset Investment application section describes the creating investment projects and viewing the results with comparisons.

The Using Connectivity Model describes the data flow and configuration of the application, the automation of the data flow and the user instructions for Connectivity Model.

The Optimizing Wind Farm operations describes the overview of the application, the custom data model information, the administration of the application and the monitoring of wind farm operations.

Contents

Chapter 1. Welcome

Welcome to the IBM Maximo Asset Performance Management for Energy & Utilities SaaS documentation, where you can find information about how to maintain and use Maximo APM for E&U SaaS.

Getting started

"Product overview" on page 3

IBM Maximo Asset Performance Management for Energy & Utilities SaaS is an analytics solution for the energy and utilities industry. You can use the solution to develop new applications to support analytics use cases for your assets and networks, and to integrate existing applications with the solution.

"What's new" on page 4

The following new features and fixes are available in IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

"Configuring the product" on page 11

The applications in IBM Maximo Asset Performance Management for Energy & Utilities SaaS can be administered with users with the correct administrative rights.

Product legal notices

Common tasks

"Using Asset Performance Management application" on page 129

The Asset Performance Management application shows how well a specific asset will provide its service in the future.

Using the Connectivity Model application

The Connectivity Model gives feedback to managers, data analysts and grid operators as to the accuracy of the phase and connectivity details of the network.

"Overview of the Asset 360 for Wind application" on page 211

The IBM Maximo Asset Performance Management for Energy & Utilities SaaS application Asset 360 for Wind provides an advisor for wind farm operations and maintenance.

"Reviewing vegetation data" on page 299

You use the Vegetation Management application to review the status of vegetation and the actions that you can take to control the level of vegetation in the corridors of your region. You can use filters to refine the views to show the information that is displayed on the **Dashboard**, **Map, and List** views.

Troubleshooting and support

IBM Software Support home page

More information

IBM Knowledge Center content in PDF format

Chapter 2. Product overview

IBM Maximo Asset Performance Management for Energy & Utilities SaaS is an analytics solution for the energy and utilities industry. You can use the solution to develop new applications to support analytics use cases for your assets and networks, and to integrate existing applications with the solution.

By using data integration, analytics, and visualization, Maximo APM for E&U SaaS provides a detailed and accurate understanding of asset and network performance. Maximo APM for E&U SaaS integrates with existing data sources and operational processes to analyze and predict asset performance and risk to help deliver safe, reliable, affordable, and sustainable energy.

You can use the open extension capabilities that are provided by the user interface framework to build application user interfaces that meet your operational requirements.

Maximo APM for E&U SaaS includes applications that provide a current view of interconnected assets that ensures that asset replacement, maintenance, and performance are optimized.

Understand asset performance

The Asset Performance Management application provides a historical and current assessment of critical assets.

By analyzing the relevant data sources and applying the built-in asset models, the solution provides a current assessment of asset performance from which you can use to make well-informed decisions about asset operations and investments.

View the current status of assets and networks

The Situational Awareness application helps you to monitor the changes to the status of assets so that action can be taken to solve issues or plan and optimize the settings for the assets.

You can understand the performance of an individual asset and network, assess conditions, and take corrective actions.

View the current status of wind farm operations

The Asset 360 for Wind application offers role-based access to historical, current, and predictive insights for wind turbines and wind farms. The Wind 360 application provides situational awareness for assets that use historical and current data that is obtained from the relevant operational systems. Integration with data from The Weather Company further optimizes wind farm performance and assesses turbine health and risk. Reports provide an overall and detailed analyses of the wind farm performance, operation, and maintenance based on KPIs.

Receive accurate reports of phase and transformer mapping

The Connectivity Model application identifies the meters that are connected to each of the electrical phases and provides recommendations to help fix connectivity records without having to send crews into the field.

The automated verification facilitates accurate analyses after system interruption, provides the detail and extent of an outage, and improves fault location, isolation, and service restoration.

Understand the status of vegetation for you region

The Vegetation Management application offers analyses of vegetation data from IBM PAIRS Geoscope. The solution identifies and alerts the vegetation manager about potential outage threats in their territory. The solution also highlights any vegetation that breaches your predetermined buffer zones.

Develop and extend your applications

The product provides REST API services that can be used to extend the product and provide integration interfaces to other systems.

- Unify systems and business processes by integrating multiple data sources such as sensors, SCADA, weather, Enterprise Asset Management (EAM).
- Deliver contextual awareness by correlating, analyzing, and visualizing data within and across systems and processes.

The UI framework can be used to develop and extend your own applications.

Analyze your data

IBM Open Platform with Apache Spark and Apache Hadoop is an open source solution for analyzing and visualizing Internet-scale data volumes that is powered by Apache Hadoop.

Manage your assets

IBM Watson IoT Platform provides a user interface where you can and add and manage your devices, control access to your IoT service, and monitor your usage.

You can integrate Maximo® APM for E&U with IBM Watson® IoT Platform to collect asset data from devices that are connected to Watson™ IoT Platform.

What's new

The following new features and fixes are available in IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

What's new June 2020

The Asset Performance Management application is updated with the following enhancements:

- Updated middleware versions including WebSphere® Application Server Liberty, IBM Db2®, SPSS Modeler, Cognos Analytics, and so on.
- Optimized asset health data management to generate low data volume for Cognos reports.
- As a solution administrator, you can create filters that are only visible for a specific groups of users
- As an reliability engineer, you can prepare a list of assets identified by their OIDs and type and configure the filter with this list.

The Vegetation Management application is updated with the following changes or enhancements:

- Solved the synchronization issues between filter menu, corridor filter, layer control and map.
- Enhanced corridor preview card to show more important information.
- Enhanced corridor segment view to enable default selection and add more information to improve visibility.
- Enhanced vegetation polygon preview card to add more information.
- Ability to filter within organizational levels beyond containers and asset categories.
- Provided a clear view of the corridor segments that are on or off cycles, as well as an indication of how far they are from the cycle.
- Support to remove the background map to clearly view the layers above.
- Filter areas based on the organizational structure that has been implemented into the tool (service territories organization).
- The sample Notebooks are reorganized and simplified.
- The corridor segments can be generated from power lines using different options.

What's new January 2020

The Asset Performance Management application is updated with the following enhancements:

- As an administrator, you can define the asset class hierarchy for the filter in the APM UI configuration.
- As an administrator, you can set your filter preset as public to share with all users.
- As a user, you can use advanced filtering to filter out assets based on their attributes or factors or the attributes or factors of their subcomponents.
- As a user, you can view the health and risk of the subcomponent of an asset's single asset report.

The Vegetation Management application is updated with the following changes or enhancements:

- The term "projects" was changed to "territories" in the user interface.
- Two KPIs were added in the out-of-the-box analysis: Vegetation Dispersion and Distance to Conductor.
- The corridor list is added in the List view. You can search for a corridor by corridor name.
- For all lists, you can adjust the column order by changing the configuration. You can jump to the map from the item in the list.

What's new November 2019

The Asset Performance Management application is updated with the following enhancements:

- As a developer, you can use the configuration file to set a virtual group of asset classes in the filter.
- As an asset manager, you can use an imported list of asset IDs in the advanced filter to narrow down the filtered results.
- As an asset manager, you can select a candidate value from a drop-down list when you type a string in the advanced filter.
- As an asset manager, you can use a scatter chart in the asset report to view the health status of an asset in relation to the asset class.
- As an asset manager, you can see the typical useful life and the maximum useful life information in the analysis results of a single asset report.
- As an asset manager, for substation transformers that you do not have edit privileges for, the DGA option chart is not shown.

What's new August 2019

The new application, Vegetation Management introduces the following features:

- As a vegetation manager, you can use the Vegetation Management application to assess vegetation information in different ways for you to understand the vegetation risk and need for action. You can create output that can be used to determine work priority and resource planning.
- As a data loading developer, you can customize the sample code to import asset and vegetation data from different data sources. You can integrate IBM Maximo Asset Performance Management for Energy & Utilities SaaS with IBM PAIRS Geoscope.
- As a data scientist, you can customize the analysis model to create actions and key performance indicators based on the needs of your utility.

The Asset Performance Management application is updated for the following features and system enhancements:

- As the system administrator, you can archive and delete historic and obsolete analysis data.
- As the user, you can export data according to your role and privilege.
- Filter text is enhanced.
- The system performance is enhanced for when many regions are part of an analysis.

What's new January 2019

The following features were included in the Asset Performance Management application:

- An administrator can assign different privileges to users and groups of users. The users see different sets of data in the application, and the user role can receive different levels of operation.
- An administrator can show or hide list columns in the user interface without changing code.
- A reliability professional can see several dissolved gas analyzes of the model on substation transformers and select one model to be one factor in the final asset health analysis model.
- A reliability professional can see the Maximo Asset Management work order list directly from the asset details and can create new work orders for the selected asset.

SaaS security

The SaaS delivery for IBM Maximo Asset Performance Management for Energy & Utilities SaaS has these security features as part of the SaaS delivery.

- SSL and RSA encryption for internal communications
- Lightweight Directory Access Protocol (LDAP) permissions for credential storage
- Abnormal user activity detection
- Single sign-on authentication

Encrypts internal communications

IBM Maximo Asset Performance Management for Energy & Utilities SaaS uses SSL with RSA encryption for the following internal server communications:

- Between the IBM HTTP Server (IIB Node) and the WebSphere Application Server Liberty (IIB Node).
- Between the IBM HTTP Server (IIB Node) and the Cognos® Analytics Server (BI Node).
- Between the ASK node (BI Node) and the Cognos Analytics Server (BI Node).
- Between the IBM Db2 (DB Node) and the SPSS® Modeler Server (SPSS Node), the WebSphere Application Server Liberty (IIB Node), the grid engine (SPSS Node), and Cognos Analytics Server (BI Node).
- Between the LDAP (DB Node) and the WebSphere Application Server Liberty (IIB Node).
- Between the LDAP (DB Node) and the Cognos Analytics Server and ASK node (BI Node).

Maximo APM for E&U SaaS uses the encryption in IBM Db2 for the encryption of data and the files that contain customer and user data and are stored on an encrypted folder on the App Node.

All external communication is encrypted by using TLS v1.2 between the browser interface and the application, and between the browser interface and the Web Node on the Cognos Analytics portal.

Uses LDAP authentication

Maximo APM for E&U SaaS uses the LDAP protocol to access the user login credentials. The credentials can be in any arbitrary data storage system that uses an LDAP API.

Detects abnormal user activity

The login and logout of users and the user activity while active is recorded and monitored to detect abnormal activities at the following levels:

Application level
The login and logout activities to an application are recorded.

Middleware level
Activities are monitored and recorded for the middleware: WebSphere Application Server Liberty, IHS, ASK server, Cognos Analytics Server, SPSS Modeler Server, IBM Db2, and Jena.

Operating system level
Logrotate is enabled to manage the log files for the following logs: `/var/log/cron`, `/var/log/maillog`, `/var/log/spooler`, `/var/log/dmesg`, `/var/log/boot.log`, `/var/log/messages`, and `/var/log/secure`.

Logrotate

The Logrotate utility is enabled for activity logs on the middleware. The application log files in Maximo APM for E&U SaaS are included in the WebSphere Application Server Liberty server on the application node.

Authenticates with single sign-on

Maximo APM for E&U SaaS uses single sign-on in the Maximo APM for E&U SaaS application to secure the transfer of the user ID and password credentials that are used to authenticate with the system. Users can switch between applications without having to authenticate again.

Provided hardware configuration

IBM Maximo Asset Performance Management for Energy & Utilities SaaS has a default hardware configuration that changes depending on the application.

Table 1. Default hardware for Asset Performance Management and Asset 360 for Wind

Node	**CPU**	**Memory**	**Disk**
IIB node	Four cores	8 GB	250 GB
DB node	Sixteen cores	32 GB	200 GB
Ana node	Four cores	8 GB	400 GB
BI node (if used)	Four cores	8 GB	100 GB

Connectivity Model on big data architecture, the default disk space for each tenant is 1 TB.

Accessibility

IBM is committed to accessibility for tools and guidance.

For more information about the commitment, see the IBM Human Ability and Accessibility Center. The IBM Human Ability and Accessibility Center is at the following web address: http://www.ibm.com/able/

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work
must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies” and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Chapter 3. Configuring the product

The applications in IBM Maximo Asset Performance Management for Energy & Utilities SaaS can be administered with users with the correct administrative rights.

Managing system access

Securing your IBM Maximo Asset Performance Management for Energy & Utilities SaaS solution is an important consideration. To ensure that the system is secure, you must manage who can access the system and assign the correct level of access within the solution.

Securing access to the solution

Maximo APM for E&U SaaS uses a IBM WebSphere Application Server Liberty Server basic user registry to authenticate and authorize users. For more information about Liberty profile user registries, see the related link https://www.ibm.com/support/knowledgecenter/SS7K4U/mapfiles/product_welcome_waszos.html.

Your administrator assigns access to features, data, and services in your solution based on the user role groups.

The following topics explain the security and how to manage user access to Maximo APM for E&U SaaS.

Adding users and user groups

To access specific features or services in the solution, users must be a member of a user role group that provides the access to that feature or service. As an administrator, you can add users and user role groups to the solution by adding users and groups to the LDAP server on the App node.

Before you begin

Decide on the groups, users, and user passwords that you want to add to the basic user registry in Maximo APM for E&U SaaS.

About this task

IBM Maximo Asset Performance Management for Energy & Utilities SaaS uses an open LDAP registry to define users and user role groups.

Procedure

1. Log in to the App Node, and enter the `IFELdapNode` Docker container.

```
sudo docker exec -it IFELdapNode bash
```

2. Add the new user or user group to the LDAP Server.
   a) Create an LDIF file to describe the group and user information.

   ```
   touch  /tmp/temp.ldif
   ```

   b) Add the new group and users of Maximo APM for E&U SaaS to the `temp.ldif` file and replace the user password attribute value with the password for the new user or user group.
   The user group `reliability_group` is to include the users `rel_engineer_01` and `rel_engineer_02`. The two users are added to the content to the `temp.ldif` file.

   ```
   dn: cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com
   objectclass: top
   objectclass: groupOfNames
   member:cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
   member: cn=rel_engineer_02,cn=ife,ou=application,dc=ibmiot,dc=com
   cn: reliability_group
   ```

```
dn: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
objectClass: person
objectClass: top
cn: rel_engineer_01
sn: rel_engineer_01
userpassword: password_for_rel_engineer_01

dn: cn=rel_engineer_02,cn=ife,ou=application,dc=ibmiot,dc=com
objectClass: person
objectClass: top
cn: rel_engineer_02
sn: rel_engineer_02
userpassword: password_for_rel_engineer_02
```

c) Use the following command to add the users to the LDAP server and replace *${LDAP_PASSWORD}* with the actual `ldapServerPassword` that is provided during installation in the `config.properties` file.

```
ldapadd -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD} -f /tmp/temp.ldif
```

3. Check the users and user groups in the LDAP server.

a) Check the new user with the command.

```
ldapsearch -x  | grep  rel_engineer_01
```

The following report is an example output:

```
[root@IFELdapNode /]# ldapsearch -x  | grep  rel_engineer_01
member: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
# rel_engineer_01, ife, application, ibmiot.com
dn: cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com
cn: rel_engineer_01
sn: rel_engineer_01
```

b) Check the new user group with the command.

```
ldapsearch -x  | grep  reliability_group
```

The following report is an example output:

```
[root@IFELdapNode /]# ldapsearch -x  | grep  reliability_group
# reliability_group, ife, application, ibmiot.com
dn: cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com
cn: reliability_group
```

Results

The new user groups and users are added to the basic user registry, and the users can be authenticated when they log on to Maximo APM for E&U SaaS.

What to do next

- To generate usage information for the IBM License Metric Tool, you must map each user role group to the relevant license type in the `slmtag_groups.properties` file on the application server. For more information, see “Mapping user groups to license types” on page 22.
- As an administrator, you can now assign access to pages and REST services in the solution for each new user role.

Modifying users, user groups, and passwords

You can change passwords and user group membership for users of IBM Maximo Asset Performance Management for Energy & Utilities SaaS. Membership of a user role group permits users to access the features and services that are associated to appropriate to that user role group.

Before you begin

- Ensure that users who you modify are not logged on to IBM Maximo Asset Performance Management for Energy & Utilities SaaS.
- Before you remove user groups from the basic user registry, ensure that the user role groups are removed from access to pages and services in the solution.

The following steps have *rel_engineer_01*, *rel_engineer_02*, and *reliability_group* as example user and group names. The steps also use the example passwords *${LDAP_PASSWORD}* and *passw0rdChange*. Use your own text strings for users, groups, and passwords.

About this task

You can change the access level of a user by updating the basic user registry, remove the user from one user role group, and add the user to another user role group. You can also update the basic user registry to delete users and user role groups.

Procedure

1. Log in the App Node and open the `IFELdapNode` docker container with the command.

   ```
   sudo docker exec -it IFELdapNode bash
   ```

2. Run the following command to create a temporary LDIF file.

   ```
   touch  /tmp/temp.ldif
   ```

3. To add a user to the user role group:

   a) Create a `user.ldif` file in the `temp` directory with the paste the following script into the file. Replace the *password_for_user* with the user password.

   ```
   dn: cn=user,cn=ife,ou=application,dc=ibmiot,dc=com
   objectClass: person
   objectClass: top
   cn: user
   sn: user
   userpassword: password_for_user
   ```

   b) Run the following command to add the user group role, replace *LDAP_PASSWORD* with the `ldapServerPassword` that is in the `config.properties` file.

   ```
   ldapadd -x -D "cn=Manager,dc=ibmiot,dc=com" -w
   LDAP_PASSWORD -f /tmp/user2.ldif
   ```

   00

   c) Add the following script to the `/tmp/temp.ldif` file: Replace the *user* variable with the user name and the *user_role_group* variable with the user role group name.

   ```
   dn:cn=user_role_group,cn=ife,ou=application,dc=ibmiot,dc=com
   changetype: modify
   add: member
   member: cn=user,cn=ife,ou=application,dc=ibmiot,dc=com
   ```

   d) Run the following command to add the contents of the `temp.ldif` file to the LDAP Server and replace the *LDAP_PASSWORD* with actual *ldapServerPassword* that is in the `config.properties` file.

   ```
   ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w LDAP_PASSWORD -f /tmp/temp.ldif
   ```

4. Change the password for the user:
    a) Add the following script to the `/tmp/temp.ldif` file: Replace the user password variable with the new password *password_Change* and the *user* variable with the user name.

    ```
    dn:cn=user,cn=ife,ou=application,dc=ibmiot,dc=com
    changetype: modify
    replace: userpassword
    userpassword: passwordChange
    ```

    b) Run the following command to add the contents of the `temp.ldif` file into LDAP Server, replace *LDAP_PASSWORD* with the `ldapServerPassword` that is in the `config.properties` file.

    ```
    ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w LDAP_PASSWORD -f /tmp/temp.ldif
    ```

5. Remove a user from the user role group.
    a) Add the following script to the file `/tmp/temp.ldif`: Replace the *user_role_group* variable with the user role group name and the *user* variable with the user name.

    ```
    dn:cn=user_role_group,cn=ife,ou=application,dc=ibmiot,dc=com
    changetype: modify
    delete: member
    member: cn=user,cn=ife,ou=application,dc=ibmiot,dc=com
    ```

    b) Run the following command to add the contents of the `temp.ldif` file into LDAP Server: Replace *LDAP_PASSWORD* with the `ldapServerPassword` that is in the `config.properties` file.

    ```
    ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w LDAP_PASSWORD -f /tmp/temp.ldif
    ```

6. Run the following command to delete a user from the system. Replace *LDAP_PASSWORD* with the *ldapServerPassword* that is in the `config.properties` file:

    ```
    ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com" -w
    ${LDAP_PASSWORD} "cn=rel_engineer_01,cn=ife,ou=application,dc=ibmiot,dc=com"
    ```

7. Run the following command to delete a user role group from the system: Run the *ldapdelete* command and replace *LDAP_PASSWORD* with the *ldapServerPassword* that is in the `config.properties` file.

    ```
    ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com" -w
    ${LDAP_PASSWORD} "cn=reliability_group,cn=ife,ou=application,dc=ibmiot,dc=com"
    ```

Adding or removing regions to user roles and privileges in the user interface

A user or user group can have access permissions to more than one region. You can add and remove the regions that a user group or user has access to in the **Roles and Privileges** user interface of IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

You must have access rights for an administrator of Maximo APM for E&U SaaS. The roles and privileges for all users shows in a table that has three columns, **ID**, **Name**, and **Detail**. The **ID** is the ID of the user or user group, and **Name** is the user role, for example, user, administrator, and operator. The **Detail** is the number of regions and assets classes the user group or user has access to.

Note: A user has the permissions of the user group they are a member of. You can add further permissions, but you cannot remove the permissions that are assigned to the user group.

The permissions that are able to be granted are:

- View
- Edit
- Configure

Procedure

1. Click **Administration** > **Roles and Privilages**.
2. Click the **ID** of the user group that you want to add the region to and click **Edit**.
3. To add a region, do the following steps:
   a) In the **Edit permissions** screen, click **Region** > **Add**.
      The region tree shows.
   b) Make a selection of region and sub region and click **Add**.
   c) Select the level of permissions for the user or user group and click **OK**.
4. To remove a region, do the following steps:
   a) In the **Edit permissions** screen, click **Region** and then click **X** in the row of the region you want to delete.
   b) Click **OK** to confirm.

Adding or removing an asset class to a user role and privilege in the user interface

A user or user group can have access permissions to more than one asset class. You can add and remove the asset class that a user group or user has access to in the **Roles and Privileges** user interface of IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

You must have access rights for an administrator of Maximo APM for E&U SaaS. The roles and privileges for all users shows in a table that has three columns, **ID**, **Name**, and **Detail**. The **ID** is the ID of the user or user group, and **Name** is the is the user role, for example, user, administrator, and operator. The **Detail** is the number of regions and assets classes the user group or user has access to.

Note: A user has the permissions of the user group they are a member of. You can add further permissions, but you cannot remove the permissions that are assigned to the user group.

The permissions that are able to be granted are:

- View
- Edit
- Configure

Procedure

1. Click **Administration** > **Roles and Privilages**.
2. Click the **ID** of the user group that you want to add the region to and click **Edit**.
3. To add an asset class do the following:
   a) In the **Edit permissions** screen, click **Asset class** > **Add**.
      The list of asset classes shows.
   b) Make a selection of the asset classes and click **Add**.
   c) Select the level of permissions for the user or user group and click **OK**.
4. To remove an asset class do the following:
   a) In the **Edit permissions** screen click **Asset class** and then click **X** in the row of the access class you want to delete.
   b) Click **OK** to confirm.

Adding a Connectivity Model tenant user

When you have a new tenant user, follow these steps to add the new user to the Connectivity Model application.

Procedure

1. Create a user in LDAP.

   a) Log in the App Node, and open the `IFELdapNode` Docker container.

   ```
   sudo docker exec -it IFELdapNode bash
   ```

   b) Run the command with user `root`:

   ```
   /config/cm_add_tenant_user.sh <username> <userpassword> <uidNumber> <LDAP password>
   ```

   Replace the parameters *<username> <userpassword> <uidNumber> <LDAP password>* with the actual value.

   Note:

   The *uidNumber* must be a unique number, not used by other members of LDAP.

   c) Check the new user, replace the parameter *<username>* with the actual value.

   ```
   ldapsearch -x  | grep  <username>
   ```

2. Create *keytab*.

   Note: The keytab is distributed to all HDP nodes.

   a) Log in the Ambari Node, and open the `kdc` Docker container.

   ```
   sudo docker exec -it kdc bash
   ```

   b) Run the command with user `root`:

   ```
   /opt/kdc/createTenant.sh <username> <keytab_name>
   ```

   Note: The *<username>* must be the same as the *<username>* added in step 2, and *<keytab_name>* suggested being *`username`*`.keytab`

   For example: If *username=cmuser*, then *keytab_name* suggested being *cmuser.keytab*.

3. Create tenant.

   a) Log in to the Notebook node, and go to the `notebook` container.

   ```
   sudo docker exec -it notebook bash
   ```

   b) Run the command with user `root`:

   ```
   su - hdfs -c "kinit -k -t /etc/security/keytabs/hdfs.headless.keytab hdfs"
   ```

   c) Run the commands with user `root`.

   ```
   /home/cmopsadmin/cm/bin/APP_createUtility.sh
   <username> <path for keytab> <user principle>
   ```

   Currently, the *<user principle>* is the same as *<username>*, the *<path for keytab>* is `/etc/security/keytabs`. The *<username>* must be the same as for previous steps.

   For example, if *<username>* is `cmuser`, then the suggested *<keytab_name>* is `cmuser.keytab`, and the command is:

   ```
   /home/cmopsadmin/cm/bin/APP_createUtility.sh cmuser
   /etc/security/keytabs/cmuser.keytab cmuser
   ```

4. Configure a Jupyter Notebook for the tenant.
   a) Log in to the Notebook node, and go to the `notebook` container.

   ```
   sudo docker exec -it notebook bash
   ```

   b) Run the command with user `root`:

   ```
   /opt/jupyter/configureNotebook.sh <username>
   <notebook-password> <notebook-port>
   ```

   Note: The *username* must be the same as in the previous steps. The *notebook-password* is the password to log in to the Notebook: you need to create and identify the password here. The *notebook-port* is the internal port for the container that is defined in the `container.ini` file for the `notebook` container.

   For example:

   ```
   /opt/jupyter/configureNotebook.sh cmuser passw0rd 8889
   ```

   c) To verify the Jupyter Notebook, open the link with your browser:

   ```
   https://<notebook_node_hostname_or_IP>:<port>
   ```

   Where the login password is *<notebook-password>* and *<port>* is the external port that maps to the internal port or *notebook-port*. The external port, internal port, and the mapping relationship are defined the `containers.ini` file.

   Note: When you run the commands to create a Notebook for the tenant, the tenant Notebook starts automatically after a restart of the Docker container.

   If you do not want the automatic start, you can comment out the line `/opt/jupyter/startNotebook.sh <username>` in the `/etc/rc.local` file.
5. Change the default `map_center` for the tenant user.

   By default, the tenant `map_center` is. To use,`(-83.44,42.60)` do the following steps to change the default setting.
   a) Log in to the Notebook node, and go to the Notebook container.

   ```
   sudo docker exec -it notebook bash
   ```

   b) Change the current user to `cmopsadmin`.

   ```
   su - cmopsadmin
   ```

   c) Run the following commands with user `comopsadmin`.

   ```
   cd /usr/hdp/current/phoenix-client/bin/
   ```

   ```
   ./sqlline.py master01:2181:/hbase-secure:
   cmopsadmin:/etc/security/keytabs/cmopsadmin.keytab
   ```

   Note: If you get an error `Caused by: org.apache.hadoop.hbase.ipc.RemoteWithExtrasException (org.apache.hadoop.hbase.security.AccessDeniedException): org.apache.hadoop.hbase.security.AccessDeniedException: Insufficient permissions (user=cmopsadmin@IBMIOT.COM, scope=SYSTEM:CATALOG, family=0:_0, params=[table=SYSTEM:CATALOG,family=0:_0],action=WRITE)`, you can ignore it. The error is a known defect from HDP.

   d) Run the following code in Phoenix `sqlline.py` to update the default map center and related configuration:

   ```
   upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
   ```

```
values ('<utilityname>','<map_center>','<map_min_zoom>',
'<map_max_zoom>','<map_default_zoom>')
```

Note:

The *<utilityname>* is the tenant user, for example cmuser or cm_sample.

The *<map_center>* is the center of map, for example: [-83.44, 42.61].

The *<map_min_zoom>* is the minimum zoom setting for the map, for example 1.

The *<map_max_zoom>* is the maximum zoom setting for the map, for example 22.

The *<map_default_zoom>* is the default zoom setting for the map, for example 12.

For example, the sql command is like this:

```
upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
values ('cmuser','[-83.44, 42.61]','1','22','12')
```

Removing a Connectivity Model tenant user

When you remove a new tenant user, follow these steps to remove the user from the Connectivity Model application.

About this task

Important: When you remove a Connectivity Model tenant user from the application, the operation cannot be undone. Make sure that you are certain when you remove a tenant user.

The following steps use **`cm_sample/CM_SAMPLE`** as the example, you need to replace it with the actual tenant user name. The tenant name is case-sensitive. You need to replace **`cm_sample`** with the same letter case as when you create the tenant.

Procedure

1. Clear hdfs files.
   a) Log in to Notebook node and Notebook container:

   ```
   sudo docker exec -it notebook bash
   ```

   b) Run the command with root user:

   ```
   su - hdfs
   ```

   c) Run the commands with hdfs user. You must replace **`cm_sample`** with the actual tenant name.

   ```
   kinit -k -t /etc/security/keytabs/hdfs.headless.keytab hdfs
   hdfs dfs -rm -r -f /user/cm_sample
   ```

   d) Verify that the folder is deleted with the command:

   ```
   hdfs dfs -ls /user
   ```

   e) Exit hdfs user with the command:

   ```
   exit
   ```

2. Clear the hbase data and tables.
   a) Log in to Notebook node and Notebook container:

   ```
   sudo docker exec -it notebook bash
   ```

   b) Run the command with root user:

   ```
   su – hbase
   ```

c) Create a file with hbase user with the content:

Note: Assume that the file path is `/home/hbase/drop_tables.sql`

```
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_ANALYSIS_HISTORY;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_ANALYSIS_HISTORY;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE_BY_LOAD;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE_BY_AMI_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE_BY_AMI_AND_SCADA_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.METER_PHASE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE_BY_LOAD;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE_BY_AMI_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.TRANSFORMER_PHASE_BY_AMI_AND_SCADA_VOLTAGE;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.LATERAL_TRANSFORMER_PHASE_ERROR;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.KPI_HISTORY_BY_UTILITY;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.KPI_HISTORY_BY_SUBSTATION_REGION;
DROP TABLE IF EXISTS CMODEL_CM_SAMPLE.KPI_HISTORY_BY_FEEDER;
DROP SCHEMA IF EXISTS CMODEL_CM_SAMPLE;
DELETE FROM CMODEL.UTILITY where UTILITY='cm_sample';
DELETE FROM CMODEL.USER where UTILITY='cm_sample';
```

Note: The last two sql commands are case-sensitive. The schema name is always converted to uppercase in hbase, so the schema name is as CMODEL_**CM_SAMPLE**. However, in the tables, the tenant name remains the same as your created tenant, so in the last two sql commands, they are **cm_sample**.

d) Drop tables with file created in last step.

```
cd /usr/hdp/current/phoenix-client/bin
./psql.py master01:2181:/hbase-secure:hbase:/etc/security/keytabs/hbase.headless.keytab /
home/hbase/drop_tables.sql
```

e) Verify that the hbase tables are cleaned:

```
cd /usr/hdp/current/phoenix-client/bin
./sqlline.py master01:2181:/hbase-secure:hbase:/etc/security/keytabs/hbase.headless.keytab
!tables
Select * from CMODEL.UTILITY;
Select * from CMODEL.USER;
!exit
```

3. Disable the Jupyter Notebook and clean the local files.

a) Log in to Notebook node and Notebook container.

```
sudo docker exec -it notebook bash
```

b) Run the command to get the Jupyter Notebook process ID:

```
ps -ef | grep cm_sample
```

Here is the sample output, **235** that is the process ID.

```
cm_samp+   235     1  0 02:52 ?        00:00:00 /usr/bin/python2 /bin/jupyter-notebook --
ip=192.168.202.72 --port=8888 --notebook-dir=/home/cm_sample --certfile=mycert.pem --
keyfile mykey.key --no-browser
```

c) Stop the Jupyter process with the command:

```
kill -9 <process id>
```

For example:

```
Kill -9 235
```

d) Open `/etc/rc.local`, and delete the line with the tenant name.

For example: Delete this line in the file:

```
/opt/jupyter/startNotebook.sh cm_sample
```

e) Run the command to delete local files:

```
rm -rf /home/cm_sample
```

4. Clean the keytab files.

   Important: You must delete the `/etc/security/keytabs/`**`cm_sample`**`.keytab` file in these containers:

   - `kdc` Container in Ambari node.
   - Notebook container.
   - All HDP subordinate containers.
   - All HDP master containers.

5. Delete the tenant user in LDAP.

   a) Log in the App node and enter the `IFELdapNode` docker container.

   ```
   sudo docker exec -it IFELdapNode bash
   ```

   b) Create a file `/tmp/delete.ldif`. Replace the file content, **`cm_sample`** with tenant name.

   ```
   dn: cn=cm,ou=application,dc=ibmiot,dc=com
   changetype: modify
   delete: memberUid
   memberUid: cm_sample
   ```

   c) Run the command, replace *${LDAP_PASSWORD}* with actual ldapServerPassword that was provided during installation in the `config.properties` file, replace **`cm_sample`** with the tenant name.

   ```
   ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com"
   -w ${LDAP_PASSWORD} -f /tmp/delete.ldif
   ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com"
   -w ${LDAP_PASSWORD} "cn=cm_sample,ou=tenant,dc=ibmiot,dc=com"
   ```

6. Verify the removal of the tenant user:

   ```
   ldapsearch -x | grep cm_sample
   ```

Providing access permission to tenant user data

After you set up access for a tenant user, you need to give the rights for users to access the Connectivity Model from user interface.

Before you begin

Before you do you the following steps, ensure that there is a valid UI user. If a UI user does not exist, you must do either Adding users and user groups to access the user interface or “Modifying users, user groups, and passwords” on page 13.

About this task

The UI user for Connectivity Model does not have the permission to read the tenant data by default. Only the tenant user can access the connectivity model data. Doing this procedure enables the UI user to access the Connectivity Model data.

Procedure

1. Log in to the Notebook node and open the Notebook container.
2. Open the file `/home/cmopsadmin/cm/conf/input.txt` and edit the contents with the format:

   ```
   <utilityId>;<username>;<role1>,<role2>;
   ```

Where <utilityId> is the utility to access, <username> is the UI user to access the utility, and <role> is the role of the UI user. For example:

```
cm_sample;Bob;admin,user;
cm_sample;user1;user;
cm_utility;Bob;admin,user;
```

3. Run this command with root.

```
/home/cmopsadmin/cm/bin/APP_manageUtilityAccess.sh
/home/cmopsadmin/cm/conf/input.txt
```

Modifying the tenant user and changing tenant user passwords for the Connectivity Model application

You can add and remove the users of Connectivity Model, and change the passwords for the users.

Procedure

1. Log in to the App Node and open the `IFELdapNode` Docker container.

```
sudo docker exec -it IFELdapNode bash
```

2. To change the password for the user *cmtestuser*.
   a) Edit the `/tmp/temp.ldif` file to add LDAP data and commands.
   b) Add the script to the file:
      Where the *passw0rdChange* is the new password:

   ```
   dn:cn=cmtestuser,ou=tenant,dc=ibmiot,dc=com
   changetype: modify
   replace: userpassword
   userpassword: passw0rdChange
   ```

   c) Using the command to add the contents of the `temp.ldif` file into LDAP Server, replace `${LDAP_PASSWORD}` with actual *ldapServerPassword* that was provided during the installation in the `config.properties` file.

   ```
   ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD} -f /tmp/temp.ldif
   ```

3. Create an LDIF file to modify the group or user.

```
touch  /tmp/temp.ldif
```

4. To remove the user *cmtestuser* from the `cm` group.
   a) Edit the `/tmp/temp.ldif` file to add LDAP data and commands.
   b) Add the script to the file:

   ```
   dn:cn=cm, ou=application,dc=ibmiot,dc=com
   changetype: modify
   delete: memberUid
   memberUid: cmtestuser
   ```

   c) Using the command to add the contents of the `temp.ldif` file into LDAP Server, replace `${LDAP_PASSWORD}` with actual *ldapServerPassword* that was provided during the installation in the `config.properties` file.

   ```
   ldapmodify -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD} -f /tmp/temp.ldif
   ```

5. To delete the user *cmtestuse* from the system, you directly use the *ldapdelete* command:

```
ldapdelete -x -D "cn=Manager,dc=ibmiot,dc=com" -w ${LDAP_PASSWORD}
"cn=cmtestuser,ou=tenant,dc=ibmiot,dc=com"
```

Mapping user groups to license types

IBM Maximo Asset Performance Management for Energy & Utilities SaaS has standard user licenses and limited user licenses. To generate usage information for the IBM License Metric Tool, you must map each user role group to the relevant license type in the `slmtag_groups.properties` file on the application server.

Before you begin

For more information about license usage metrics in Maximo APM for E&U SaaS, see "License usage metrics" on page 27.

About this task

To map a user role group to a license type, edit the `slmtag_groups.properties` file that is packaged in the `ife_service_ui war` folder. There are two properties in the file: *groups_StandardUser* is the property for the standard user license, and *groups_LimitedUser* is the property for the limited user license.

Procedure

1. Log on to the application server as a user who has edit access to the `slmtag_groups.properties` file, for example, the wlp user.
2. Edit the `slmtag_groups.properties`. By default the file is in `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_frwk_app.ear/ife_service_ui.war/WEB-INF/classes/services` file.

   The default file content maps the admins sample group to the standard user license, and maps the users sample group to the limited user license:

   ```
   groups_StandardUser=admins
   groups_LimitedUser=users
   ```

3. To map a group to the standard user license, add the group name as a value for the *groups_StandardUser* property. Use a comma as the delimiter between group names.
   For example, `groups_StandardUser=standardUserGroup1,standardUserGroup2`.
4. To map a group to the limited user license, add the group name as a value for the *groups_LimitedUser* property. Use a comma as the delimiter between group names.
   For example, `groups_LimitedUser=limitedUserGroup1,limitedUserGroup2`.

Results

Usage information for the two types of licensed users in Maximo APM for E&U SaaS is generated for the IBM License Metric Tool.

Connecting to a Geographic Information System

Before you can use the applications Asset Performance Management, Asset 360 for Wind, and Connectivity Model, you need to connect to a Geographic Information System (GIS) system.

You can change the map default configuration for IBM Maximo Asset Performance Management for Energy & Utilities SaaS with the instruction for the specific application.

Map configuration controls for the Asset Performance Management application that includes the Asset Investment application

For Asset Performance Management the location for the map configuration is in: `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_ah_app.ear/ife_ah_web.war/config/model.json.`

For Asset Investment the location for the map configuration is in: `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_aip_app.ear/ife_aip_web.war/config/model.json`

The map features for Asset Performance Management use OpenLayers 3.5. Maximo APM for E&U SaaS can connect to all GIS systems that supports OpenLayers 3.5.

The controls for the map features are:

center
Specifies the initial center point. Center is an array that represents the `xy` coordinates. Example: `[-83.27366, 42.62893]`.

zoom
Specifies the initial zoom level, the range of values are 1 to `maxZoom`.

maxZoom
Specifies the maximum zoom level and depends on the `baseLayerUrl` provider.

baseLayerUrl
Specifies the base layer URL. Asset Performance Management uses `xyz` layers by default. The layer source for tile data with URLs in a set XYZ format that are defined in a URL template. Must include {x}, {y} or {-y}, and {z} placeholders.

```
"map":{
     "center":[-83.44, 42.60],
     "zoom":12,
     "maxZoom":19,
     "baseLayerUrl":"///server.arcgisonline.com/ArcGIS/rest/services
/World_Topo_Map/MapServer/tile/{z}/{y}/{x}"
```

Map configuration controls for the Wind 360 application

For Wind 360, the map configuration is in `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/wind_page.war/config/monitor/center.json`

The map features for Wind 360 use OpenLayers 3.5.

center
Specifies the initial center point. Center is an array that represents the `xy` coordinates. Example: `[-83.27366, 42.62893]`.

zoom
Specifies the initial zoom level, the range of values are 1 to `maxZoom`.

maxZoom
Specifies the maximum zoom level and depends on the `baseLayerUrl` provider.

baseLayerUrl
Specifies the base layer URL. Wind 360 uses `xyz` layers by default. The layer source for tile data with URLs in a set XYZ format are defined in a URL template. Must include {x}, {y} or {-y}, and {z} placeholders.

```
"Layers": [
     {
          "type":"ol.source.XYZ",
          "visible":true,
          "title":"XYZ",
          "Parameters":{
               "maxzoom":13,
               "url":"http://server.arcgisonline.com/ArcGIS/rest/services
/World_Topo_Map/MapServer/tile/{z}/{y}/{x}"
          }
     }
],
"view":{
     "center":[-8.8090, 33.2234],
     "zoom":2
}
```

Map configuration controls for the Connectivity Model application

The map for the Connectivity Model configuration depends on the tenant. Each tenant has its own map configuration. You can update the default map configuration that uses step 5 of Adding a Connectivity Model tenant user, "5" on page 17.

```
upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
values
    ('${utilityname}','${map_center}','${map_min_zoom}','${map_max_zoom}','${map_default_zoom}')
where utility = '${utilityname}';
```

map_center
: The center of map, for example: [-83.44, 42.60]

map_min_zoom
: The minimum zoom setting for the map. For example, 1.

map_max_zoom
: The maximum zoom setting for the map. For example, 22.

map_default_zoom
: The default zoom setting for the map. For example, 8.

For example: to change the utility `cm_sample`, you need to run the following `sql` on `hbase`:

```
upsert into CMODEL.UTILITY (utility,map_center,map_min_zoom,map_max_zoom,map_default_zoom)
values
    ('cm_sample','[-83.44, 42.60]','1','22','8') where utility = 'cm_sample';
```

Changing the number of columns in the list view

You can change the number of columns that shows in the list view of all applications of IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

The list view has an editing **Columns display** icon. Using the **Columns display** tool, you select the columns that show in the view.

A list view has these columns:

- Asset Name
- Feeder
- Container
- Health,
- Risk,
- Criticality,
- Failure,
- Effective age,
- Age.

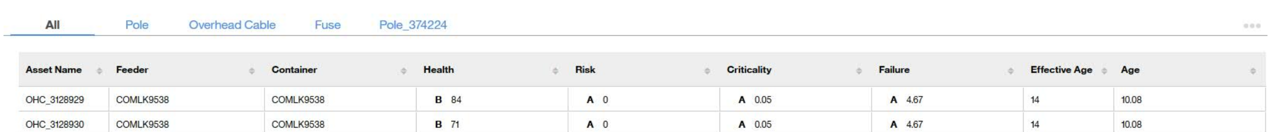


Figure 1. List view

Procedure

1. Sign on IBM Maximo APM for Energy & Utilities as a user.
2. Use the **Filter** selector to make a selection of the **Asset Class**, **Prioritization Criteria**, **Others**.

3. In the navigation segment bar click **List**.
   The list is displayed.
4. Click the **Column display** icon in the header bar.
5. Select or clear the columns that you need to view.
6. Click OK.

Results
Only the columns that you select show in the list view.

Monitoring system status

The monitoring of the system is fundamental to the maintenance of the solution and the applications.

To get the best from the applications, you can monitor the system on three levels: infrastructure, middleware, and web link URL. You monitor these system items to maintain the solution for each application and the status of each node.

Monitoring the Asset Performance Management and Asset 360 for Wind applications

These items are monitored to maintain the solution and status of each node for the Asset Performance Management and Asset 360 for Wind applications.

Infrastructure

SPSS Node, BI Node, IIB Node, DB Node
CPU
Disk
Memory
Network connectivity
Process
cpu_iowait
hostname_changed
max_open_file
max_processes
system_free_memory
system_swap_free
system_swap_pfree
system_total_memory
system_uptime
uname_changed

Middleware

SPSS node
status_spss_modeler_server

BI node
status_cognos_server

IIB node
status_ldap_server
status_http_server
status_liberty_server_framework_server

status_pmo_broker

status_pmo_queue_manager

DB Node
status_DB2_server

Web links URL

Asset Health & Wind 360 applications
URL links

Monitoring the Connectivity Model application

Monitor these items to maintain the solution and status for each node for the Connectivity Model application.

Infrastructure

Ambari node and all HDP nodes, including HDP subordinate, master, Notebook, and client nodes
CPU

Disk

Memory

Network connectivity

Process

cpu_iowait

hostname_changed

max_open_file

max_processes

password_file_changed

running_processes

system_free_memory

system_swap_free

system_swap_pfree

system_total_memory

system_uptime

uname_changed

HDP Services

HDFS node

Hbase

Web links URL

Connectivity Model URL link
URL link

Zeppelin URL link
URL link

Ambari interface URL link
URL link

Jupyter notebook URL links
URL links

License usage metrics

IBM License Metric Tool helps Passport Advantage clients determine their full and subcapacity PVU licensing requirements.

Learn more: IBM License Metric Tool.

IBM Maximo Asset Performance Management for Energy & Utilities SaaS

When IBM Maximo Asset Performance Management for Energy & Utilities SaaS is running, license management information is logged every day to the `/opt/IBM/energy/slmtags` directory on the application server in one tag file that is created for IFE framework and applications.

Maximo APM for E&U SaaS framework user and asset information

The file `ae796422b666e95033a951734467639f.slmtag` contains information for three types of usage:

Standard user
The usage information that is logged is the number of licensed standard users in the system.

Limited user
The usage information that is logged is the number of licensed limited users in the system.

Asset analytics
The usage information that is logged is the number of managed assets in the system. For the IBM Maximo Asset Performance Management for Energy & Utilities SaaS, this value is always 0.

Note: The numbers of licensed standard users and limited users in the system are retrieved from the basic user registry that is deployed with Maximo APM for E&U SaaS. To ensure the accuracy of these numbers, the configuration file that maps user groups to license types must be kept up-to-date.

The following content is an example of usage information from the `ae796422b666e95033a951734467639f.slmtag` file:

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
        <PersistentId>e137414b35d140dca5fd631df1098e0d</PersistentId>
        <Name>IBM IoT for Energy and Utilities</Name>
        <InstanceId>/opt/IBM/energy</InstanceId>
</SoftwareIdentity>
<Metric logTime="2017-07-20T17:20:55+08:00">
        <Type>AUTHORIZED_USER</Type>
        <SubType>Standard User</SubType>
        <Value>1</Value>
        <Period>
                <StartTime>2017-07-20T17:20:55+08:00</StartTime>
                <EndTime>2017-07-20T17:20:55+08:00</EndTime>
        </Period>
</Metric>
<Metric logTime="2017-07-20T17:20:55+08:00">
        <Type>AUTHORIZED_USER</Type>
        <SubType>Limited User</SubType>
        <Value>3</Value>
        <Period>
                <StartTime>2017-07-20T17:20:55+08:00</StartTime>
                <EndTime>2017-07-20T17:20:55+08:00</EndTime>
        </Period>
</Metric>
<Metric logTime="2017-07-20T17:20:55+08:00">
        <Type>ASSET</Type>
        <SubType>Assets in Asset Health</SubType>
        <Value>41278</Value>
        <Period>
                <StartTime>2017-07-20T17:20:55+08:00</StartTime>
                <EndTime>2017-07-20T17:20:55+08:00</EndTime>
        </Period>
</Metric>
```

Optimizing performance of the IBM Db2 database

You can use the **update configuration** and the **ALTER bufferpool** commands to optimize the IBM Db2 settings for the Asset Performance Management application in IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

Before you begin

You need access to the DB node for Maximo APM for E&U SaaS as the instance owner, for example, db2inst1.

The minimum hardware configuration for the database is 16 CPU cores and 32G of memory.

If hyper-thread functions are enabled and you use only one core, the **DFT_DEGREE** parameter must specify the degree of intrapartition parallelism based on the number of processors. The **NUM_IOCLEANERS** parameter must specify the number of page cleaners. If these parameters are not specified, then the Db2 SQL optimization ignores the intrapartition parallelism.

Procedure

1. Log in to the db node and change the user to db2inst1.
2. Run the following command to connect to Db2.

   ```
   db2 connect to ifedb
   ```

3. Run the following commands:

   ```
   db2 update db cfg for ifedb using DFT_DEGREE any
   db2 update db cfg for ifedb using SELF_TUNING_MEM OFF
   db2 update db cfg for ifedb using SHEAPTHRES_SHR 614400 automatic
   db2 update db cfg for ifedb using SORTHEAP 204800 automatic
   ```

   ```
   db2 ALTER  bufferpool IBMDEFAULTBP size 50000 automatic
   db2 ALTER  bufferpool BP32K_01 IMMEDIATE size 150000 automatic
   db2 ALTER  bufferpool BP32K_02 IMMEDIATE size 50000 automatic
   db2 ALTER  bufferpool BP32K_03 IMMEDIATE size 250000 automatic
   db2 ALTER  bufferpool BP32K_04 IMMEDIATE size 300000 automatic
   ```

4. Reset the current connection:

   ```
   db2 connect reset
   ```

Performing IBM Watson IoT Platform integration administration

You can integrate IBM Maximo Asset Performance Management for Energy & Utilities SaaS with IBM Watson IoT Platform to collect data from connected devices to provide data management, visualizations, and analytic capabilities from existing devices within Watson IoT Platform.

After you create an integration, all connected devices are detected and data collection begins. The files are automatically parsed to identify the variables, attributes, and dimensions.

The integration with Watson IoT Platform enables Maximo APM for E&U SaaS to instantly save the data on the IFE server. Every event is saved as a single file. When the connection is disconnected, the subscription is also disconnected and Maximo APM for E&U SaaS no longer saves the events.

Creating integrations

You can integrate IBM Maximo APM for Energy & Utilities with IBM Watson IoT Platform to collect asset data from devices that are connected to Watson IoT Platform.

Procedure

1. Sign on IBM Maximo APM for Energy & Utilities as an administrator.
2. Click **Administration** > **IOT** > **Add Integration**.

3. Type the following information.
    a) The name of the integration.
    b) The organization ID.
    c) The API key.
    d) The authentication token.

What to do next

Before you can connect to an integration, you must add a subscription to an iteration.

Adding a subscription to an integration

After you integrate with IBM Watson IoT Platform, you need to configure the integration by adding a subscription to the devices in IBM Watson IoT Platform.

Procedure

1. Click **Add Subcription** to the integration you need.
2. Type the name of the subscription.
3. The name and location of the subscription file is automatically generated. You can edit the subfolder name, but you must use the location of the integration folder.

   For example, if the current integration name is `INTG1`, the subscription name is SUB1 then the address shows as `INTG1/SUB1`.
4. Select the devices for this integration.

   You can add all devices, or make a selection.
5. Click **Save** to save the subscription choices.

Results

You can view the details of the devices in the list by **Device ID**, **Device Type**, **Events**, and **Date Added**.

Editing integrations

After you create an integration with a Watson™ IoT Platform organization, you can manage and edit the integration, for example, to change the devices from which you collect data.

Procedure

1. Click **Administration** > **IOT**.
2. Click the integration card that you need to edit.
3. If the integration is connected, click **Disconnect**.
4. Click **Edit** and edit as necessary.
5. Click the **Back** icon to exit the current configuration.

What to do next

You can click **Connect** to reconnect an integration.

Managing integrations

You can manage your integrations by connecting to and disconnecting from an integration. You can also delete an integration.

Procedure

1. Click **Administration** > **IOT**.
2. Hover over an integration card to view the connection status.

   You can disconnect or connect to an integration, depending on the status, and delete an integration.

3. If you select **Connect** or **Disconnect**, the status changes without further input.
4. If you select **Delete**, you are asked to confirm before deletion.
5. Click the **Back** icon to exit the current configuration.

Results

The event files are found on the IIB server at the location `/opt/IBM/energy/IOT/<the subscription location user input from UI>/<Device type name>/<Device id>/<event name>/yyyy_MM_dd_HH_mm_ss_SSS.txt`.

Archiving the event files

The archive facility in IBM Maximo Asset Performance Management for Energy & Utilities SaaS sends all the event files that are created before a target date to an archive. If no target date given then Maximo APM for E&U SaaS sends all the event files that are created before today's date.

Procedure

1. Find the archive script at the location `/opt/IBM/energy/IOT/archive.sh`.
2. Use the following syntax to start the archive:
   - The script `archive.sh /opt/IBM/energy/IOT` creates the archive to include all event files before the present date.
   - The script `archive.sh /opt/IBM/energy/IOT <target date yyyymmdd>` creates the archive to include all event files before the target date.

Results

You can find the archive at the location `/opt/IBM/energy/IOT/Archive/<date>.zip`.

Reusing existing LDAP Server for IBM Maximo Asset Performance Management for Energy & Utilities SaaS

You can use an existing LDAP server for application authentication in Maximo APM for E&U SaaS.

Preparing for configuring the LDAP registry for IBM Maximo Asset Performance Management for Energy & Utilities SaaS

You need to make sure that you can access the LDAP server from the Liberty server before you can gather the information that is for configuration.

Procedure

1. Log in the Liberty node and run the command:

   ```
   telnet <ldap server> portnumber
   ```

   where *<ldap server>* is the name of the LDAP server and the *portnumber* is the port that you want to test the connection to.

   Important: If you are using a Docker environment, this command must be run in the IFEAppNode container.
2. Add the Signer certificate for the LDAP server to the truststore for the liberty server.

   The following example is with OpenLdap.

   a) Run the command to find the location of the `ldap.conf` file:

   ```
   find / -name ldap.conf
   ```

   b) Open the `ldap.conf` file, and check the location of the `cacert.pem` file for TLS_CACERT.

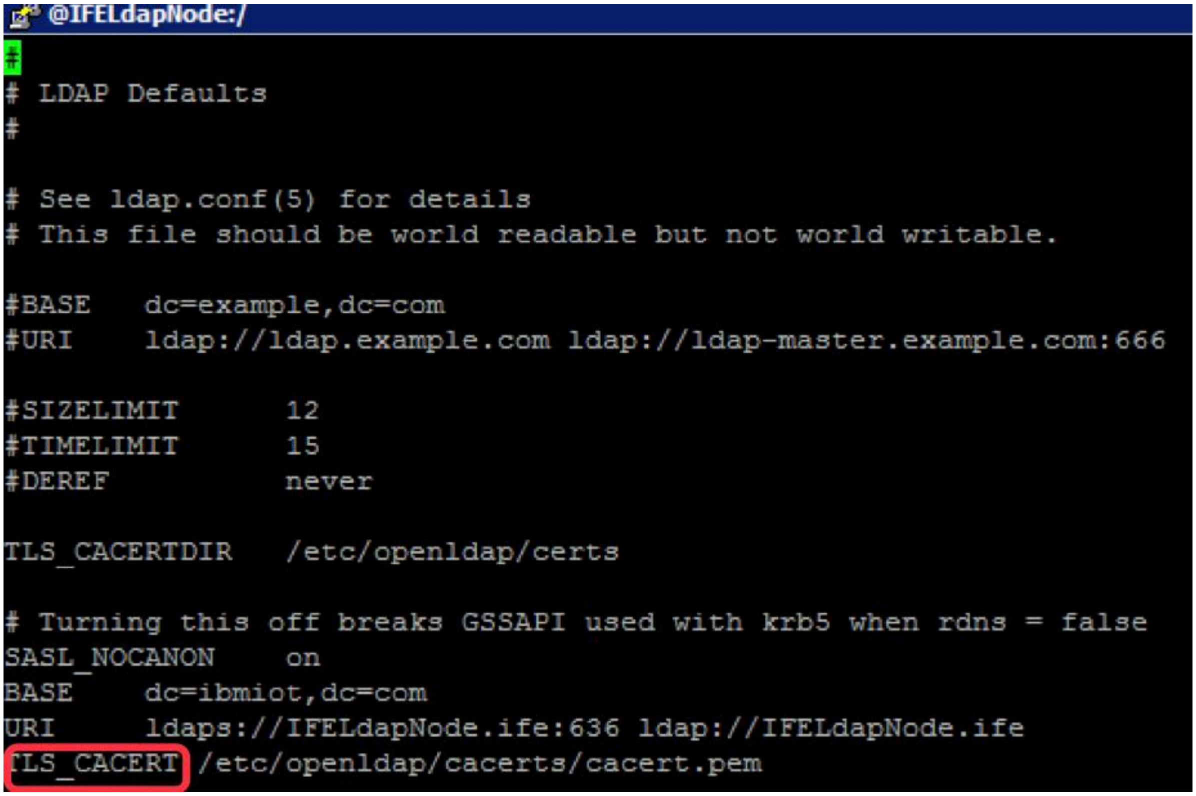


Figure 2. The location of cacert for TLS_CACERT

c) Copy the `cacert.pem` file to the server that has Liberty on it for importing into truststore later.

The `cacert.pem` for OpenLdap looks as follows:


```
-----BEGIN CERTIFICATE-----
MIIDLTCCAhWgAwIBAgIJAL9PxE/vLoR0MA0GCSqGSIb3DQEBCwUAMC0xCzAJBgNV
BAYTAkNOMRAwDgYDVQQIDAdCZWlqaW5nMQwwCgYDVQQKDANJQk0wHhcNMTcwOTA4
MDcwODIwWhcNMjcwOTA2MDcwODIwWjAtMQswCQYDVQQGEwJDTjEQMA4GA1UECAwH
QmVpamluZzEMMAoGA1UECgwDSUJNMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAr+UmpGi3rMLBNpOygp1N7OiNKmE55qZu56IxGX+byRBRN9Qi90uNujWI
rPGsd/2IBBzElCGNT31rbfySrdHdo9+YsJicuaZH6RAH4L7K3cyQfVMp6h9qqrxc
WHuD2WGvAvPbMkWKmGY/HFIcGo1hYESo7J9a9LcCQzNM9Od6Ho7Qo3IUnuiZ0wC3
pRrAXyVc4ywh6xfU05ZSxDdjGu2vqG5Z4xr1gmtCHOqYqUIEwlfT489lv+C5RNS3
sN7Lp5//OPvMR6CveVJiH88ENA7ukOR3A5FQrDWeTUyywEq+jTeYMD1D0DhCmiC8
JSzeJWYPoDFRYmDPMQYvFnSeTOrUswIDAQABo1AwTjAdBgNVHQ4EFgQUsnjOaXnF
qnfXEOREJOi+rRwDviYwHwYDVR0jBBgwFoAUsnjOaXnFqnfXEOREJOi+rRwDviYw
DAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAgEhVSEEyqv/QvrYMb6+L
aWoBKh9jLVQuX1nlE6MiejoSio0RW53GPG2Ejqkoa84qWC72eg/pQXBVPb0Tyefj
B2U+s7ynmCKbQUl5bpgh27/PhtwV44lOpew/++fjDf1/tS9/yutKaVBgiXdXSsgv
rIet7MB2FBtgNFpb6sFKhSh5zX67anfEPSxp6pjx/FBH1W65tXC9OHtfnF1/7U7L
MZNMRhMJwfIVGyWQPWdsecqpb9FnEO17XLCiDSIb0vLVhcO1TfIux3y6OWVLocI+
eMHj9rfOl70O/VqC3iIa+kVqMioCGz2SdBATzfG3wCfjWD5Y8dunPhHUsAnGAocU
8g==
-----END CERTIFICATE-----
```


d) Copy the `cacert.pem` file onto the `/tmp` directory of the Liberty node.

If you use the Docker environment for Maximo APM for E&U SaaS 2.5, do these steps to copy the `cacert.pem` file:

1) Copy the `cacert.pem` to the AppNode.

2) Copy the `cacert.pem` from the AppNode to IFEAppNode container in the `/tmp` directory on the App node.

```
sudo docker cp /tmp/cacert.pem IFEAppNode:/tmp
```

3. Gather information from the LDAP Server that is used for configuring the `ldapRegistry.xml` file.

a) Get the user group information.

Check with your LDAP administrator for the user group name or search on the LDAP server:

1) You need top know how users are grouped in the LDAP server, for example in OpenLDAP the alternatives are `objectClass posixGroup` and `objectClass groupofNames`.
2) Run the command to search, and find the corresponding entries that match objectClass value.

```
ldapsearch -x
```

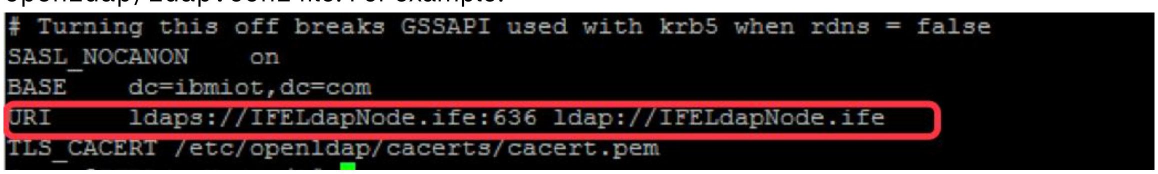
An example output looks as follows:

```
…….
# admins, ife, application, ibmiot.com
dn: cn=admins,cn=ife,ou=application,dc=ibmiot,dc=com
objectClass: top
objectClass: groupOfNames
member: cn=Bob,cn=ife,ou=application,dc=ibmiot,dc=com
cn: admins
……
```

3) Check with your LDAP administrator to understand which user groups or users you need to access the Maximo APM for E&U SaaS offering.

b) Get the LDAP server connect information.

1) Check that the LDAP server is enabled for SSL communication and get LDAP port number. For example: OpenLDAP normally uses port 389 for non-SSL by default and port 636 for SSL communication. In the Maximo APM for E&U SaaS version 2.5, the configuration is in the `/etc/openldap/ldap.conf` file. For example:



```
# Turning this off breaks GSSAPI used with krb5 when rdns = false
SASL_NOCANON    on
BASE    dc=ibmiot,dc=com
URI     ldaps://IFELdapNode.ife:636 ldap://IFELdapNode.ife
TLS_CACERT /etc/openldap/cacerts/cacert.pem
```

Figure 3. An example configuration for SSL communication

2) Obtain the Base DN or Bind attributes for the LDAP server. Check with your LDAP administrator for this information.
3) Obtain the administrative password for LDAP to authenticate with the LDAP server. Check with your LDAP administrator for this information. For example, in OpenLDAP, the configured is in `/etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif` file as the `olcRootPw` attribute.

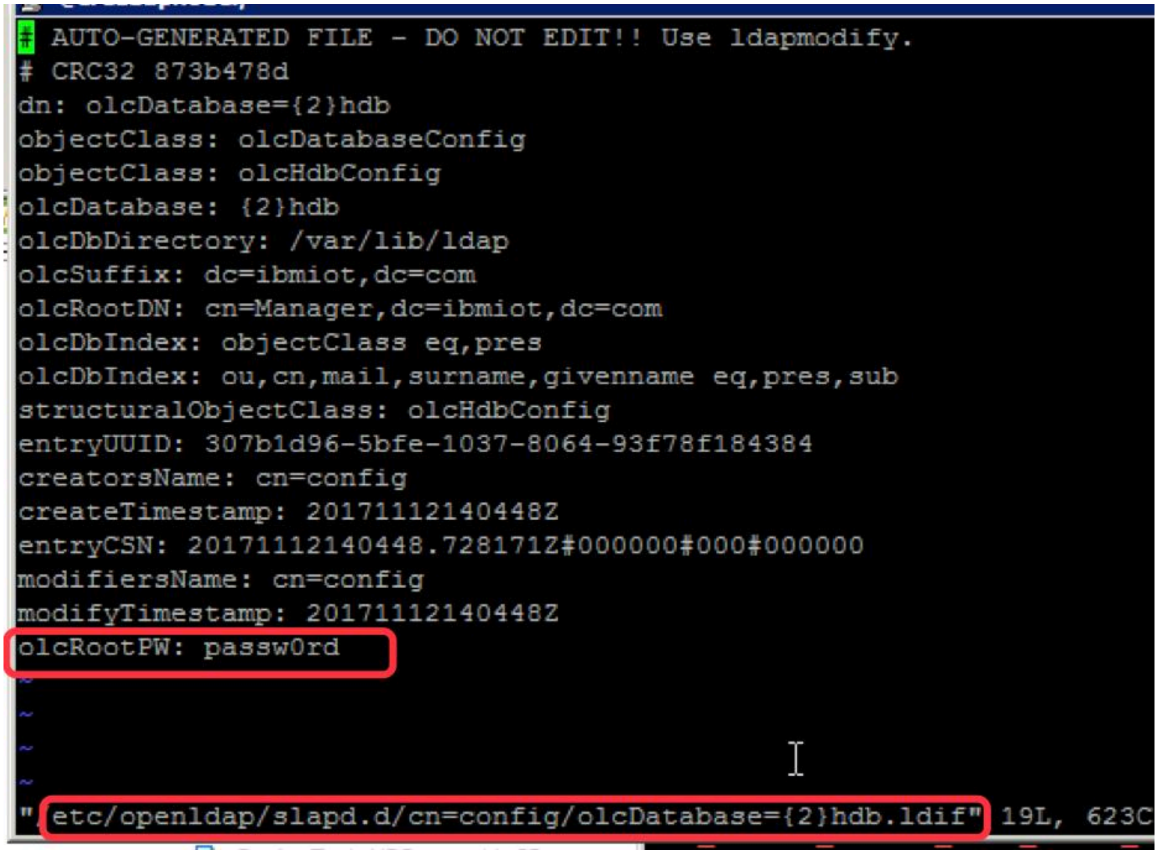


Figure 4. The attribute olcRootPw

4) Set the baseDN for the point from where an Ldap server searches for users. For example: , if you have a bindDN `cn=admin, dc=example, dc=com`, then you can use `dc=example,dc=com` as baseDN.

4. Retrieve the password details.

   The password for access to the liberty keystore `key.jks` is provided during installation. It is used when you import the Signer certificate into the liberty keystore. In Maximo APM for E&U SaaS V2.1, this password is set during installation. In Maximo APM for E&U SaaS V2.5 has a default value that is `passw0rd`.

Configuring the LDAP user registry in Liberty

Before you can add access for a new user group, you need to configure the LDAP user registry.

Procedure

1. Import the LDAP certificate.
   a) Locate the path to `keytool`.
   b) Run the commands to import the LDAP certificate and specify a password for the keystore `ldapKeyStore.jks`. Replace the path value to where `cacert.pem` is copied and the password for access to `key.jks`.

   For example, for OpenLdap use:

```
IM_Java=`find / -name jre_* | grep InstallationManager/eclipse`
$IM_Java/jre/bin/keytool -import -alias ldap_cert -noprompt -storepass passw0rd -
file /tmp/cacert.pem -keystore /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/
resources/security/ldapKeyStore.jks

$IM_Java/jre/bin/keytool -import -alias ldap_cert -noprompt -storepass
<setYourLdapKeystorePassword> -file /tmp/cacert.pem -keystore /opt/IBM/WebSphere/
Liberty/usr/servers/framework_server/resources/security/key.jks
```

2. On the Liberty Node, run the commands to comment out the `basicRegistry` in `server_ife_frmk.xml` and `server_wind_web.xml` files.

   Run this command for all applications:

   ```
   sed -i 's/<basicRegistry/<!-- basicRegistry/g' /opt/IBM/WebSphere/Liberty/usr/servers/
   framework_server/server_ife_frwk.xml
   sed -i 's/basicRegistry>/basicRegistry -->/g' /opt/IBM/WebSphere/Liberty/usr/servers/
   framework_server/server_ife_frwk.xml
   ```

   Run this command for Asset 360 for Wind only:

   ```
   sed -i 's/<basicRegistry/<!-- basicRegistry/g' /opt/IBM/WebSphere/Liberty/usr/servers/
   framework_server/server_wind_web.xml
   sed -i 's/basicRegistry>/basicRegistry -->/g' /opt/IBM/WebSphere/Liberty/usr/servers/
   framework_server/server_wind_web.xml
   ```

   Note: The user groups in the `server_ife_frwk.xml` file apply to all applications. The user groups in `server_wind_web.xml` file apply for Asset 360 for Wind only.

3. Create a file: ldapRegistry.xml.

   Replace the **Bold** sections with actual values.

   ```
     <server>
       <featureManager>
         <feature>ldapRegistry-3.0</feature>
         <feature>appSecurity-2.0</feature>
         <feature>ssl-1.0</feature>
       </featureManager>
     <ldapRegistry ldapType="Custom" port="ldap_port" realm="WebRealm"
   host="ldap_hostname" bindDN="bindDN" bindPassword="ldapAdminPassword"
   id="ldapid" baseDN="baseDN" sslEnabled="sslEnabled" sslRef="ldapssl">
       <customFilters userFilter="(&amp;(cn=%v)(objectclass=person))" groupFilter="(&amp;(cn=%v)
   (|(objectclass=posixGroup)(objectclass=groupOfNames)))"
   groupMemberIdMap="posixGroup:memberUid;groupOfNames:member">
       </customFilters>
     </ldapRegistry>

       <ssl keyStoreRef='ldapKeyStore' id="ldapssl"></ssl>
       <keyStore id="ldapKeyStore" location="${server.config.dir}/resources/security/
   ldapKeyStore.jks"  type="JKS" password="ldapKeystorePassword" />

       <ltpa keysFileName="${server.config.dir}/resources/security/ltpa.keys"
   keysPassword="SetYourLtpaPassword"></ltpa>

     </server>
   ```

   Note: 1. If you do not want to use plain text passwords, you can use `securityUtility` to get an encrypted string to use in the xml file, for example:

   ```
   /opt/IBM/WebSphere/Liberty/bin/securityUtility encode <password string>
   ```

   Note: 2. Update `customFilters` segment to filter out the user groups or users in LDAP server who need to access Maximo APM for E&U SaaS. Check the Liberty IBM Knowledge Center for details: https://www.ibm.com/support/knowledgecenter/en/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/twlp_sec_ldap.html.

   Note: 3. Maximo APM for E&U SaaS version 2.5 has a built-in `OpenLDAP` server. Liberty uses this `OpenLDAP` server for authentication. The `ldapRegistry.xml` in Maximo APM for E&U SaaS version 2.5 is configured as follows:

```
<server>
    <featureManager>
        <feature>ldapRegistry-3.0</feature>
        <feature>appSecurity-2.0</feature>
        <feature>ssl-1.0</feature>
    </featureManager>

    <ldapRegistry ldapType="Custom" port="636" realm="WebRealm"
        host="IFELdapNode.ife" bindDN="cn=Manager,dc=ibmiot,dc=com" bindPassword="{xor}Lz4sLChvLTs="
        id="ldapid" baseDN="dc=ibmiot,dc=com" sslEnabled="true" sslRef="ldapssl">
             <customFilters userFilter="(&amp;(cn=%v)(objectclass=person))"
             groupFilter="(&amp;(cn=%v)(|(objectclass=posixGroup)(objectclass=groupOfNames)))"
             groupMemberIdMap="posixGroup:memberUid;groupOfNames:member">
             </customFilters>
    </ldapRegistry>

    <ssl keyStoreRef='ldapKeyStore' id="ldapssl"></ssl>
    <keyStore id="ldapKeyStore" location="${server.config.dir}/resources/security/ldapKeyStore.jks"  type="JKS" password="{xor}Lz4sLChvLTs=" />

    <ltpa keysFileName="${server.config.dir}/resources/security/ltpa.keys" keysPassword="{xor}Lz4sLChvLTs="></ltpa>
</server>
```

Figure 5. The details for the LDAP Registry

4. Regenerate `ltpa.keys`.
   a) Delete the previous `ltpa.keys` file in the folder `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/resources/security/`.

   ```
   rm -rf  /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/resources/security/
   ltpa.keys
   ```

   The `ltpa.keys` file is generated again after you restart `framework_server`.

   b) Restart the `framework_server` to update `server.xml` to include `ldapRegistry.xml`, and restart `framework_server` for the LDAP user registry to take effect.

   ```
   sed -i '$d' /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/server.xml
   addConfigure='\t<include location="${server.config.dir}/ldapRegistry.xml" />\n</server>'
   echo -e $addConfigure >> /opt/IBM/WebSphere/Liberty/usr/servers/framework_server/
   server.xml

   /opt/IBM/WebSphere/Liberty/bin/server stop framework_server
   /opt/IBM/WebSphere/Liberty/bin/server start framework_server
   ```

Configuring the Vegetation Management application

You use the REST API to make the connections and import data to the Vegetation Management application. You configure the application, so it meets the needs of the business users.

To set up the environment for the Vegetation Management application, sample Python Notebooks are available from the dashboard of IBM Maximo Asset Performance Management for Energy & Utilities SaaS. The sample Notebooks are used to load data and run analyses. To open and configure the Notebooks, you must have the role of an administrator, data scientist, or data-loading developer.

The configuration of the Vegetation Management application includes an optional topic for creating corridor data that is based on the overhead cable data. You must do this task if you do not have corridor data available in your EAM system.

Loading asset data into the custom data model

You use the REST API to make the connections and import data into the Vegetation Management application in IBM Maximo Asset Performance Management for Energy & Utilities SaaS. You configure the application to meet the needs of the business users by using a Python Notebook.

You must have the role of data scientist or administrator to import data.

The following steps are required to import asset data from an existing EAM system into the custom data model:

1. Export the asset data from an existing EAM system.
2. Prepare to load the data to the custom data model.
3. Load data into the custom data model.

The detailed instructions are in the `LoadAssetData.ipynb` Python Notebook available on the Maximo APM for E&U SaaS desktop.

Creating corridor data based on overhead cable data

When you have asset data in IBM Maximo Asset Management (SaaS) but not corridor data, you can use this Notebook to generate the data based on overhead cable data.

Before you begin

Two queries must exist in Maximo Asset Management (SaaS) for `mx_feeder_query` and `mx_ohc_query`, for example:

- `mx_feeder_query = "query_location_feeder_lily"`
- mx_ohc_query = "query_location_ohc_lily"

About this task

If you do not have corridor data, you can use the sample code to generate corridor data based on overhead cables. The detailed instructions are in the `LoadCorridorData.ipynb` Python Notebook available on the Maximo APM for E&U SaaS desktop.

Procedure

1. Define the variables.
2. Install and integrate the Shapely Python package.
3. Define the functions.
4. Optionally, delete unwanted corridor, corridor segment, zone, and location information.
5. Query the locations.
6. Create a union of polygons for the corridor.
7. Insert corridor data into Maximo Asset Management.
8. Generate the model for the corridor segment.

Loading vegetation data to the custom data model

You use the REST API to make the connections and import vegetation data into the Vegetation Management application in IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

Before you begin

You must have the role of data scientist or administrator to import data.

About this task

The detailed instructions are in the `Load_Vegetation_FromTWC` Python Notebook available on the Maximo APM for E&U SaaS desktop.

Procedure

1. Enter the IBM PAIRS Geoscope credentials and coordinates.
2. Enter the Cloud Object Storage credentials.
3. Enter the Vegetation Management API Credentials.
4. Install the dependency libraries.
5. Define the utility functions.
6. Download the vegetation layer from PAIRS Geoscope.
7. Generate the shape file that is used for a mask.
8. Convert the PAIRS tiff file to a vegetation polygon shapefile.
9. Upload the vegetation data to the Vegetation Management API.

Running a Vegetation Management analysis

After you load the asset and vegetation data to IBM Maximo Asset Performance Management for Energy & Utilities SaaS, as a data scientist you can create an analysis of the vegetation data for the vegetation manager.

Before you begin

You need the role of data scientist to be able to run an analysis.

About this task

The detailed instructions are in the `Run_Vegetation_Analysis.ipynb` Python Notebook available on the Maximo APM for E&U SaaS desktop.

Procedure

1. Enter the Vegetation Management API Credentials.
2. Install the dependency libraries.
3. Define the utility functions.
4. Download data from the Vegetation Manager API.
5. Run an analysis.
6. Upload the model outputs to the custom data model.

Configuring the Standard Operating Procedures

A Standard Operating Procedure (SOP) is a set of instructions that describes all the relevant steps and activities of a process or procedure.

When you define an SOP, you define activities that are included in the SOP. SOP enables an administrator to organize personnel, information, and tasks in response to events and incidents in order to achieve a comprehensive control of the operation. A SOP consists of these components:

Standard Operating Procedure Definition
An SOP definition is the template that is used when a SOP is instantiated in response to a particular occurrence. A SOP Definition is made up of activities that are described by Activity Definitions.

Activity Definition
A SOP Definition contains one or more Activity Definitions. An activity definition sets the individual instructions that need to be run as part of the SOP.

SOP Instance
A single Instance of an SOP in response to a particular event or occurrence. One SOP Definition can be used for many SOP Instances. An SOP Instance can be in one of these states.

- Active
- Started
- Stopped
- Completed
- Canceled

Activity Instance
An Activity Instance is the instantiation of a single Activity Definition. A single Activity Definition can be used to create multiple Activity Instances. An Activity Instance can be in a number of states:

- Active
- Waiting
- Started
- Skipped
- Completed

References
Supplemental information that is relevant to a Standard Operating Procedure or Activity. References can also be used to define email templates.

Roles
There are two abilities, Owners and Readers that can be set against administrative and user roles.

- A Reader can monitor the activities that are associated with a standard operating procedure.
- An Owner can monitor and complete the activities that are associated with the standard operating procedure.

Activity Type
The Activity Type describes the response to the activity. The activities can be of different types and execution models. Any combination of different activities in an SOP is allowed.

- Manual: This type of activity must be manually carried out by the owner of the SOP.
- If-Then-Else Activity: A conditional activity that allows branching based on specific criteria. The user can choose which of the SOP definitions to instantiate when starting the activity. Either enter or select values for Then and Else.
- Alert Activity: This activity displays an email template for the SOP owner to complete and send an email notification to predefined personnel.
- REST Activity: An activity that creates a REST service call. The user can specify the service URL and any required authentication information to be started when the activity is started.
- SOP Activity: An activity that starts another standard operating procedure.

Roles for Standard Operating Procedures

The abilities for each of the roles for SOPs are as follows:

SOP Administrator roles

- View and delete an SOP definition
- Launch, view, and edit an SOP instance
- Start and complete activities in an SOP instance

SOP author roles

- Create, edit, view, and delete an SOP definition
- Create an SOP draft
- View, edit, and delete an SOP activity
- Submit an SOP draft for approval
- Approve an SOP draft

Reference Librarian Role
Create shared references

Owner Roles (SOP definition)

- Create an SOP draft
- View, edit, and delete an SOP definition
- Edit and delete an SOP activity
- Submit an SOP draft for approval
- Approve an SOP draft
- Launch, view, and edit an SOP instance

Reader roles (SOP Definition)

- View an SOP definition
- View an SOP instance from My Activities

- View an SOP activity, provided the user has Reader role in the Activity definition

Owners roles (SOP activity)

- View an SOP instance from My Activities
- Start and complete activities in an SOP instance for their own activities from My Activities

Reader roles (SOP activity)
View SOP instance from My Activities

Approval lifecycle for a Standard Operating Procedure

An SOP definition can assume different status during its lifecycle.

- Draft: When the SOP is first created, a draft version is saved initially. From an approved version of an SOP, it is also possible to create another draft version, when it is necessary to change the SOP definition that uses the approved version as a base. A draft can be edited, submitted for approval, or discarded.
- Pending approval: This is a draft SOP definition that is submitted for approval, ready to be approved or disapproved. The name of the version is defined in this status and it names the SOP definition version if approved. If this version is not approved, the SOP definition goes back to the draft version status.
- Approved: When an SOP definition is approved, it is ready to be started.

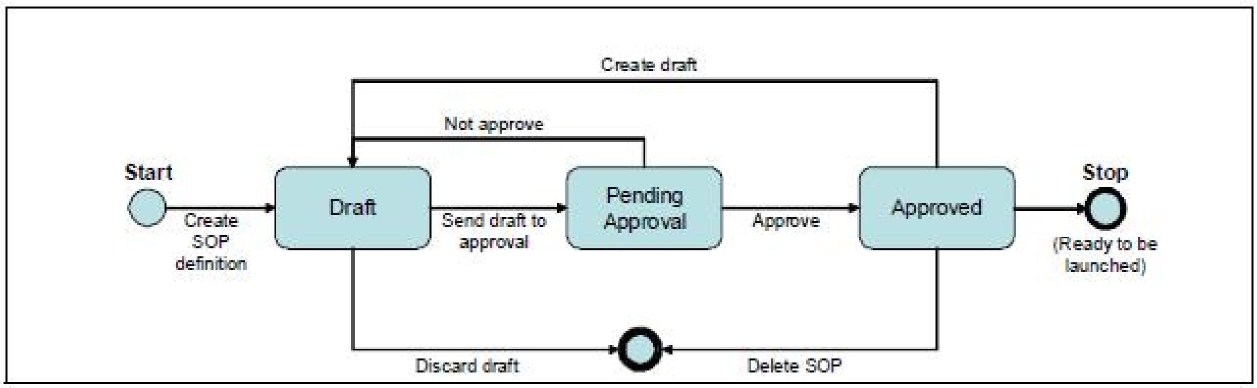


Figure 6. SOP Life Cycle

Defining a Standard Operating Procedure

Define a Standard Operating Procedure for all activities.

Procedure

1. In IBM Maximo Asset Performance Management for Energy & Utilities SaaS, click **Administration** > **Standard Operating Procedures** > **SOP Definition**.
2. In the **Basics** section, define the basic information for the SOP.
   a) Click **Create**.
   b) In the **Basics** section, type a name and description for the SOP.
   c) If the activities need to be carried out in a particular order, select **Activities are done in order** in the **General Settings** field.
   d) Click **Add References** to add supplemental information.
   e) Click **Next** to go to the **Roles** section.
3. In the **Roles** section, define the roles for the SOP.
   a) Select the **Roles** that will be assigned as **Owners** and **Readers** for the SOP created with this definition.
      - If you want the role to be able to monitor activities that are associated with the standard operating procedure, select **Reader**.

- If you want a role to be able to monitor and complete activities that are associated with the standard operating procedure, select **Owner**.

b) Click **Next** to go to the **Activities** section.

4. In the **Activities** section, define the activities for the SOP.

a) Click **Add**.

b) If the activity is required, select **Required**. If not, the SOP can move on to the next activity during instantiation.

c) Select **Autostart** if this activity is to start automatically without owner operation.

d) You can select the roles for the Owners and Readers. If not, the roles are inherited from the SOP definition.

e) Set the duration of the activity. The duration is the length the activity takes once started.

f) Type a description of the activity.

g) Set the Activity type.

The options are:

- Manual Activity
- If-Then-Else Activity
- Alert Activity
- REST Service
- SOP Activity

h) To add more activities click **Add** and define the next activity for the SOP.

i) Click **Next** to go to the **Summary** section.

5. Review the SOP.

a) Review the information in the summary.

b) Click **Save**.

Important: The SOP is in now draft state. The SOP must be approved before it can be instantiated.

In the draft state, the SOP can be edited or deleted if not required.

Configuring Standard Operating Procedures for different activities

One of the important tasks for when defining a Standard Operating Procedures (SOP) is to define the set of activities that composes the procedure.

The activities can be of different types and execution models. Any combination of different activities in an SOP is allowed.

Configuring activities to start in sequence

When you specify that activities are done in sequence, you will not be able to start an activity until the predecessor activity is completed.

Whether the activities of an SOP are executed sequentially or not is specified in the General Settings section of the SOP definition.

Specify that activities must be executed sequentially where the activities must be executed in a chronological order or where activities depend on the result of previous activity.

Examples of sequential activities are:

- MA1 - Approve an operation.
- SA2 - Collect information about and incident area and the number of victims.
- SA3 - Prepare personnel for standby.
- SA4 - Redirect traffic to clear an incident area.

The Sequential activities figure shows an activity flow with activities that run in sequential order.

Note:

- The user can start SA2 only if MA1 is complete, SA3 after SA2 is complete, and SA4 after SA3 is complete. The sequential property of an SOP applies to all activities, either all activities are sequential or none for the entire SOP.
- t1, t2, t3 and t4 represents the duration of the activity.

Configuring required activities

A required activity is one that is mandatory in a Standard Operating Procedure (SOP). You cannot skip over an activity that is specified as a required activity in the SOP definition.

Any of the activity types can be configured as required. Required is an attribute of an activity therefore an SOP can have activities that are required and activities that are optional (they can be skipped over). Examples of required activities are:

- MA1 - Approve operation. Mandatory.
- A2 - Collect information about incident area and number of victims. Optional.
- A3 - Prepare personnel for standby. Optional.
- A4 - Redirect traffic to clear incident area. Optional.

The mandatory activities figure shows an activity flow with activities that have mandatory and optional activities.

Note:

- MA1 is mandatory. All the other activities are optional and do not need to be performed sequentially.
- A2, A3, and A4 can be run in parallel.
- t1, t2, t3 represents the duration of the activity.

Configuring for manual activities

A manual activity type is an activity that is a manual task of the owner after the Standard Operating Procedure is started.

Manual activities are the most basic and essential type. Examples of manual activities are:

- A1 - Collect information about the incident area and number of victims.
- A2 - Prepare personnel for standby.
- A3 - Redirect traffic to clear incident area.

The manual activities figure shows an activity flow with activities that are started manually.

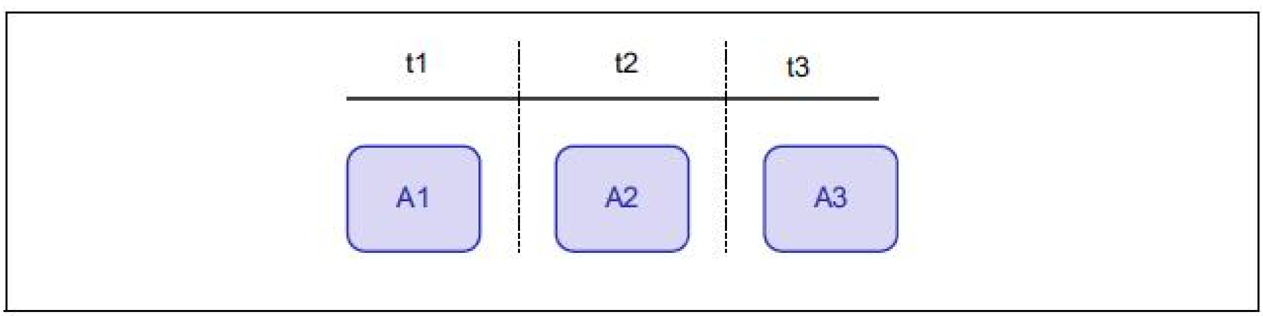


Figure 7. Manual activities

Note: A1, A2, and A3 are all manual activities. t1, t2, t3 represents the duration of each activity.

Configuring conditional activities

Conditional activities are conditional activities that allow branching based on If-Then-Else criteria.

When the activity is launched, the operator decides whether to execute one set of actions (Then) or another set (Else).

The conditional activity is useful in situations where you want to use the results from other activities to make a decision and launch a new SOP to respond to the situation. Examples of a conditional activity is:

- MA1 - Approve operation.
- SA2 - Collect information about incident area and number of victims.
- SA3 - Prepare personnel for standby.
- SA3 If (team is not available = true) Then initiate contingency plan. Conditional activity.
- SA4 - Redirect traffic to clear incident area.

The conditional activity figure shows a conditional activity flow.

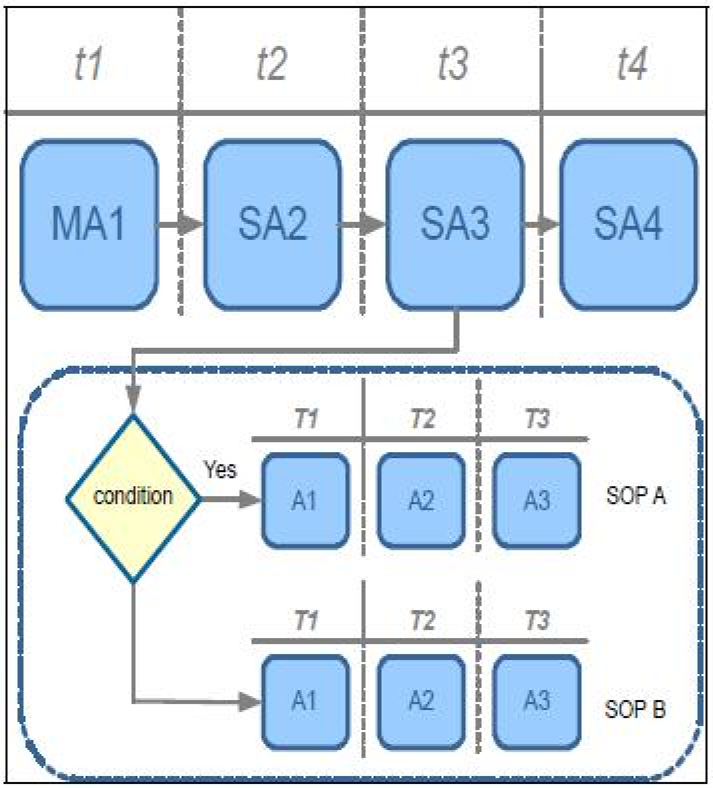


Figure 8. Conditional activity

When the SOP instance runs, the operator decides whether to launch the Then SOP or the Else SOP, if one is available, by selecting **Start Then** or **Start Else** options in my activity page.

Configuring notification activities

Notification activities enable the operator of IBM Insights Foundation for Energy to complete an email an send it. The email notification occurs as part of the required activity.

You need to set up an SMTP server to send the email notifications before configuring and launching a notification activity.

To set up an SMTP server, go to the `sysprop.json` table in the IFE database and update the following system properties in the SOP group:

- MailServerHostname: hostname of the SMTPserver
- MailServerPort: SMTP server port
- MailSender: sender of emails to be sent by the activity in the **From** field.

Email templates can be created and then reused for a notification activity. All email templates are stored as **References**.

Here are example templates There are some email examples created as references, please access them using these urls where <liberty server ip> is the ip address of the IFE server:

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_fireEvent.txt
```

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_bombThreat.txt
```

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_disturbanceEvent.txt
```

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_evacuation.txt
```

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_prep_for_power_loss.txt
```

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_radiationHazard.txt
```

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_transitionToNewERlevel.txt
```

```
https://<liberty server ip>:9443/ibm/ife/sop/app/js/ife/sop/email/emailTemplate_weatherEvent.txt
```

To create a notification template:

1. Create a text file that contains the email template. The To, Subject, and Body fields are optional. Here is an example template:

   ```
   ----------------------------------------------------------------------------------------------------
   To:
   Subject:Attention: Emergency incident detected
   Body:Emergency incident detected. Please contact administrator for further
   Information.
   ----------------------------------------------------------------------------------------------------
   ```

2. Publish the template in a location that accessible from IBM Insights Foundation for Energy.
3. Create a reference in the Standard Operating Procedures References and point the address to the URL where the template is published.
4. Make sure the first line of the description field contains only the keyword `NOTIFICATION`.

   You can add further description details in subsequent lines.

Note: If you receive a 401 error when you try to load the template into the email window, you need to include your credentials in the URI, for example: https://user_name:password@<host>:<port>/test-url/notification.

Creating a reference for Standard Operating Procedures

References are supplemental information relevant to an SOP or an activity. References can also be used to define e-mail templates.

About this task

A user can create references for Standard Operating Procedures.

Procedure

1. In Insights Foundation for Energy, click **Administration** > **Standard Operating Procedures** > **References**.

2. Click **Add**.
3. In the **Name** field, type the name of the reference.
4. In the **URI** field, type or paste the web accessible address.
5. In the **Description** field, type a description of the reference.
6. If you want to restrict the use of the reference, select the **Private** check box. Otherwise the reference can be shared to other users.

Editing a Standard Operating Procedure

Edit an Standard Operating Procure.

About this task

To be able to edit an approved Standard Operating Procedure (SOP), the SOP must be returned to a draft version.

Note: You can always edit an SOP in a draft version.

Procedure

1. In Insights Foundation for Energy, click **Administration** > **Standard Operating Procedures** > **SOP Definition**.
2. From the list of SOP definitions, select the SOP to be edited.
3. From the **Version** field, select the version to be edited.
4. Click **Create Draft**.

   The SOP state is now in draft.
5. Click the SOP you want to edit.
   - To add an activity, click **Add** in the **Activities** tab.
   - To edit an activity, click the activity and then click **Edit**.
   - To delete an activity, click the activity and then click **Delete**.
   - To edit a role, select the **Roles** tab and then click **Edit**.
   - To edit references, select the **References** tab, click **Edit** or **Delete** for a specific reference or click **Add** to add a new reference.
6. When the edits are complete, click **Save**.

What to do next

Before the edits are accepted, you must submit the draft version for approval.

Submitting a draft Standard Operating Procedure for approval

You can submit a draft version of an SOP for approval or discard it.

Before you begin

The SOP must be set to Draft version, and you must have the administrator role.

Procedure

1. In Insights Foundation for Energy, click **Administration** > **Standard Operating Procedures** > **SOP Definition**.
2. Select the SOP for submission that is in **Draft** state, and click **Submit for Approval**.

   Attention: An SOP that has been submitted for approval cannot be edited. The administrator can either Approve or Disapprove the SOP.

Testing a Standard Operating Procedure

Test an approved version of a Standard Operating Procedure (SOP).

About this task

An SOP must be in an approved state before it can be tested.

Procedure

1. In Insights Foundation for Energy, click **Administration** > **Standard Operating Procedures**.
2. Click on the approved SOP Definition that you want to test.
3. Click **Definition Actions** and from the drop-down list select **Launch**.

   Once launched, the SOP Definition list will update to indicated the number of Active SOPs based on this definition.

   The **My Activities** banner indicator shows that there are one or more activities that require attention.
4. You can perform the SOP and check for completeness.
5. If the SOP needs further work, click **Create Draft** to return the SOP to a draft state.

Exporting a Standard Operating Procedure

You can export your Standard Operating Procedure definitions as an XML file. This capability is useful for migration purposes.

About this task

To export an SOP definition performs the following steps:

Procedure

1. In Insights Foundation for Energy, click **Administration** > **Standard Operating Procedures** > **SOP Definition**.
2. Select **Standard Operating Procedures Definition** administration page.
3. From the **Definition Actions** drop-down list, select **Export All**.
4. In the pop-up window that is displayed, click **Save File** > **OK**.
5. Navigate to the directory where you want to save the `SopDefinitions.xml` file and click **Save**.

Importing a Standard Operating Procedure

You can import a previously exported Standard Operating Procedure definitions file. This capability is useful in cases where the organization has predefined SOPs that were implemented using a different tool.

About this task

To import an SOP definition performs the following steps:

Procedure

1. In IBM Maximo APM for Energy & Utilities, click **Administration** > **Standard Operating Procedures** > **SOP Definition**.
2. Select **Standard Operating Procedures Definition** administration page.
3. From the **Definition Actions** drop-down list, select **Import**.
4. In the **Import Standard Operating Procedures** window, select the XML file to upload.

   The imported definitions are displayed in the list of SOP definitions. You can import an SOP definition that has the same name as an existing SOP definition. In this case, the existing SOP definition is not overwritten. Instead, two SOP definitions with the same name are displayed in the list of definitions.

5. Click **Import**.

Reverting to a particular version of a Standard Operating Procedure

You can revert back to a particular version of a Standard Operating Procedure (SOP).

About this task

The Standard Operating Procedures shows the latest version of a definition. If you need to revert to a earlier version of a definition, you can use these steps.

Procedure

1. In Insights Foundation for Energy, click **Administration** > **Standard Operating Procedures** > **SOP Definition**.
2. From the list of SOP definitions, select the SOP to be edited.
3. From the **Version** field, select the version to be reverted to.
4. Click **Create Draft**.

   The SOP state is now in draft.

What to do next

Before the revert is accepted, you must submit the draft version for approval.

Viewing a Standard Operating Procedure

You can view a Standard Operation Procedure (SOP) as an administrator.

There are two ways to view an SOP instance:

- From the SOP Administration page.
- From the **My Activities** widget.

Viewing a Standard Operating Procedure as an administrator

You can view a Standard Operation Procedure (SOP) as an administrator.

About this task

To view the SOP instance from the SOP Administration do the following steps:

Procedure

1. In IBM Maximo Asset Performance Management for Energy & Utilities SaaS, click **Administration** > **Standard Operating Procedures**.
2. Click on the approved SOP Definition that you want to view.
3. Click **Definition Actions** and from the drop-down list select **Launch**.

   Once launched, the SOP Definition list will update to indicated the number of Active SOPs based on this definition.
4. Click the approved SOP Definition.
5. Click the **Instances** tab.
6. Click the SOP instance to view the details.

Viewing a Standard Operating Procedure as a user

You can view a Standard Operation Procedure (SOP) as a user.

Before you begin

The Standard Operating Procedure must have been launched before a user can view the instance.

About this task

Users that are not authorized to view the Standard Operation Procedures Definition administration page, can view an SOP instance from My Activities.

Procedure

1. In IBM Maximo Asset Performance Management for Energy & Utilities SaaS, click **Administration**.
2. Click **My Activities**.
3. Click the SOP instance you want to view.

Chapter 4. Using the Custom Data Model tool for the Asset Performance Management application

The Custom Data Model tool is used to add assets to the data model, to upload a data source and to modify an existing asset in the data model.

The Custom Data Model tool includes a template for loading data for the CIM based data model and for customizing the application for IBM Maximo Asset Performance Management for Energy & Utilities SaaS to include extra asset classes.

You can edit the files in the template using a text editor of your choice.

Note: Do not use a spreadsheet to edit these files as unwanted meta data can be introduced to the file.

The template files can be found here: Log into IBM Maximo Asset Performance Management for Energy & Utilities SaaS and go to **Administration** > **Custom Data Model** > **Upload Data Source** > **Download Template**

The Custom Data Model

The custom data model in IBM Maximo Asset Performance Management for Energy & Utilities SaaS is based on the Common Information Model (CIM) data model. The custom data model needs be populated with data through data loading.

The CIM is a standard adopted by the International Electromechanical Commission (IEC) for the reporting and exchange of information about electrical supply networks. Three types of data need be loaded into the IFE Data Model:

- Ontology - used to define the supported assets classes and properties of the asset classes.
- Static data - used to define the nameplate data, the asset identity and geographical location, the network connectivity, the measurement definitions and the relationships between assets.
- Dynamic data - is the data that changes over time, the measurements and events.

Ontology

Ontology is used to define the supported assets classes and properties on asset classes. IBM Maximo Asset Performance Management for Energy & Utilities SaaS integrates data from different data sources that is consolidated into common asset class and properties definitions.

The mapping between external object id and internal object id is also maintained by Maximo APM for E&U SaaS .

Ontology can also be used to map between semantics from all the data sources and a common semantics. Data sources can have their own semantics; they often do not use the same vocabulary of terms and data concepts. For example, a power transformer can be called differently in different data sources. Maximo APM for E&U SaaS maps semantics from all the data sources into the IEC CIM semantics.

There are four types of table in ontology:

- Object ID
- ResourceType
- SubType
- PropertyType

ObjectID table

All objects are identified by a database generated object identifier, and a unique URI that is in two parts: namespace and local name. For example: for the URI `http://cityName#transformer1`, the namespace is `http://cityName#` and its local name is `transformer1`.

When data is loaded to Maximo APM for E&U SaaS, we assume all data objects have an unique id in the format of a URI. Maximo APM for E&U SaaS generates a unique number based the ObjectID for internal use only, for example as primary key or foreign key.

The ObjectID table is used to support this mechanism, providing the mapping between internal object id (oid) and external object URI. This table stores the mapping for Logical Resource, Geo Location, Connectivity Node, Terminal, Measurement, Support Association, Flow Role.

Table 2. ObjectID table

Column name	Type	Description	Sample
iod	BIGNT	Internal object ids which uniquely identifies an object. It is generated by the database.	
namespace	VARCHAR(256)	The namespace part of the URI.	
localName	VARCHAR(256)	The local name part of the URI.	

ResourceType table

The ResourceType table is use to store the metadata of the CIM **`PowerSystemResource`**. For example: The CIM asset classes **`PowerTransformer`**, and **`Switch`**, are subclasses of **`PowerSystemResource`**. The user can register these standard or custom defined power system resources to the `ResourceType` table.

In addition to “PowerSystemResource”, IEC CIM also defines a set of class as asset containers, for example: GeoGraphicalRegion, SubGeographicalRegion, Substation, and Bay. The ResourceType table can also be used to register these standard or any custom defined container types.

Table 3. ResourceType table

Column name	Type	Description	Sample
id	BIGINT	The primary key id of the resource type.	
code	VARCHAR(128)	A unique string based code of the resource type.	
name	VARCHAR(128)	Display name of the resource type.	
description	VARCHAR(512)	Description of the resource type.	

Table 3. ResourceType table (continued)			
Column name	**Type**	**Description**	**Sample**
isAsset	CHAR(1)	Indicates whether resources of this type can be linked to physical asset.	1 = yes 0 = no If `isAsset = 1`, then the table field holds the name of table used to store the properties of the asset.
isResource	CHAR(1)	Indicates whether this resource type is a power system resource or a container.	0 = is a container for example, a geospatial region. 1 = is a power system resource. For example, a transformer.
cimID	VARCHAR(128)	Class name of this resource type as in the IEC CIM when the resource type is defined in IEC CIM.	PowerTransformer
table	VARCHAR(256)	Used when `isAsset = 1`. The fully qualified table name used to store the properties of the asset.	
serviceURL	VARCHAR(256)	The relative service URL which exposes the instance data of this resource type.	

SubType table

The SubType table is used further classify a specific resource type. For example, the resource type, distribution transformer, can be one of several types, two of the types being a pole-mount transformer and a pad-mount transformer.

Table 4. SubType table			
Column name	**Type**	**Description**	**Sample**
resourceType	BIGINT	Primary Key, Foreign Key of the resource type.	
subType	BIGINT	Primary Key of the sub resource type.	
name	VARCHAR(128)	The display name of the sub resource type.	
description	VARCHAR(512)	The description of sub resource type.	

Note: For each existing resource type add a record in the subType table:

- subType = 0
- name = default
- description = default.

For a dynamic protective device, add second subtype:

- subType = 1
- name = Oil circuit breaker (OCB).
- description = Oil circuit breaker

PropertyType table

The `PropertyType` is used to store the properties metadata of each resource type when `ResourceType.isAsset = 1`.

Table 5. PropertyType table

Column name	Type	Description	Sample
id	BIGINT	The primary key, the ID of the property type.	
resourceType	BIGINT	The foreign key, the ID of resource type.	
code	VARCHAR(128)	A unique string based code of the property type.	
name	VARCHAR(128)	Display name of property type.	
description	VARCHAR(512)	Description of property type.	
cimID	VARCHAR(128)	The property name of this property type in the IEC CIM when defined in IEC CIM. For example, `Asset.lifecycle.installationDate`.	
column	VARCHAR(128)	Column name in target table which been used to store this property value.	
dataType	VARCHAR(32)	The data type of the property.	
length	INTEGER	The length of the property.	
scale	INTEGER	The scale of the property.	
isEnum	CHAR(1)	Indicates if the property value is an enumeration.	
isReference	CHAR(1)	Not used.	

Table 5. PropertyType table (continued)			
Column name	**Type**	**Description**	**Sample**
isMinimal	CHAR(1)	Indicates if the property is a minimal property. Minimal property is an important property to filter and sort data.	
tag	VARCHAR(128)	Tags on property type.	
group	VARCHAR(128)	The group name of the property. Used to group related properties.	
unit	VARCHAR(128)	The unit of the property value.	

Logical Resource

The logical resource is abstract of components of an electrical network.

When a transformer is referred to in an electrical network, we are interested in its role in the network, how it connected and which physical device acts upon it. In this example the transformer is the logical resource within the electrical network.

The types of resource varies; the type is defined in the `ResourceType` table. Some types of logical resources are associated with properties that are stored in a separate child table. For example, a logical resource can also have a location that is stored in the `Location` table.

Logical resources have different types that are defined in `ResourceType` table. Based on the definition in the `ResourceType` table, some types of logical resources can be associated with properties that are stored in a separate children table, for example: Pole, SubstationTransformer.

The relationship between logical resources are described by a container and the objects it comprises. The container is used to group together components of an asset, or to group assets of a service region. For example, an overhead cable is composed of line segments, the overhead cable is the container and the line segment are its objects. The aggregation of the relationship can be hierarchical, that is, a container can contain other containers.

Attributes for the Resource table

oid
: BIGINT

  Primary key, the object id of the resource type.

  The Object ID is generated in the ObjectID table.

lastUpdateTime
: TIMESTAMP

  Generated at the time of last update.

resourceType
: BIGINT

  Foreign key, the type of this logical resource with reference to the ResourceType table.

subType
: BIGINT

  The sub resource type with reference to the `subType` table.

mRID
: VARCHAR(256)

Master resource ID. The string based identifier of the logical resource. A resource is a role in the network.

name
VARCHAR(128)

The name of logical resource.

description
VARCHAR(512)

The description of the logical resource.

isContainer
CHAR(1)

Indicates if the logical resources is a container of other components.

location
BIGINT

Foreign key, the location of the logical resource with reference to the Location table.

container
BIGINT

Foreign key, the container or parent of this logical resource, with reference to self.

feeder
BIGNT

The oid of the feeder in the resource table.

substation
BIGNT

The oid of the substation region in the resource table.

phase
INTEGER

The bitwise code for the phase, phase A = 001, phase B = 010, phase C = 100, phase AB = 011, phase ABC = 111. For example: 010 represents phase B where 010 is the binary value and the database integer value is 2.

Asset identity, attributes and templates

The asset identity and attributes describe physical asset of a logical resource at a point in time.

Each logical resource can be associated with more than one physical asset. Each asset described indicates which physical asset is acting as this logical role over different time periods. For example the logical resource `transformer1` is associated with two physical transformers, one was active from the year 2000 to 2010 and the other has been active from the year 2010 to the present day.

The separation of the logical resource from the physical asset has several benefits.

- You are able to track the replacement history of a logical resource.
- The role and responsibility is made clear: the role of the logical resource is to describe the measurement, the connections and location of an asset, and the relationship between assets. The physical asset focuses on the identity of the asset and the static attributes of the nameplate data.

Static attributes are the asset specific nameplate information including asset type, model, date of manufacture, manufacturer specifications, serial number, and default operating state. This information typically resides in an Enterprise Asset Management (EAM) system, an Asset Management System (AMS), or in a Geographic Information System (GIS) for assets in the distribution grid. The AMS captures asset attributes beyond what is available in the GIS system.

For each resource type registered, when isAsset = 1 in the ResourceType table, there is a corresponding asset identity table used to store the attributes of the asset. Some attributes are also registered in the PropertyType table.

Maximo APM for E&U SaaS provides 10 asset identity and attribute tables:

- Pole
- SubstationTransformer
- DistributionTransformer
- InstrumentTransformer
- OverheadCable
- UnderGroundCable
- CircuitBreaker
- Tower
- Battery
- Switchgear

You can file the asset identify and attributes in template files in "Downloading the data source template" on page 73.

Geometry

The geospatial information that specifies both the geometry and the location of a logical resource in a network.

Geospatial location is a key attribute in a physical infrastructure that enables geospatial analytics and visualization of the entities on a map.

Location attributes

The location attributes are typically stored in a GIS system, an example being the Environmental Systems Research Institute (ESRI) ArcGIS.

oid
BIGINT

Primary key, object id of the location.

Object id is generated in ObjectID table.

lastUpdateTime
TIMESTAMP

Generated last update time.

mRID
VARCHAR(256)

Master resource id. Which is a string based identifier.

isActive
CHAR(1)

Indicate if the location is active or not.

mainAddress
VARCHAR(256)

Main address of the location.

phone1
VARCHAR(32)

Phone number.

phone2
VARCHAR(32)

Phone number.

secondaryAddress
VARCHAR(256)

Secondary address of the location. For example, PO Box address may have different ZIP code than that in the 'mainAddress'.

geometry
GEOMETRY

The geospatial location in GIS.

locationDescription
VARCHAR(256)

Description of the location.

Relationships and connectivity

The relationships that include network connectivity, the supporting structure, and flow roles are described here.

The IEC CIM is the template for model network connectivity between logical resources. In IEC CIM, ConnectiivtyNode and Terminal are used to model connectivity between conducting equipment.

- ConnectivityNode: points where terminals of conducting equipment are connected together with zero impedance.
- Terminal: An electrical connection point to a piece of electrical equipment. Terminals are connected at physical connection points called "connectivity nodes".

The same model is used to describe the connectivity between logical resources.

Attributes for the Connectivity Node

oid
BIGINT

Primary key, object id of the connectivity node.

Object id is generated in ObjectID table.

lastUpdateTime
TIMESTAMP

Generated last update time.

mRID
VARCHAR(256)

The master resource ID. This is a string based identifier for the connectivity node.

isActive
CHAR(1)

Indicates if current connectivity node is active.

name
VARCHAR(128)

The name of the connectivity node.

description
VARCHAR(512)

The description of the connectivity node.

container
BIGINT

A foreign key, the parent of this connectivity node with reference to the resource table.

Attributes for the terminals

oid
BIGINT

Primary key, object id of the terminal.

Object id is generated in ObjectID table.

lastUpdateTime
TIMESTAMP

Generated last update time.

mRID
VARCHAR(256)

The master resource ID. This is a string based identifier for the connectivity node.

isActive
CHAR(1)

Indicates if the current terminal is active.

name
VARCHAR(128)

The name of the terminal.

description
VARCHAR(512)

The description of terminal.

sequenceNumber
INTEGER

The sequence number of the terminal. If a resource has multiple terminals, the sequence number is used to put the terminals in order.

connected
CHAR(1)

Indicates if the terminal is connected.

phases
VARCHAR(32)

The phases of the terminal.

resource
BIGINT

A foreign Key, the resource end of the terminal, with reference to the Resource attributes.

connectivityNode
BIGINT

A foreign key, the connectivity node end of the terminal, with reference to the ConnectivityNode attributes.

The attributes for support association

For supporting structural relationships, the supporting structure relationship does not indicate any electric transmission between assets. For example, if several poles are used support the an overhead cable, then there is a supporting structure relationship between the poles and the overhead cable, however, this relationship cannot be modeled using the connectivity relationship as no electric current flows from the cable to the poles.

oid
BIGINT

The primary key, the object id of the support association.

The object id is generated in the ObjectID table.

lastUpdateTime
TIMESTAMP

Generated last update time.

mRID
VARCHAR(256)

The master resource id. This is a string based identifier for the support association.

isActive
CHAR(1)

Indicates if current support association is active.

supporter
BIGINT

A foreign key, the oid for the supporter resource, with reference to the Resource attributes.

supported
BIGINT

A foreign key, the oid of the supported resource, with reference to the resource attributes. For example if the supporter is a pole then the oid is for the pole, the supported is the oid of the cable.

The attributes for flow role

Flow role is used to describe the role of a logical resource in a network. For example, we can assign the role power source to one asset, and assign the role power sink of to another asset, we can then use these roles to calculate power flow.

oid
BIGINT

Primary key, foreign key, object id of the resource.

feeder
BIGINT

Primary key, foreign key, object id of the feeder.

lastUpdateTime
TIMESTAMP

Generated last update time.

isActive
CHAR(1)

Indicates if the current flow role is active.

role
INTEGER

The role of the logical resource in the network.

Measurement and event reading data

Measurement and event data is time-series data that has regular time intervals; it describes the status of the assets.

There are different types of measurements and events for different resources. For example, a transformer can have measurements for oil temperature and load, a pole can have a measurement for inspection.

Measurement and event data can come from a SCADA system (Supervisory Control And Data Acquisition) that connects to sensors to collect data.

Inspection records can also be used to collect measurement and event data from an EAM (Enterprise Asset Management) system like Maximo.

IBM Watson IoT Platform use two types of table to model the measurement and event reading data:

- The Measurement table describes the measurement metadata, including the measurement that is related to the logical resource or terminal.
- The value tables stores the time-series value from a specific measurement. There are multiple value tables used to store different types of value. For example, AnalogValue, DigitalValue. You can also define a custom value table to store more complex attributes like pole inspection for time-series value.

Measurement attributes

oid
: BIGINT

  Primary key, object id of the measurement. The object ID is generated in ObjectID table.

lastUpdateTime
: TIMESTAMP

  Generated last update time.

mRID
: VARCHAR(256)

  Master resource id. Which is a string based identifier of measurement.

isActive
: CHAR(1)

  Indicates if current measurement is active.

name
: VARCHAR(128)

  The name of measurement.

measurementType
: VARCHAR(128)

  The type of measurement.

phaseCode
: VARCHAR(32)

  The phase code of measurement

unitMultiplier
: VARCHAR(32)

  The multiplier of the unit. For example “k” indicates one thousand, if the symbol is V, then the fully qualified unit is kV.

unitSymbol
: VARCHAR(32)

  The symbol of the unit. For example, V indicates voltage.

resource
: BIGINT

  Foreign key, indicates with the logical resource the measurement being measured.

terminal
: BIGINT

  Foreign key, indicates with the terminal of the specified logical resource the measurement being measured.

Common attributes for the value tables

IBM Watson IoT Platform provides several value tables.

- DigitalValue: numeric value.
- AnalogValue: text value with status.

- DGAValue: values for dissolved gas analysis, each column represents the percentage of a gas.
- PoleInspection: values for the pole inspection.
- Condition: values for the circuit breaker.

Columns of the value tables are different for each type of value. Some common columns list here:

measurement
BIGINT

Primary key, foreign key, object id of the measurement.

timestamp
TIMESTAMP

Primary key, the value of the timestamp.

DigitalValue table

measurement
BIGINT

Primary key, foreign key, object id of the measurement.

timestamp
TIMESTAMP

Primary key, the value of the timestamp.

value
VARCHAR(128)

The value of the current reading.

status
VARCHAR(128)

The status of the current reading.

AnalogValue Table

measurement
BIGINT

Primary key, foreign key, object id of the measurement.

timestamp
TIMESTAMP

Primary key, the value of the timestamp.

value
REAL

The value of the current reading.

DGAValue Table

measurement
BIGINT

Primary key, foreign key, object id of the measurement.

timestamp
TIMESTAMP

Primary key, the value of the timestamp.

CH4
REAL

CH4 percentage

C2H4
: REAL

C2H4 percentage

C2H2
: REAL

C2H2 percentage.

PoleInspection table

measurement
: BIGINT

Primary key, foreign key, object id of the measurement.

timestamp
: TIMESTAMP

Primary key, the value of the timestamp.

serialNumber
: VARCHAR(128)

Serial number of pole be inspected

inspectionId
: VARCHAR(32)

The inspection ID.

rejectStatus
: VARCHAR(32)

The reject status.

rejectStatusDescription
: VARCHAR(128)

The description of the reject status.

primaryRejectReason
: VARCHAR(128)

The primary cause for the reject status.

enclosedPockets
: INT

The number of enclosed pockets.

exposedPockets
: INT

The number of exposed pockets.

trimmerDamage
: CHAR(1)

Presents of trimmer damage.

impactDamage
: CHAR(1)

Presents of impact damage.

internalDecay
: CHAR(1)

Internal decay detected.

heartRot
: CHAR(1)

Heart rot detected.

maxExternalDecayDepth
DECIMAL(8 , 4)

The maximum external depth of decay.

minInternalShellThickness
DECIMAL(8 , 4)

The minimum internal shell thickness.

percentPoleStrength
DECIMAL(8 , 4)

The percentage of pole strength.

flurodsInstalled
SMALLINT

The number of flurods installed.

poleStatus
VARCHAR(32)

The status of the pole.

poleReinforcable
CHAR(1)

Denotes if the pole can be reinforced.

lastInspectedYear
VARCHAR(32)

The year of the last inspection.

lastInspectedBy
VARCHAR(32)

The name of the person who carried out the last inspection.

Condition table

measurement
BIGINT

Primary key, foreign key, object id of the measurement.

timestamp
TIMESTAMP

Primary key, the value of the timestamp.

rating
CHAR

Rating in A, B, C or D.

Defining a new asset class in the data model

You want to integrate a new asset class into IBM Maximo Asset Performance Management for Energy & Utilities SaaS and the assets belonging to the asset class.

Assets belonging to this asset class need to be integrated into current system. These are the steps required to integrate the new asset class into Maximo APM for E&U SaaS .

The custom data model aims to help operators create and manage the asset classes and the measurement reading table.

Maximo APM for E&U SaaS integrates data from different data sources, for example EAM System, GIS System. Data from different sources are consolidated into a common asset class and properties definitions. The mapping between an external object ID and an internal object ID is also maintained by Maximo APM for E&U SaaS.

Adding an asset class

The resource type is the category of the asset. In this example the resource type is *Meter*. When you have created the resource type you can add the instruments to this resource type.

About this task

To define a new asset class type, you need to add the asset class to the resource type table in the data base for IBM Maximo APM for Energy & Utilities.

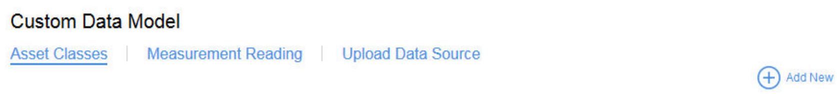


Figure 9. Add an asset class

Procedure

1. Open IBM Maximo Asset Performance Management for Energy & Utilities SaaS and select **Custom data model** > **Asset Classes** from the drop-down menu.
2. On the **Asset Classes** tab, click **Add New** to open the attributes dialog box for the asset class.
    a) Type the code. The code is a unique alpha-numeric code for the asset class. The letters a to Z and numbers 0 to 9 are supported.
3. Type the name of the new asset class.

   In the example `Meter` is the name of the asset class and is a globalization attribution. Click the edit icon and you will see the dialog. You can input one or more languages. Click the **Save** button to save the name.
4. Type the table name. The table name is in the form `<schema name>.<table name>`, for example `CIM.METER`.
5. Type the description for the asset class.
6. Select the asset type. Each asset type has a setting for the `isAsset` and `isResource` attributes.
    - Container,
        - `isAsset = 0`
        - `isResource = 0`
    - Logical resource
        - `isAsset = 0`
        - `isResource = 1`
    - Physical asset
        - `isAsset = 1`
        - `isResource = 1`
7. Click **Save**.

   The asset type and asset table are created with four default attributes for each asset type: `serial number`, `is Active`, `Installation data`, `remove date`.

Defining the property type of an asset class

You must add the property type to the new asset class in IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

You can edit an asset class when you click the edit button in a row.

Figure 10. Row details of an asset class.

Procedure

1. Click the **Add New** icon in the row for the asset class that you want to add a property type to.
   a) Type the **Code**. The code is a unique alpha-numeric code for the asset type property. The letters a to Z and numbers 0 to 9 are supported.
   b) Type the name of the property type in the **Column Name** field. The column name is the name of the property using the letters a to Z and numbers 0 to 9 without spaces. For example if the **Name** of a property is `Serial Number`, then the **Column Name** is `SERIALNUMBER`.
   c) Type the length of the string **Column Name** in the **Column Length** field.
   d) Type the scale of the asset in the **Column scale** field.
   e) Type `No` in the **Allow null** field if you do not want the column to allow null values and you need a default value. If you do not need a default value, then type `Yes`.
2. Click **Save**.

Defining the measurement type for an asset class

You can add a measurement type to an asset class.

About this task

This procedure adds a measurement type to the .csv reading file located here: /opt/IBM/energy/data/<your_directory_name>/reading. The columns 1 and 2 are generated with default names, **measurement** and **timestamp**, you are creating additional measurement types from the names already created in the reading .csv file.

Procedure

1. In the list of **Asset Classes** select the row for the asset class you want to edit and click the **Edit** icon.
2. Click the **Measurement** tab.
3. In the **Measurement Type** field type the measurement description. The letter a to Z and the number 0 to 9 are supported.
4. Select the target table.

   The target table is already defined in the data base. If not you can create a new measurement reading table.
5. Click the **Add New** icon.
6. Type the **Code**. The code is a column name in the .csv reading file. The code is a unique alpha-numeric code for the measurement type property. The letters a to Z and numbers 0 to 9 are supported.
7. Select the **Type**. The type is the column type in the .csv file.
8. Select the **Target Column**. This is the column in the database to where you want to import the .csv file.
9. Click **Save** > **save & back** to exit.

Preparing the data for loading

When you have asset classes defined, you can prepare the data files, define the assets, and load the data files to the database.

The data files are .csv files that are can be edited in a text editor of your choice.

Important: Do not use a spreadsheet to create or edit these .csv files. A spreadsheet can introduce formats that stops the data load.

Defining the reading data

The measurement reading table is used to store the data from measurements made on an asset class.

About this task

The tab Measurement Reading is where you define the table name and description of the measurement. The measurement reading is used when defining the measurement type for an asset class.

Procedure

1. In IBM Maximo Asset Performance Management for Energy & Utilities SaaS, select **Administration** > **Custom Data Model**.
2. Select the **Measurement Reading** tab, and click the **Add New** icon.
3. Add the **Table Name**.

   The table name must contain an existing schema name and a table name, for example **cim.testmeasurement**.
4. Add the **Description** of the measurement table.
5. Click **Save**.
6. Click the edit button to edit the newly created measurement reading table. You can add or delete table columns.
7. Click **save & back** to exit.

Defining the resource data

The resource data is saved in a .csv file that you need to create and add content to.

Procedure

1. Go to `/opt/IBM/energy/data/<your_directory_name>/resources` on the application server, and create a file called `resource_<asset_class>.csv`.

   Note: The directory `<your_directory_name>` is created by you and has the name that you require.

   Note: Where `<asset_class>` is the asset class that you are to load the data for. For example the file name for the asset class *pole* is `resource_pole.csv`.
2. Add the following columns names to the file as a single row:

   ```
   mRID,isActive,resourceType,subType,name,
   description,isContainer,container,substation,feeder,phase,location,geometry
   ```

   The data in the `resource_<asset_class>.csv` file is loaded to the ObjectID, Resource and Location tables. The columns in the file include:

   Table 6. The columns in the .csv file

Column name	Type	Nullable	Description
mRID	VARCHAR(255)	N	Master resource ID. A string based identifier of the logical resource.
isActive	CHAR(1)	N	Indicates if the asset is currently active, 1 = active, 0 = inactive.
resourceType	VARCHAR(128)	N	Code of the resource type defined in the ResourceType table.

Table 6. The columns in the .csv file (continued)			
Column name	**Type**	**Nullable**	**Description**
subType	BIGINT(8)	Y	The subtype of the resource.
name	VARCHAR(128)	N	The name of the logical resource.
description	VARCHAR(512)	Y	The description of the logical resource.
isContainer	CHAR(1)	N	Indicates if the logical resources is a container (has children), 0 not container, or 1 is container.
container	VARCHAR(255)	Y	Foreign key, the parent of this logical resource, reference to self.
substation	VARCHAR(255)	Y	The substation of the resource.
feeder	VARCHAR(255)	Y	The feeder of the resource.
phase	INTEGER(4)	Y	The phase of the resource.
location	VARCHAR(256)	Y	Foreign key, location of the logical resource, reference to Location.
geometry	GEOMETRY	Y	The geospatial location in GIS.

3. Here is a sample data that is added to the `resource_pole.csv` file:

```
mRID,isActive,resourceType,subType,name,description,
isContainer,container,substation,feeder,phase,location,
geometry
"Pole_271759",1,"Pole",3,"Pole_271759","Pole 271759",
0,"Substation_LILY","Substation_LILY","","","Loc_Pole_271759",
"POINT (-83.375137568 42.572365742)"
```

Defining the measurement data

The measurement data is saved in a `.csv` file that you need to create and add content to.

Procedure

1. Go to `/opt/IBM/energy/data/<your_directory_name>/measurements` on the application server, and create a file named `measurement_<asset_class>.csv`.

   Note: The directory `<your_directory_name>` is created by you and has the name that you require.

   Note: Where `<asset_class>` is the asset class that you are to load the data for. For example the file name for the asset class *pole* is `maintenance_pole.csv`.

2. Add the following content to the file:

```
<mRID>,<isActive>,<name>,<measurementType>,<phaseCode>,<unitMultiplier>,
<unitSymbol>,<resource>,<terminal>
```

The measurement CSV file is loaded to the ObjectID and Measurement table. The columns in the CSV file include:

Table 7. Columns in the measurement table		
Field	**Type**	**Description**
mRID	VARCHAR(256)	Master resource ID. a string based identifier of the measurement.
isActive	CHAR(1)	Indicates if current measurement is active.
name	VARCHAR(128)	The name of the measurement.
measurementType	VARCHAR(128)	The type of measurement.
phaseCode	VARCHAR(32)	The phase code of the measurement.
unitMultiplier	VARCHAR(32)	Multiplier of unit. For example: k indicates one thousand, if the symbol is V, then the fully qualified unit is kV.
unitSymbol	VARCHAR(32)	The symbol of the unit. For example: V indicates voltage.
resource	VARCHAR(256)	The mRid of the logical resource that the measurement is measuring.
terminal	VARCHAR(256)	The mRid of the terminal that the measurement is measuring.

As the data of meter measurement indicates the power payload, the sample csv section needs to be added to `meter.csv` file.

```
mRID,isActive,name,measurementType,phaseCode,unitMultiplier,
unitSymbol,resource,terminal
"M_MT_148610_Payload","1","PayLoad","PayLoad",,"none","none","Meter_1813699",
```

3. Save and close the file.

Defining the connectivity node data

The connectivity node data is saved in a `.csv` file that you need to create and add content to.

Procedure

1. Go to `/opt/IBM/energy/data/<your_directory_name>/connectivityNodes` on the application server, and create file called `ConnectivityNode.csv`.
2. Add the following content to the file:

```
<isActive>,<mRID>,<name>,<description>,<container>
```

The `ConnectivityNode.csv` file is loaded to the ObjectID, Resource and Location tables. The columns in the CSV file include:

Table 8. The columns in the .csv file		
Field	**Type**	**Description**
isActive	CHAR(1)	Indicates if the asset is currently active, 1 = active, 0 = inactive.
mRID	VARCHAR(256)	Master resource id. A string based identifier of the connectivity node.
name	VARCHAR(128)	The name of the connectivity node.
description	VARCHAR(512)	The description of the connectivity node.
container	VARCHAR(256)	The Master resource id of the container of the connectivity node.

As the meter asset is connected to distribution transformer here is a sample csv section that needs to be added to `ConnectivityNode.csv`. In the terminal data file, it defines Meter_1813699 is connected to distribution transformer DT_101221, through 2 terminals and 1 connectivity node, for example: this sample defines the detail of the connectivity node.

```
isActive,mRID,name,description,container
"1","CN_cn85555","cn85555","cn5555","Substation_NIXON"
```

Defining the terminal data

The terminal data is saved in file called `Terminal.csv` that you need to create and add content to.

Procedure

1. Go to `/opt/IBM/energy/data/<your_directory_name>/terminals` on the application server, and create a file named `terminal.csv`.

   Note: The directory `<your_directory_name>` is created by you and has the name that you require.
2. Add the following content to the file:

   ```
   mRID,isActive,name,description,sequenceNumber,connected,phases,resource,connectivityNode
   ```

   The data in the `terminal.csv` file is loaded to the ObjectID, Resource and Location tables. The columns in the .csv file include:

Table 9. The columns in the .csv file		
Field	**Type**	**Description**
mRID	VARCHAR(256)	Master resource id. A string based identifier of the terminal.
isActive	CHAR(1)	Indicates if current terminal is active.
name	VARCHAR(128)	The name of the terminal.
description	VARCHAR(512)	Description of terminal.

Table 9. The columns in the .csv file (continued)

Field	Type	Description
sequenceNumber	INTEGER	The sequence number of terminal. If a resource has multiple terminals, the sequence number is used to order the terminals in sequence.
connected	CHAR(1)	Indicates if the terminal is connected.
phases	VARCHAR(32)	The phases of the terminal.
resource	VARCHAR(256)	The mRID of the resource that the terminal is connected to.
connectivityNode	VARCHAR(256)	The mRID of the connectivity node that the terminal is connected to.

The meter asset is connected to distribution transformer through terminals and connectivity nodes. If the meter Meter_1813699 is connected to the distribution transformer DT_101221, the terminal of Meter_1813699 is Terminal_Meter_1813699_1 and the terminal of DT_101221 is Terminal_DT_101221_2. The two terminals are both connected to connectivity node CN_cn85555. Here is a diagram for the connection information:

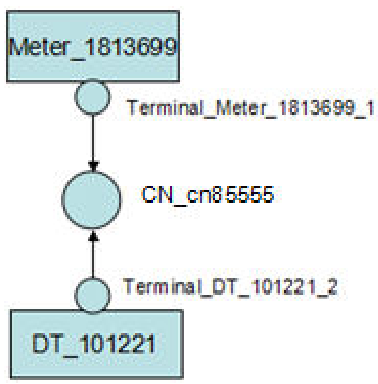


Figure 11. A diagram that shows the connection information

3. Here is a sample `Terminal.csv` file.

```
mRID,isActive,name,description,sequenceNumber,connected,phases,resource,connectivityNode
"Terminal_Meter_1813699_1","1","Terminal 1","Terminal 1","1","1",,"Meter_1813699",
"CN_cn85555""1",
"Terminal_DT_101221_2","Terminal 2","Terminal 2","2","1",,"DT_101221","CN_cn85555"
```

Defining the support association

The support association is saved in the file `support.csv` file that you need to create to add data to.

About this task

In comparison to the network connectivity relationship, a support association does not indicate any electric transmission between the assets. The support association data describes the physical relationship between poles and overhead cables. A pole is the supporter of a cable, the cable is supported by a pole.

Procedure

1. Go to `/opt/IBM/energy/data/<your_directory_name>/supportAssociation` on the application server, and create a file named `support.csv`.

   Note: The directory `<your_directory_name>` is created by you and has the name that you require.
2. Add the following content to the file:

   ```
   isActive,mRID,supporter,supported
   ```

   The data in the `support.csv` file is loaded to the ObjectID, Resource and Location tables. The columns in the .csv file include:

   Table 10. The columns in the .csv file

Field	Type	Description
isActive	CHAR(1)	Delete tag, indicates if current terminal is active.
mRID	VARCHAR(256)	Master resource id. A string based identifier of the support association between the supporter and supported.
supporter	BIGINT	Foreign key, the oid of the supporter resource, reference to the Resource table.
supported	BIGINT	Foreign key, the oid of the supported resource, reference to the Resource table.

   The pole assets with the mRID Pole_398166 and Pole_398167 support the overhead cable with the mRID OHC_1142035 Here is a diagram for the connection information:

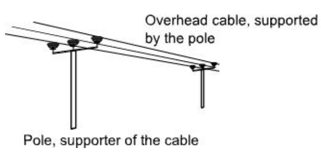
   

   Figure 12. A diagram showing the support association between pole and cable

Defining the flow role for an asset

The flow role describes the role of a logical resource in the network, for example, you can assign a power source role to an asset, and assign power sink role to another asset. These roles can be used to calculate power flow.

About this task

The flow role data is saved in file called `flowRole.csv` that you need to create and add content to.

Procedure

1. Go to `/opt/IBM/energy/data/<your_directory_name>/flowRole` on the application server, and create a file named `flowRole.csv`.

   Note: The directory `<your_directory_name>` is created by you and has the name that you require.
2. Add the following content to the file:

   ```
   isActive,role,feeder,mRID
   ```

The data in the `flowRole.csv` file is loaded to the ObjectID, Resource and Location tables. The columns in the .csv file include:

Table 11. The columns in the .csv file

Field	**Type**	**Description**
isActive	CHAR(1)	Indicates if current terminal is active.
role	INTEGER	The role of logical resource in the network.
feeder	BIGINT	Primary key, foreign key, the object id of the feeder.
mRID	VARCHAR(256)	Master resource id. A string based identifier of the terminal.

3. Here is a sample `roleFlow.csv` file.

```
isActive,role,feeder,mRID
1,1,"COMLK","brk_493981"
1,1,"COMLK","brk_493980"
1,1,"COMLK","brk_493979"
1,0,"COMLK","ST_1400502"
```

4. Save and close the file.

Uploading of bulk data

IBM Maximo Asset Performance Management for Energy & Utilities SaaS has two methods to load data. This method is for the bulk uploading of data for a utility.

The uploading of bulk data consists of configuring the data loader and running the data loader.

Configuring the data loader

The data loader needs to be configured and the `config.properties` file created.

About this task

Create the contents for the `config.properties` file.

Procedure

1. Login to the App node and then su to root.

```

```

2. Save the data folder <*your_directory_name*> on the App node, for example: save it on `/tmp/<your_directory_name>`
3. If you have a Docker install, run these commands:

```
docker cp /tmp/<your_directory_name> IFEAppNode:/tmp
```

```
docker exec -it IFEAppNode bash
```

The following steps are run in the docker container.

4. Move the folder to the `/opt/IBM/energy/data` directory with the command:

```
mv /tmp/<your_directory_name> /opt/IBM/energy/data
```

5. Change the owner of the folder to the user of the liberty server if it is not root, the user is wlp for a docker environment.

```
chown -R <liberty_user>:<liberty_user>
/opt/IBM/energy/data/<your_directory_name>
```

For example:

```
chown -R wlp:wlp /opt/IBM/energy/data/<your_directory_name>
```

6. Change the user to <*liberty_user*> to run the following steps:

```
su - <liberty_user>
```

For example:

```
su - wlp
```

7. Go to `/opt/IBM/energy/data/<your_directory_name>` and create a file called `config.properties`.
8. Add content to the file for the `resources/meter.csv` to the `adapters.1.file` section.

For example:

```
#namespace of the data
namespace=http://<your_data_name>#
#number of rows of single insert/update
batch=1000
#number of threadsconcurrency=10
#service url
serviceBaseURL=https://<app_server_host_name>:<port_number>
serviceUser=<user in EnergyAdmin group>
servicePassword=<password>

#adapter and load sequence
#logical resources
adapters.1.name=com.ibm.ife.ah.dataloader.Resource
adapters.1.file=resources/substation.csv,resources/meter.csv

adapters.2.name=com.ibm.ife.ah.dataloader.ConnectivityNode
adapters.2.file=connectivityNodes/meter.csv

adapters.3.name=com.ibm.ife.ah.dataloader.Terminal
adapters.3.file=terminals/meter.csv

adapters.4.name=com.ibm.ife.ah.dataloader.Measurement
adapters.4.file=measurements/meter.csv

#physical assets
Adapters.5.name=com.ibm.ife.ah.dataloader.Asset
Adapters.5.file=assets/meter.csv
Adapters.5.type=Meter
```

Where:

- The resource `resources/meter.csv` is in the `adapters.1.file` section.
- The connectivity node `connectivityNodes/meter.csv` is in the `adapters.2.file` section.
- The terminal `terminals/meter.csv` is in the `adapters.3.`name section.
- The measurement `measurements/meter.csv` is in the `adapters.4.file` section.
- The physical asset `assets/meter.csv` is in the `adapters.5.file` section.

9. Make sure that the service base URL is HTTPS.
10. Save the file `config.properties`.

Running the data loader

User can use the data loader tool to validate the data in the csv files and load the data into database.

Procedure

1. Login to the App node with root user.
2. If you have a Docker environment, run this command:

```
docker exec -it IFEAppNode bash
```

The following steps are run in the docker container.

3. Change the user to the one who starts the liberty server if <liberty_user> is not root. The <liberty_user> is wlp for a docker environment.

   ```
   su - <liberty_user>
   ```

   For example:

   ```
   su - wlp
   ```

4. Find the data loader tool in the directory:

   `/opt/IBM/energy/data/runDataLoader.sh`

5. Run the data loader tool to validate data loader with the following command:

   ```
   ./runDataLoader.sh validate <your_directory_name>
   ```

   <your_directory_name> is the folder name you put your data. It is in `/opt/IBM/energy/data/<your_directory_name>`

6. When the data loader passes validation, run the data loader tool with the following command:

   ```
   ./runDataLoader.sh <your_directory_name>
   ```

Uploading of data for the development of the application

You can use the user interface to upload small data sets for development use.

You must have the data in the same format as the template provided by IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

Downloading the data source template

You can use the template to configure your data for uploading to IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

IBM Maximo Asset Performance Management for Energy & Utilities SaaS provides the template for you to use.

Procedure

1. From the menu bar, click **Administration** > **Custom Data Model** .
2. In Custom Data Model click **Upload Data Source** > **Download Template**.
3. Save the template `utilityname.zip` for you to use.

   You can change the name of the file to a relevant name for your use. The file to upload back to Maximo APM for E&U SaaS must be a zip file.

Uploading development data via the user interface

You can use the user interface to upload small data sets to upload data to Maximo APM for E&U SaaS.

About this task

You must have the data in the same format as the template provided by the file `utilityname.zip`. The file name can be your own choosing.

Note: The user interface does not provide a validation step; the data is loaded directly. You can view the details in the log file provided after upload.

Procedure

1. From the menu bar click **Administration** > **Custom Data Model**
2. From **Custom Data Model** click **Upload Data Source** > **Upload Data Source**.
3. You can either drag and drop the zip file to the Upload Data Source window, or browse to the location where you have saved the zip file.

   The **Custom Data Model** window shows the status of the upload, and a log file is avaiable in the **View Log** column.
4. Click the icon in the **View Log** column to view the log file.

Integration with IBM Cognos Analytics

You can integration IBM Maximo Asset Performance Management for Energy & Utilities SaaS with IBM Cognos Analytics where you can develop you own Cognos Analytics reports, and add the report to Maximo APM for E&U SaaS as a page.

Developing a report within Cognos Analytics

The procedure makes use of the data model that you can leverage to develop a Cognos Analytics report.

Procedure

1. Open the link to Cognos Analytics: `https://<node of the ip where the IFEIHSNode is deployed on>/bi`
2. In **Team Content**, you can find where the data model "ahe" has been deployed.

   Your organization's content is in the Team content folder.

   This is where you find reports, packages, dashboards, stories, models, and more. Items in Team content are organized in folders, so you can search with keywords.

   You can find the information for getting started in Cognos Analytics here: https://www.ibm.com/support/knowledgecenter/en/SSEP7J_11.0.0/com.ibm.swg.ba.cognos.wig_cr.doc/c_gtstd_ica_overview.html

Obtaining the relative URI for the Cognos Analytics report

You can find the relative URI and add and change the parameters associated with the URI.

Procedure

1. Open the link to Cognos Analytics: `https://<node of the ip where the IFEIHSNode is deployed on>/bi`
2. Click **Team Content** > **Report**.

   You can see the reports listed, for example: **ah-cognos-report**
3. Right click the report and click **Embedded**.
4. Save the URI into a text editor.
5. Remove the `https://<ip>` part of the URI.

   Note: Only the relative part of the URI is required.

   For example: `bi/?perspective=authoring&pathRef=.public_folders%2Freport%2Fah-cognos-report&ui_appbar=false&ui_navbar=false&action=run&format=HTML&prompt=true`
6. You can add parameters to the relative URI if you do not want the URI content to be editable.

   For example: for the "ah-cognos-report" sample report that includes a history and a year filter, you can directly pass them by setting the prompt as false and adding `p_history_id` and `p_year` parameters in the URI: `/bi/?perspective=authoring&pathRef=.public_folders%2Freport%2Fah-cognos-`

```
report&ui_appbar=false&ui_navbar=false&action=run&format=HTML&prompt=false&p
_history_id=1521884434&p_year=2018
```

Note: You can obtain the `p_history_id` by checking the AH.HISTORY table or by using the current analysis details that you can find in xhr request in the application.

Adding a new page and page hierarchy to IBM Maximo Asset Performance Management for Energy & Utilities SaaS

After you have obtained the URI relative link, you can add the new page to IBM Maximo Asset Performance Management for Energy & Utilities SaaS

Procedure

1. Enter the URL into the address field of the browser.

   Note: The fully qualified domain name is required in the URL, for example, `https://web_hostname/wps/myportal` where *web_hostname* is the host name of the web server. If you use the IP address instead of the registered fully qualified domain name, some windows do not open correctly. Also, if you do not use the https protocol, the link is redirected to use the https protocol.
2. On the login page, enter your user ID and password.
3. Click **Log In**.
4. Click the menu icon and click **Administration** > **Administration Console**.
5. Click **Page** > **Create**
6. Type the title of the page and description in the **Create a Page** dialog box.

   For example **Cognos Visualization**.
7. Paste the URI that you have obtained in the **URI** field.
8. Click **Save**.
9. Click Page Hierarchy and hover over the page where you want the page to show in the hierarchy listing.

   For example: **Asset Performance Management** > **Transmission**.
10. Click the **+** icon to add one more page in this hierarchy.
11. Specify the name, for example **Cognos Visualization**, the sequence, and choose the new page you created in **Page Definition**.
12. Click **Save**.
13. Refresh the page, and you can see your new added page under **Asset Performance Management (Transmission)**
14. Click the navigation and find your new added page, click the new page to check the report content.

Chapter 5. Customization of IBM Maximo Asset Performance Management for Energy & Utilities SaaS

You can develop and extend the applications that are part of IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

Customizing the Asset Performance Management application

You can customize the solution to suit your business requirements.

Use the Administration Console view to register custom user interface components and to customize the user interface to suit your particular operation.

Customizing the user interface

You can use the extension capabilities that are provided by the user interface framework to build application user interfaces that meet your operational requirements.

IBM Maximo Asset Performance Management for Energy & Utilities SaaS provides a number of reusable user interface components including pages, styles, layouts, and widgets. You can also add custom widgets and layouts to your solution by using the user interface extension framework. Use the Administration Console view to register new user interface components, to add new pages, and to configure the style, layout, widgets, access controls, and Representational State Transfer (REST) services for the pages in your solution.

Configuring pages

If you are an administrator, you can add and configure custom pages. Each page has a style and a layout to manage the widgets on the page.

About this task

Use the **Administration** view to add and configure custom pages. Each page is defined by a set of properties, and each property value that you enter is validated by the solution upon entry. You can view pages that are provided with the solution, but you cannot delete them, and you can edit only the layout properties for these pages.

Procedure

1. In the **Administration** view, click **Page**.

Adding a custom page

2. Click **Create**.
3. In the **Create a Page** window, enter the details for your custom page.
   a) In **Title**, enter a unique page title.
   b) Optional: In **Description**, enter a description of the page.
   c) In **URI**, enter a valid URI, for example `/ibm/ife/sample/index/HTML`.
   d) Click Save. The new page is listed under the **Create** button.
   e) For a newly created page, select **Groups** in the **Access control** field if you want those groups to access this page. Click **Save**.

Editing information for a custom page

4. Edit the page title, description, URI and access control for a custom page.
5. Click **Save** to save the changes to the page's configuration.

Deleting a custom page

6. Click the delete icon beside the custom page, and then in the confirmation window, click **Yes**.

Note: You cannot delete a custom page if the page is included in the configuration of a page hierarchy.

Configuring page hierarchies

If you are an administrator, you can add and remove page hierarchies, and you can configure the contents of page hierarchies. Page hierarchies that contain one or more pages are displayed in the main navigation bar.

About this task

Use the **Administration** view to add, edit, and remove page hierarchies. Each hierarchy has a label and can contain one or more pages or page hierarchies. Each element of a page hierarchy is defined by a set of properties, and each property value that you enter is validated by the solution upon entry. The position of a page hierarchy in the main navigation bar is determined by the value of the hierarchy's sequence property. You cannot change the properties of system page hierarchies that are provided with the solution, but you can remove them or change their contents.

Procedure

1. In the Administration Console view, click **Page hierarchy**.

Adding a page hierarchy

2. Add a label to create a new page hierarchy. You can add a top-level page hierarchy or you can add a page hierarchy to an existing page hierarchy. The label for a page hierarchy is displayed in the main navigation bar.
   - To add a top-level hierarchy, click **Create**.
   - To add a hierarchy in an existing page hierarchy, click the add icon beside the page hierarchy.
3. In the **Create an Item** window, enter the label details for your custom page hierarchy.
   a) In **Type**, select **Label** in the drop-down list to add a page hierarchy.
   b) In **Name**, enter a name for the page hierarchy. The name is displayed in the main navigation bar.
   c) Optional: In **Description**, enter a description of the page hierarchy.
   d) In **Sequence**, enter a number greater than or equal to 0 as the sequence number for the page hierarchy. The sequence number is a relative value that determines the position of a page hierarchy in relation to other page hierarchies in the main navigation bar. For example, if you have two top-level page hierarchies, then the page hierarchy with the lower sequence value is positioned to the left in the main navigation bar.
   e) Click **Save**.

   Note: A page hierarchy is not displayed in the main navigation bar until it contains one or more pages.

Adding a page to a hierarchy

4. Click the add icon beside the page hierarchy.
5. In the **Create an Item** window, enter the details for your custom page.
   a) In **Type**, select **Page** from the drop-down list.
   b) In **Name**, enter the page name to display in the page hierarchy.
   c) Optional: In **Description**, enter a description of the page.
   d) In **Sequence**, enter a number greater than or equal to 0. The sequence number is a relative value that determines the position of a page in a hierarchy in relation to other pages or page hierarchies.
   e) In **Page definition**, select the page to add to the hierarchy from the drop-down list.
   f) Click **Save**.

   Note: You must refresh the browser to display the new page in the main navigation bar.

Editing a page hierarchy

6. Select a page hierarchy to edit the name, description, and sequence for the page hierarchy.

7. Select a page in a page hierarchy to edit the name, description, sequence, and page definition for the page.
8. Click **Save**.

Deleting a page hierarchy

9. Click the delete icon beside the page hierarchy, and then in the confirmation window, click **Yes**.

Deleting a page from a page hierarchy

10. Click the delete icon beside the page, and then in the confirmation window, click **Yes**.

Changing the number of columns in the list view

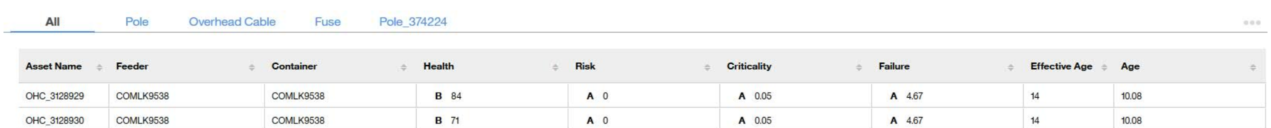
You can change the number of columns that shows in the list view of all applications of IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

The list view has an editing **Columns display** icon. Using the **Columns display** tool, you select the columns that show in the view.

A list view has these columns:

- Asset Name
- Feeder
- Container
- Health,
- Risk,
- Criticality,
- Failure,
- Effective age,
- Age.


All Pole Overhead Cable Fuse Pole_374224

Asset Name	Feeder	Container	Health	Risk	Criticality	Failure	Effective Age	Age
OHC_3128929	COMLK9538	COMLK9538	B 84	A 0	A 0.05	A 4.67	14	10.08
OHC_3128930	COMLK9538	COMLK9538	B 71	A 0	A 0.05	A 4.67	14	10.08

Figure 13. List view

Procedure

1. Sign on IBM Maximo APM for Energy & Utilities as a user.
2. Use the **Filter** selector to make a selection of the **Asset Class**, **Prioritization Criteria**, **Others**.
3. In the navigation segment bar click **List**.
   The list is displayed.
4. Click the **Column display**  icon in the header bar.
5. Select or clear the columns that you need to view.
6. Click OK.

Results

Only the columns that you select show in the list view.

Configuring REST services

If you are an administrator, you can register and configure custom Representational State Transfer (REST) services.

About this task

Use the **Administration** view to register and configure custom REST services. Each service is defined by a set of properties, and each property value that you enter is validated by the solution upon entry.

Procedure

1. In the **Administration** view, click **Service**.

Registering a custom REST service

2. Click **Create**.
3. In the **Create a Service** window, enter the details for your custom service.
   a) In **Name**, enter the name of the service.
   b) In **URI**, enter the URI for the resource.
      For example, `/ibm/ife/api/ui-service/style`.

      Note: You can append `/*` to the base URI for the service to include all the resources that are managed by the service. For example, `/ibm/ife/api/ui-service/*`.
   c) Optional: In **Description**, enter a description of the service.
   d) Click **Save**. The new service is listed under the **Create** button.

      Edit the service to assign access rights to the service to user groups in your solution.

Editing a custom REST service

4. Select the service and edit the values.
   a) Edit the values for the name, URI, and description.
   b) For **Access Control** , assign access rights to the service for the user role groups in your solution. For each user role group, you can select one or more of the access controls that are labeled **Create**, **Read**, **Update**, and **Delete**. For more information about user role groups, see the related link.
   c) Click **Save** to save the changes.

Deleting a custom REST service

5. Click the delete icon beside the service, and then in the confirmation window, click **Yes**.

Configuring notifications

To be able to receive notifications, the user must configure the notifications in the administration page.

After notifications are set up, the user is then able to receive notifications via the end user page or email.

Making a subscription for Notification in IBM Maximo Asset Performance Management for Energy & Utilities SaaS

After Notification is set up you can receive Notification on your system dashboard. You can also receive email notifications.

About this task

On the Notification admin page, you can make a subscription for the system notification.

Procedure

1. In Insights Foundation for Energy, click **Administration** > **Notification** > **Alert Settings**.
2. Click **System Message** > **Item Correlation**.

3. In the **System Dashboard** and **email** fields, you can select the **Receive notifications** check-box for the method for receiving system notifications. You can also select how long a message shows in your System Dashboard.
4. Click **Save**.

Configuring the application

IBM Maximo Asset Performance Management for Energy & Utilities SaaS provides the capability to configure the user interface, focusing on the filter bar and the connectivity filter.

Configuring Time in the filter bar

You can configure the options for the time filter in the table AHA.ANALYSIS_YEAR.

The options for the time filter are derived from the configuration of year analysis table AHA.ANALYSIS_YEAR.

TIME_BUCKET

Interger, is the yearly interval to analyze the given scope.

ANALYSIS_DURATION

Integer, the number of years to be analyzed from the start year.

ANALYSIS_YEAR

Integer, the start year to take analysis as nnnn.

UI Framework

The user interface framework accelerates the development of an application on the front end.

The UI framework consists of several parts:

- Bootstrap
  - Is based on HTML/JSP bootstrap
  - Calls runtime to load the configuration files.
  - Responsible for loading the javascript libraries and the cascading style sheets.
- Configuration
  - Is a JSON format configuration file.
  - Configures the widget parameters.
  - Configures the widget containment.
  - Configures the dependency injection.
- Runtime Library
  - Is a lightweight javascript library.
  - Enables the API to load configuration file and initialize UI widget.
  - Enables the API to dynamically add/remove widgets.
- Ready-to-use widgets - consists of:
  - Container widgets (BorderLayout, TabContainer, ContentPane,)
  - Functional widgets (Map, List, Chart,)
  - Data Model widgets
  - Property binding widgets
  - Behavior widgets
- Ready-to-use application template - The template includes html bootstrap code, the configuration file and custom widgets.

Bootstrap

Bootstrap is an jsp file, that loads the style sheet, loads and initializes the JSP libraries, defines the page level style and calls the runtime to load the configuration.

User can start from below default bootstrap as a template:

- Import custom javascript libraries (e.g. jquery, react.js, etc...)
- Import custom stylesheet

Configuration

The configuration is managed by one JSON file that is loaded by Runtime and includes other required JSON files.

Each item inside configuration file is composed of:

- `id` - a unique id of the widget.
- `module` - the module name the widget.
- `container` - an optional container of the widget.
- `regions` - an optional list of regions if the widget is a container.
- `parameters` - optional parameters for the widget.

Example one - Configure Widget Containment

The widget containment relationship represents a parent - child relationship between the widgets. There is a root container that contains everything to be displayed. The root container widget is contained by a DOM node in the Bootstrap file. The example shows that `div id ="root"` is the division and can be used as the root container.

```
<body class="oneui">
    <div id="root"></div>
    <script>
        require(["ifef/Runtime.js"], function(Runtime){
            window.rt = Runtime;
            Runtime.load("config3.json").then(function(widgets){
                //console.debug("Success",widgets);
            });
        });
    </script>

</body>
```

Figure 14. The division as the root container

Children widgets are contained by a the container widget in a specified region. The example shows the `left` widget being be part of the `main` widget and is located in the left region.

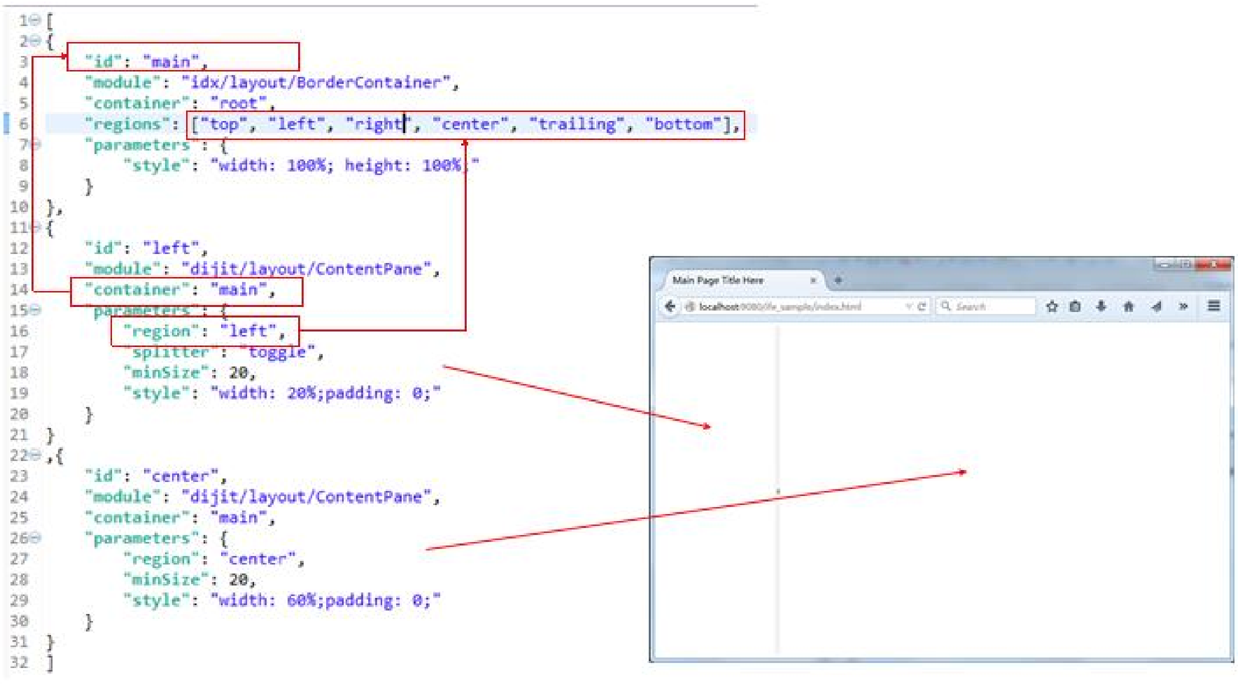


Figure 15. Region container widget

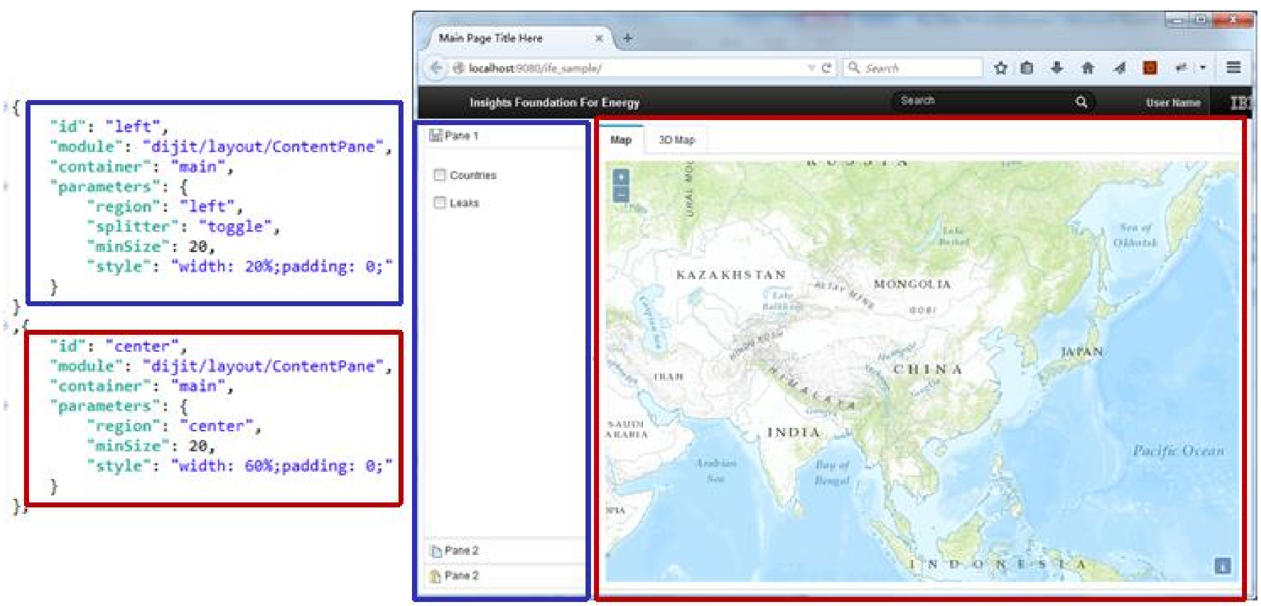


Figure 16. Result for left and center containment

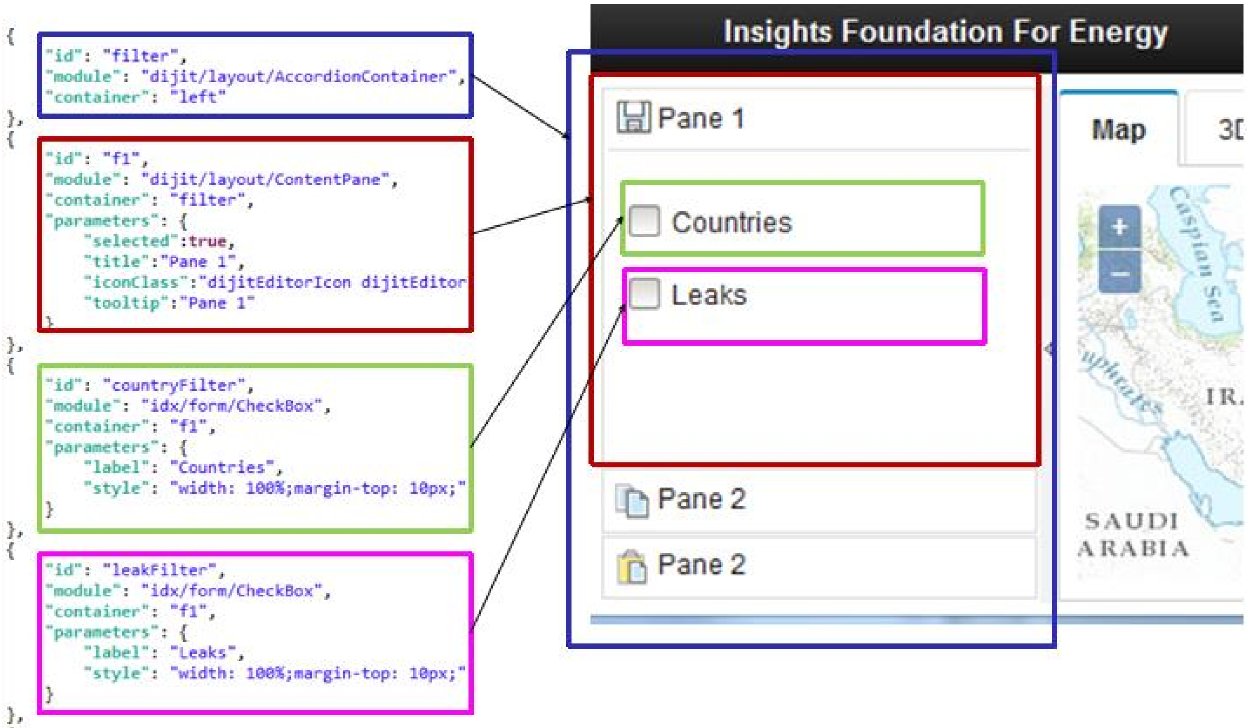


Figure 17. Filter Panes are contained by left widget

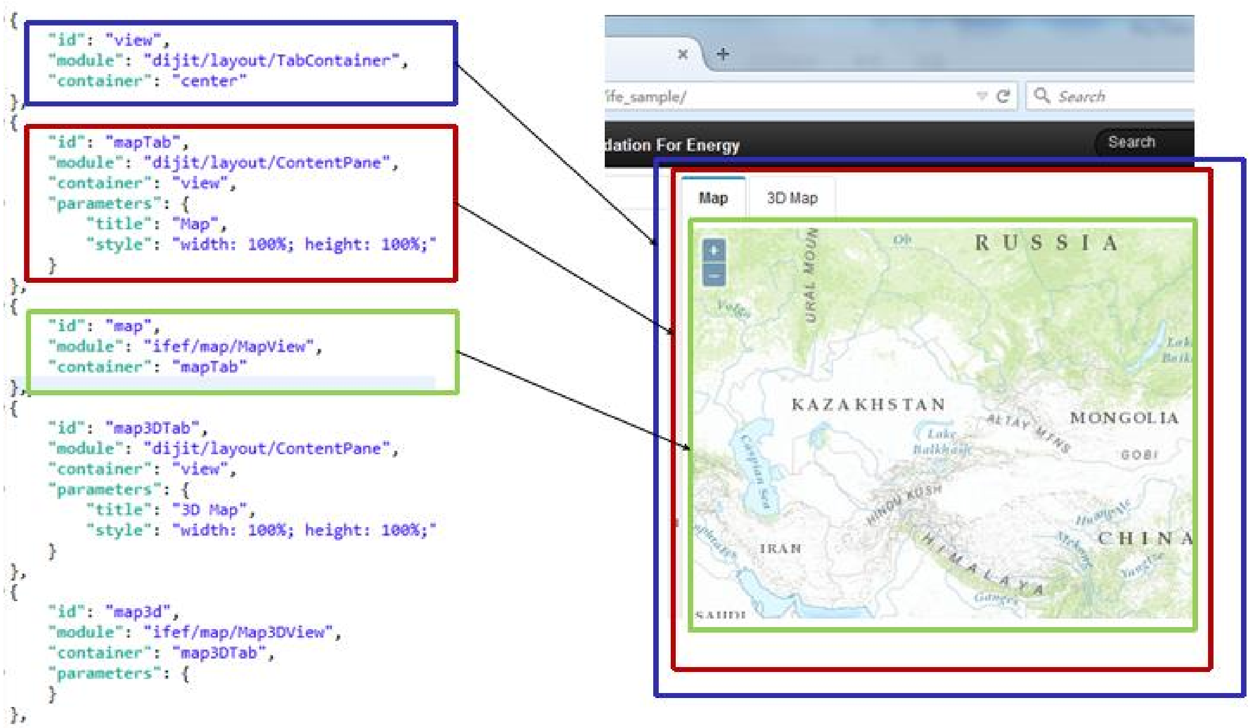


Figure 18. Map & 3D Map contained by center widget

Example two - Configure Dependency Injection

Dependency represents the interaction relationship among different widgets. Dependency injection is the way of UI framework to configure this kind of relationship in JSON format configuration file. The example shows the countryLayer module has the map and the countryFilter dependency is injected, respectively.

Figure 19. The countryLayer module has {map} and {countryFilter} is the dependency

Runtime library

The Runtime library enables the API to dynamically manipulate UI configurations.

The commands for the Runtime Library are:

`add`: add new widget(s)

Parameters:

- `widget`: a widget configuration or an array of widget configurations, a widget instance or array of widget instances.
- `container`: optional, the ID of the container where the widgets are added.
- `region`: optional, the name of region of the container.

Returns: a `dojo/Deferred` object.

`remove`: remove widget

Parameter:

- `id`: The ID of the widget to be removed.

`Returns`: the array of the specified widget and its children that are removed. The user can reattach them to another container or region by calling an add command, or totally remove the widgets.

`get`: get widget instance

Parameter:

- `id`: the ID of the widget.

Returns: the widget instance of the specified ID.

`load`: load widget from file

Parameters:

- `url`: the address of the configuration file.

Returns:

- `dojo/Deferred` object.

Note:

- The Runtime library also maintains a global registry of widgets, the contents of the registry changes when user calls an add, `remove`, or `load` command.
- The Runtime library can be used in both bootstrap and custom widgets.

```
<script>
    define([
        "dojo/_base/declare", // declare
        "dojo/_base/lang",
        "dojo/Stateful",
        "ifef/Runtime"
    ], function(declare, lang, Stateful, Runtime){
    });
</script>
```

Figure 20. Bootstrap and custom widgets

Ready-to-use widgets

Ready-to-use widgets are widgets which can be used directly.

There are several types of widgets listed:

- Container widgets

  Dojo and IDX container widgets are available to use, the most common container widgets listed below. For other IDX container, please refer to IDX documentation.
- Reusable functional widgets: IFE provided widgets for a specific functionality:
  - Filter
  - Preview Card
  - Map
  - Data Layer
  - Logical Map
  - List
  - List Container
  - LineChart
  - BarChart
  - HeaderButton
  - Timer
- Behavior widgets
  - Property Binding Widgets. Property binding depends on the mechanism of getting, setting, and watching for property changes, we leverage **dojo/Stateful** and **dijit/_WidgetBase** to provide this mechanism. For further information refer to Stateful and WidgetBase in the Dojo Toolkit Reference Guide.

    The example shows the binding from the **checked** property of **countryFilter** to**filterSelected** property of **countryLayer**. When user checks the checkbox, **countryLayer** is notified: the **_filterSelectedSetter** method of countryLayer will be called to execute the corresponding actions:

```
},
{
    "id": "bind1",
    "module": "ifef/behavior/Bind",
    "parameters": {
        "bindings": [
            {"source": "countryFilter", "sourceProp": "checked", "target": "countryLayer", "targetProp": "filterSelected"}
        ]
    }
}
]
```

Figure 21. The filterSelected method

```
define(["dojo/_base/declare", "dojo/_base/lang", "dojo/Stateful"],
function(declare, lang, Stateful){
    // module:
    //      ifef/behavior/CountryLayer
    return declare("ifef.hebavior.GeoJsonLayer", [Stateful], {
        data: null,

        _filterSelectedSetter: function(selected/*boolean*/){
            if(selected){
                this.layer = this.layer ? this.layer : new ol.layer.Vector({
                    source: new ol.source.Vector({
                        format: new ol.format.GeoJSON(),
                        url: this.data
                    })
                });
                this.map.map.addLayer(this.layer);
            } else {
                if(this.layer){
                    this.map.map.removeLayer(this.layer);
                }
            }
        },

```

Figure 22. Execute action

– Special Usage for Binding Widget

bindOnlyIfUnequal: only notify the target if the old value is not equal to new value.

```
{
    "id": "bind1",
    "module": "ifef/behavior/Bind",
    "parameters": {
        "bindings": [
            {
                "source": "filter1",
                "sourceProp": "checked",
                "target": "countryLayer",
                "targetProp": "filterSelected",
                "bindOnlyIfUnequal": true
            }
        ]
    }
}
```

Figure 23. Notifies the target if the two values do not agree

convertor and func: function of injected convertor will be called before setting the value to the target. The functions give a conversion between different formats.

```
{
    "id": "bind1",
    "module": "ifef/behavior/Bind",
    "parameters": {
        "convertor": "@{convertor1}",
        "bindings": [
            {
                "source": "filter1",
                "sourceProp": "checked",
                "target": "countryLayer",
                "targetProp": "filterSelected",
                "bindOnlyIfUnequal": true,
                "func": "conver"

            }
        ]
    }
}
```

Figure 24. Convertor and function

- Data Model widgets
  - Model

    The Data Model widget wraps the service created by the service framework. Two parameters should be specified for it:

    `target:` The URL address of the service.

    `idProperty:` The ID property of the response value.

```
{
    "id": "model1",
    "module": "ifef/model/Model",
    "properties": ["criteria", "store"]
    "parameters": {
        "target": "/url/to/rest/service",
        "idProperty": "id"
    }
}
```

Figure 25. ID property of the response value

The Data Model widget provides two properties for binding:

`criteria:` the filter criteria.

`store:` the dojo store that provides access to data that is applied to the filter criteria, where filter1 is the filter widget which provide filter criteria, and layer1 is the map layer widget that consumes the dojo store:

```
{
    "id": "bind1",
    "module": "ifef/behavior/Bind",
    "parameters": {
        "bindings": [
            {"source": "filter1", "sourceProp": "criteria", "target": "model1", "targetProp": "criteria" },
            {"source": "model1", "sourceProp": "store", "target": "layer1", "targetProp": "store"}
        ]
    }
}
```

Figure 26. Provides access to data

  - ModelSelector

    The ModelSelector widget is used to dynamically determine the model at the runtime. Its parameter is **models** that lists each key associated with each respective model. An example is as below:

```
{
    "id": "readingSelector",
    "module": "ifef/model/ModelSelector",
    "properties": ["selector", "store"],
    "parameters": {
        "models": [
            {"key": "Load_Index", "model": "@{loadIndexModel}"},
            {"key": "Health_Index", "model": "@{healthIndexModel}"},
            {"key": "Impact_Level", "model": "@{impactLevelModel}"}
        ]
    }
}
```

Figure 27. The Models parameter with associated keys

ModelSelector widget provides two properties available for binding:

`selector:` the dojo object consists of a key and filter criteria.

`store:` the dojo store that provides access to data that is applied according to the selector. In the example, the selection in the measurement list triggers the change of reading list.

```
{"source": "@{measurementList}", "sourceProp": "selector", "target": "@{readingSelector}", "targetProp": "selector"},
{"source": "@{readingSelector}", "sourceProp": "store", "target": "@{readingList}", "targetProp": "store"},
```

Figure 28. Provides access applied according to the selector

Reusable widgets
Widgets are provided that can be modified to the needs of the user.

The parameters for each of the widgets are set by the user, the properties set how the widget displays in the user interface.

Filter widget

The filter widget is used to organize the filter criteria for the result of IFE REST service.

The widget is made up of a check box and an additional dialog box.

ifef.widget.filter.Filter

label:

String, shows as the label for the check box, e.g. Name.

enabled:

Boolean, specifies whether the criteria is enabled or not.

searchFields:

An array, describes the search fields in the more dialog filter:

- name: String, the name of the field.
- type: String, the type of field. Supports three string types: string, number, date.
- label: String, label of the field to show in the more dialog box
- regExp: String, regular expression that validates the user input value. Available only when the type is string and ‘options’ is not defined.

options:
An array, predefined values that the user can select:

- label: the label shows from a drop-down list.
- value: predefined values

The properties

The properties are:

criteria:

An object that describes filter criteria. The property is changed when the user either clicks a check box or the **More Filter** dialog box is changed.

Example

The example shows the use of a filter widget.

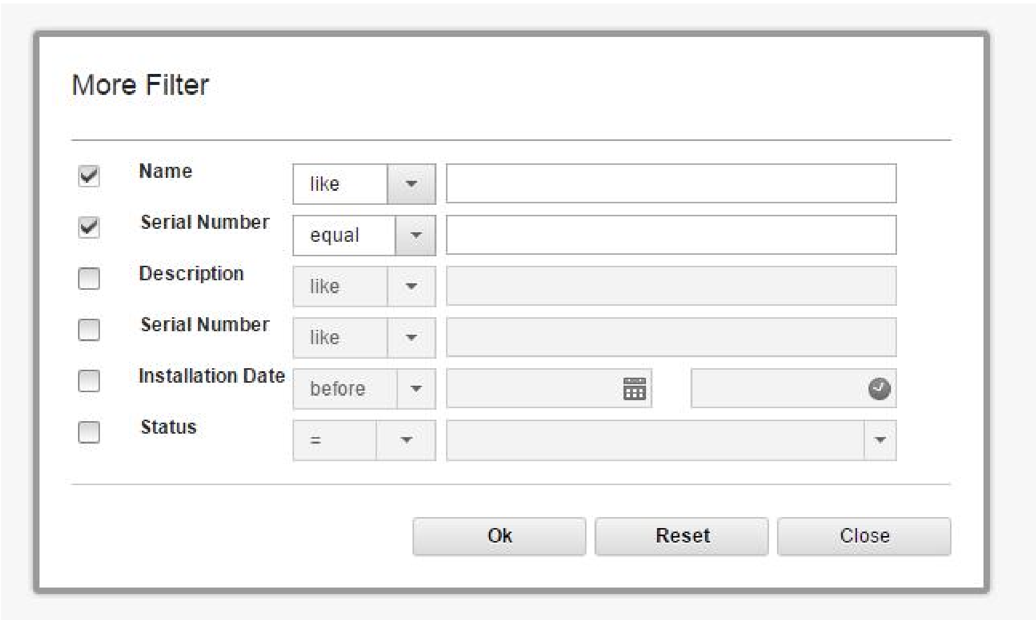


Figure 29. Example filter widget

```
{
       "id":"filter1",
       "module":"ifef/widget/filter/Filter",
       "container":"group1",
       "properties":["criteria"],
       "parameters": {
          "label": "Filter 1",
          "enabled":true,
          "logistic":"and",
          "searchFields":[
               {"name":"name", "type":"string",
      "label":"Name", "regExp":"([A-Za-z0-9][A-Za-z0-9-_]*)"},
               { "name":"createTime","type":"date", "label":"Create
Time"},
               {"name":"updateTime","type":"dateTime", "label":"Update
Time"},
               {"name":"isActive","type":"boolean", "label":"Active"},
               {
      "name":"status","label":"Status", "type":"string", "multiple":true,
               "options":[
               {"label":"Critical","value":"1"},
               {"label":"Warning", "value":"2"},
               {"label":"Ok", "value":"3"},
               { "label":"NoScore", "value":"4" }]
               }
      ]
}}
```

PreviewCard widget

The PreviewCard is a temporary card that shows the details panel for an asset, with the ability to select a **More Details** dialog box and an **More Actions** menu.

The user can select an asset from the map, then preview the details on a preview card with the information of asset.

ifef.widget.previewcard.PreviewCard

titleProperty:

String, the name of property in the data object. The property value shows as the title of preview card.

position:

A String or Array that specifies the position of the preview card. If it is a string, then the string content is the property name Position in the data object. The property name Position is defined by an array of two numbers in the data object or a domNode. If it is a array with two numbers, the previewcard shows in the fixed position. The first number of array specify x value and the second number specify y value of current page.

properties:

An array that specifies the properties that show on the preview card content panel or more details dialog box.

- name: String, the property name.
- label: String, label of property that shows on content panel.
- isKey: Boolean, if true this property shows on content panel, if false the property shows on the **More details** dialog box.
- index: Numeric, if an index is specified, shows the indexed order.
- render: Object, defines the rules to render the property value:
  - numberFormatOptions: Object, shows the options to format a number as a string. Refer to the dojo/number document: http://dojotoolkit.org/api/?qs=1.10/dojo/number.
  - - dateFormatOptions: Object, shows the options to format a date as a string. Refer to the dojo/date/locale document http://dojotoolkit.org/api/?qs=1.10/dojo/date/locale.
- templates: Array, defines the conditions and template strings to render property values.
  - condition: String, a condition returns either a true or false status. Any property in `data` can be referenced as a condition, for example: STATUS=1
  - content: String, an html segment template that contains variables.
  - variables: Object, key or value pairs used to replace the variables in the content.

moreActions:

Array, shows the **More actions** menu

- label: String, shows the name of the menu item.
- children: Array, defines the sub-menu items.
- func: Function, invoke when you click the menu item. Available when sub-menu items are not specified.
- options: Object, used by the ‘func’ function.

The properties

The properties are:

data:

A key or value object, when the data is set the review card is filled by the data object, and shows.

Example
The example shows the use of a preview card.

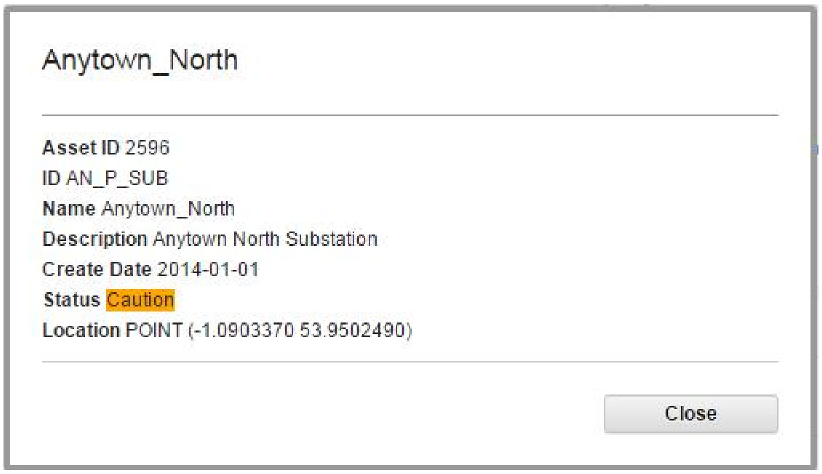


Figure 30. Preview card

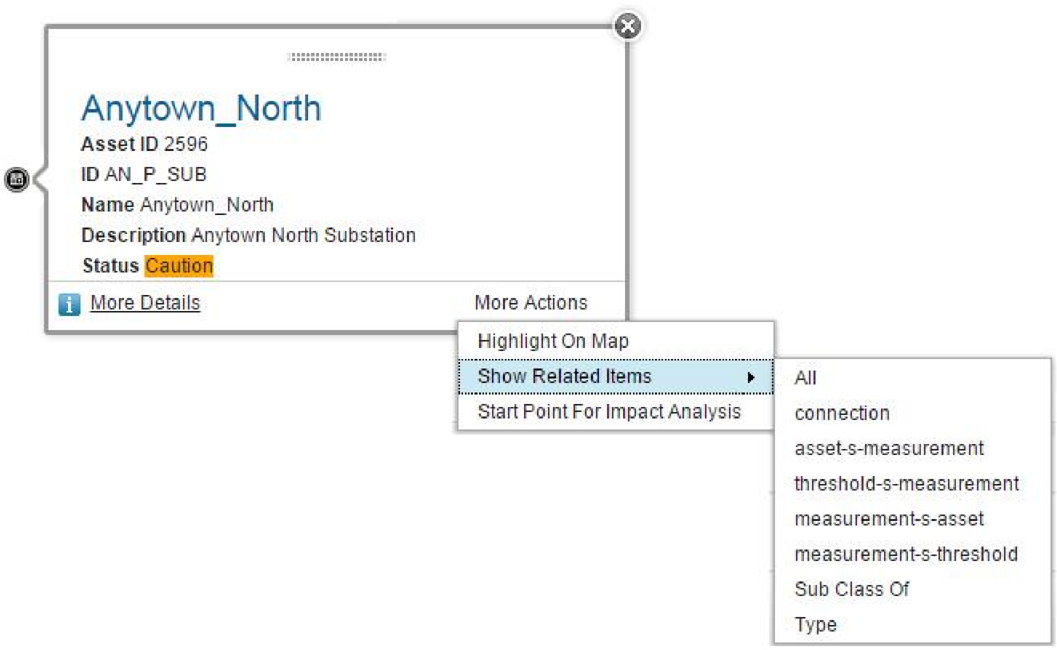


Figure 31. Preview card with more details and more action dialog boxes

```
{
     "id":"overheadlineMapPreviewCard",
     "module":"ifef/widget/previewcard/PreviewCard",
     "properties":["data"],
     "parameters": {
          "titleProperty": "NAME",
          "position": "_position",
          "properties": [
               {"label":"@{dno_nls.Asset_ID}", "name":"ASSET_ID",
"isKey":true},
               {"label":"@{dno_nls.Serial_Number}","name":"SERIAL_NUMBER",
"isKey":true},
               {"label":"@{dno_nls.NAME}","name":"NAME", "isKey":
true},
               {"label":"@{dno_nls.Description}","name":"DESCRIPTION",
"isKey":true},
               {

"label":"@{dno_nls.STATUS}",
                    "name":"STATUS",
                    "isKey": true,
```

```
                "render":{
      "templates": [
           {
           condition":"STATUS==0",
           "content":"<span style='background-color:green'>${Acceptable}</
span>",
           "variables":{"Acceptable":"@{dno_nls.Acceptable}"}
      },{
           "condition":"STATUS==1",
           "content":"<span style='background-color:orange'>${Caution}</
span>",
           "variables":{"Caution":"@{dno_nls.Caution}"}
      }]
           }},
                {"label":"@{dno_nls.Location}","name":"LOCATION"}
           ],
           "moreActions":[  { "label":"@{nls.HighlightOnMap}",
           "func":"@{highlightOnMap.highlightOnMap}"} ]
             }
      }
```

Map widget

The administrator can configure the map controls that show on map for the user.

The administrator can what zoom and positing controls show on a map for the user. The administrator can also configure multiple base layers with the control parameters and attributes for each base layer.

ifef.widget.Map

The parameters for the mapwidget are:

controls:

The administrator can configure which map controls show on a map.

- type： Possible values are "ol.control.Zoom", "ol.control.ZoomSlider", "ol.control.FullScreen"," ol.control.MousePosition", ol.control.OverviewMap, "ol.control.Rotate", "ol.control.ScaleLine"
- parameters: each parameter has a shows a different type of mapping. See the type and parameters mapping table bellow:

Table 12. Settings for the control parameter

Type	Parameters	Description
ol.control.Zoom	zoomInLabel: Text label used for the zoom in button. Default is +. zoomOutLabel: Text label to use for the zoom out button. Default is - zoomInTipLabel: Text label to use for the button tip. Default is Zoom in. zoomOutTipLabel: Text label to use for the button tip. Default is Zoom out.	A control with 2 buttons, one for zoom in and one for zoom out. This control is one of the default controls of a map. default is show.
ol.control.ZoomSlider:	nul	A slider type of control for zooming in and out.

Type	Parameters	Description
ol.control.FullScreen:	label: Text label to use for the button. The default is an arrow. tipLabel: Text label for the button tip. Default is Toggle full-screen. ol.control.MousePosition: A control to show the 2D coordinates of the mouse cursor.	A button that when clicked fills the screen with the map. When in full screen mode, a close button shows to exit full screen mode.
ol.control.OverviewMap:	collapsed: Whether the control should start collapsed or expanded. Default to true. collapseLabel: Text label to use for the expanded overviewmap button. Default is «. collapsible:Whether the control can be collapsed or not. Default to true. label:Text label to use for the collapsed overviewmap button. Default is ». layers:Layers for the overview map. tipLabel:Text label to use for the button tip. Default is Overview map	Create a new control with a map acting as an overview map for an other defined map

Table 12. Settings for the control parameter (continued)

view

The administrator can configure the view portion of the map.

- **center**: An array of numbers representing an xy coordinate [pointX, pointY]. This is the initial center for the view. The coordinate system for the center is **EPSG:4326**
- **zoom**: Only used if the resolution is not defined. The zoom level is used to calculate the initial resolution for the view. The initial resolution is determined using the **ol.View#constrainResolution** method.
- **extent**: An array of numbers representing an extent: [minx, miny, maxx, maxy]. The extent constrains the center, the center cannot be set outside the extent. The default is undefined. The coordinate system for the extent is **EPSG:4326**.
- maxZoom: The maximum zoom level used to determine the resolution constraint. It is used together with minZoom (or maxResolution) and zoomFactor. Th default is 28.

  Note: If **minResolution** is provided, it has precedence over **maxZoom**.
- **minZoom**:The minimum zoom level used to determine the resolution constraint. It is used together with maxZoom (or minResolution) and zoomFactor. Default is 0. Note that if maxResolution is also provided, it is given precedence over minZoom.

layers

The administrator can configure multiple base map layers.

- **visible**: boolean type, the default is true.
- **title**: The title of the layer, you can view it on the map control.
- **type**: The possible values are **ol.source.OSM**, **ol.source.MapQuest**, **ol.source.stamen**, **ol.source.xyz**, **ol.source.BingMaps**, **ol.source.TileArcGISRest**, **ol.source.TileJSON**, **ol.source.TileWMS**. Different layer types use different parameters.
- **parameters**: The object type, the parameters it is different according to the type you select. Parameters is the same with openlayer3. See the type and parameters mapping table bellow.

Table 13. Type and parameter mapping

Type	Description
ol.source.OSM	Optional.. A URL template that includes {x}, {y} or {-y}, and {z} placeholders. The default is http://{a-c}.tile.openstreetmap.org/{z}/{x}/{y}.png.
ol.source. MapQuest	Layer. The possible values are: `osm`, `sat`, and `hyb`. A URL template that includes {x}, {y} or {-y}, and {z} placeholders. A value for Layer is mandatory, the URL is optional.
ol.source.stamen	Layer. The possible values are watercolor, terrain-labels. A URL template that includes {x}, {y} or {-y}, and {z} placeholders. A value for Layer is mandatory, the URL is optional.
ol.source. BingMaps	The Bing map key is mandatory, an example key is: Ak-dzM4wZjSqTlzveKz5u0d4IQ4bRz VI309GxmkgSVr1ewS6iPSrOvOKhA-CJlm3 The options for the `imagerySet`: are: `Road`, `Aerial`, `AerialWithLabels`, `collinsBart`, and `ordnanceSurvey`.
ol.source.xyz	A URL template that includes {x}, {y} or {-y}, and {z} placeholders.
ol.source. TileArcGISRest	Call for an ArcGIS REST service URL for a Map Service or Image Service. The url should include either `/MapServer` or `/ImageServer`.
ol.source. TileJSON	The URL link to the address to the `TileJSON` file.

Table 13. Type and parameter mapping (continued)	
Type	**Description**
ol.source. TileWMS	WMS service URL. serverType: The available server types are: '`carmentaserver`, `geoserver`, `mapserver`, `qgis`. params: WMS request parameters. A LAYERS parameter is mandatory. STYLES is ' ' by default. VERSION is 1.3.0 by default. WIDTH, HEIGHT, BBOX and CRS (SRS for WMS version < 1.3.0) are set dynamically.
ol.source.WMTS	url: A URL for the service. For KVP encoding, it is a normal URL. For the RESTful request encoding, this is a URL template of the pattern {?-?}, for example: subdomain{a-f}.domain.com. attributions:Attributions. layer: Layer name as advertised in the WMTS capabilities. Mandatory. matrixSet: Matrix set. Mandatory. format: Image format. Default is image/jpeg. projection: Projection. tileGrid: Tile grid. Set the grid pattern for sources accessing WMTS tiled-image servers.

Example

The example shows the use of a map widget.

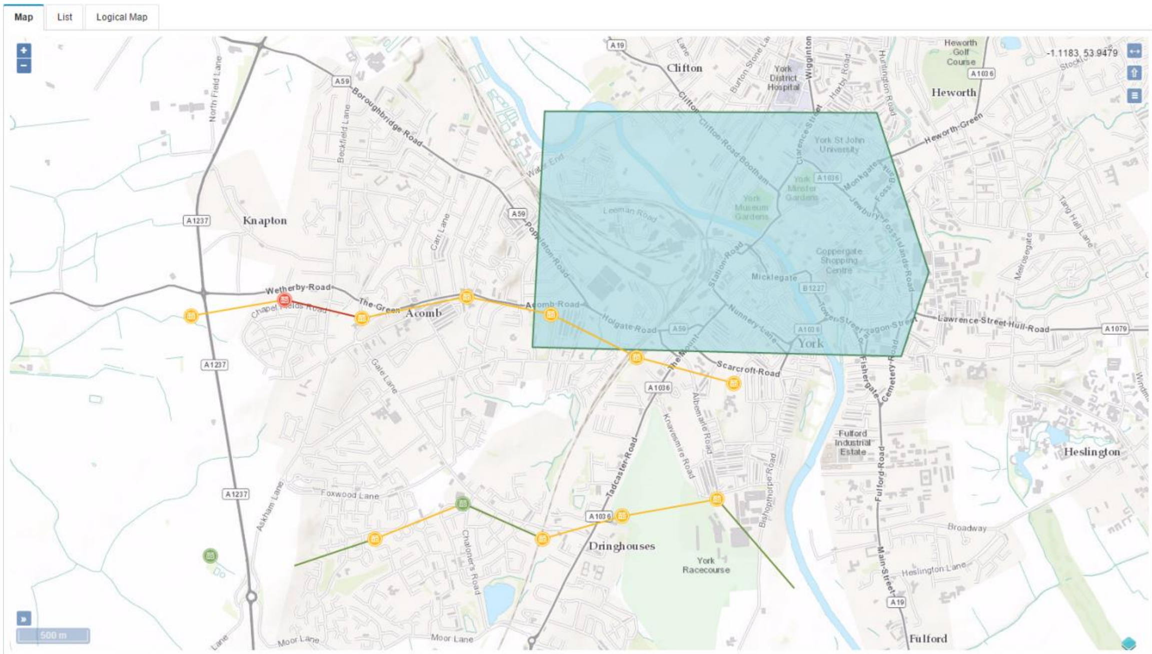


Figure 32. Map widget

```
{
     "id": "map",
     "module": "ifef/widget/Map",
     "container": "mapTab",
          "parameters": {
               "controls":
[
                            {

"type":"ol.control.Zoom",
                               "parameters":
{

"zoomInTipLabel":"@{nls.Map_Zoom_In_Tip_Label}",


"zoomOutTipLabel":"@{nls.Map_Zoom_Out_Tip_Label}"

                                            }
                            },
                            {

"type":"ol.control.ZoomSlider",
                               "parameters":{}
                            },
                            {

"type":"ol.control.ScaleLine",
                               "parameters":
{
                                    "minWidth":"64",

"units":"metric"
                                            }
                            },
                            {

"type":"ol.control.FullScreen",
                               "parameters":
{

"label":"\u2194",

"tipLabel":"@{nls.Map_Full_Screen_Tip_Label}"

                                            }
                            },
                            {
```

```
"type":"ol.control.Rotate",
                              "parameters":
{
"autoHide":false,
"tipLabel":"@{nls.Map_Rotate_Tip_Label}"
                                            }
                           },
                           {
"type":"ol.control.OverviewMap",
                              "parameters":
{
"tipLabel":"@{nls.Map_Overview_Tip_Label}"
                                            }
                           },
                           {
"type":"ol.control.MousePosition",
                              "parameters":
{
                                   "projection":
"EPSG:4326"
                                            }
                           }
                         ],
               "layers":[
                           {
"type":"ol.source.OSM",
"visible":false,
"title":"@{nls.Map_Layer_Title_OSM}",
                              "parameters":{}
                           },
                           {
"type":"ol.source.MapQuest",
"visible":false,
"title":"@{nls.Map_Layer_Title_MapQuest_Sat}",

                              "parameters":
{
                                   "layer":
"sat"
                                            }
                           },
                           {
"type":"ol.source.Stamen",
"visible":false,
"title":"@{nls.Map_Layer_Title_Stamen}",
                              "parameters":
{
                                   "layer":
"watercolor"
                                            }
                           },
                           {
"type":"ol.source.XYZ",
"visible":true,
"title":"@{nls.Map_Layer_Title_XYZ_ArcgisOnline}",

                              "parameters":
{
                                   "url":"http://server.arcgisonline.com/ArcGIS/
rest/                                                                     services/
World_Topo_Map/MapServer/tile/{z}/{y}/{x}"
                                            }
                           },
                           {
```

```
"type":"ol.source.BingMaps",

"visible":false,

"title":"@{nls.Map_Layer_Title_BingMaps}",
                              "parameters":
{
                                   "key":"Ak-dzM4wZjSqTlzveKz5u0d4IQ4bRzVI309GxmkgS
      Vr1ewS6iPSrOvOKhA-
CJlm3",
                                   "imagerySet":
"Road"
                                           }
                           },
                           {

"type":"ol.source.TileArcGISRest",

"visible":false,

"title":"@{nls.Map_Layer_Title_TileArcGISRest}",

                              "parameters":
{
                                   "url":"https://services.arcgisonline.com/arcgis/
rest/
      services/ESRI_Imagery_World_2D/
MapServer"
                                           }
                           },
                           {

"type":"ol.source.TileJSON",

"visible":false,

"title":"@{nls.Map_Layer_Title_TileJSON}",
                              "parameters":
{
                                   "url":"http://
api.tiles.mapbox.com/v3/
      mapbox.geography-class.jsonp"
                                           }
                           },
                           {
                              "type":"ol.source.TileWMS",

"visible":false,

"title":"@{nls.Map_Layer_Title_TileWMS}",
                              "parameters":
{
                                   "url":"http://demo.boundlessgeo.com/geoserver/
wms",
                                   "params": {"LAYERS":
"ne:ne"}
                                           }
                           },
                           {

"type":"ol.source.WMTS",

"visible":false,

"title":"@{nls.Map_Layer_Title_WMTS}",
                              "parameters":
{
                                   "url":"http://services.arcgisonline.com/arcgis/
rest/
                                          services/Demographics/
USA_Population_Density
                                          /MapServer/
WMTS/",
                                   "layer":
"0",
                                   "matrixSet":
"EPSG:3857",
                                   "format": "image/
png",

"projection":"EPSG:3857",
```

```
"style":"default",
                                    "wrapX":
"true",
                                    "tileGrid":
{
"type":"ol.tilegrid.WMTS",
                                         "parameters":
{
                                         "origin":
[-20037508.342789244,

20037508.342789244],
                                         "resolutions":
[156543.03392804097,

78271.51696402048,

39135.75848201024,

19567.87924100512,

9783.93962050256,

4891.96981025128,

2445.98490512564,

1222.99245256282,

611.49622628141,

305.748113140705,

152.8740565703525,

76.43702828517625,

38.21851414258813,

19.109257071294063],
                                         "matrixIds":[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                                                      11, 12,
13]
                                                        }

                                              }
                            }
                         }
                         ],
                "view": {
                            "zoom":5,
                            "maxZoom": "4",
                            "center": [104.31,28.72],
                            "extent":[-180,-80,180,80]
                            "minZoom": "8"
                         }
                       }
    }
```

DataLayer widget

The Datalayer widget is used to show a data layer on the map.

The administrator can configure additional data layers in Insight Foundation for Energy. The data can come from an IFE REST service or another data file in standard format. The formats that are currently supported are: GeoJSON, TopoJSON, EsriJSON, IGC, Polyline, WKT, GML, GPX, KML, OSMXML.

You can specify the data type in the Datalayer widget with the formatType parameter.

You can specify the data link in the ifef/model/Model widget by the target parameter.

You can configure the style for each data layer and set a condition for each style. Each data layer can be set for a custom range, fill, stroke and shape. You can also set text styles for each layer. The user can set the order of the layers.

Container

The container for the DataLayer must be a map.

Properties

Store
: You can retrieve stored data.

criteria
: An object that describes the filter criteria, this property is changed when the user click selects a check box or changes the more filter. If you bind this property with a model and the model is bound to a filter, you can use this properties to control how a layer is displayed or not.

Selected
: Indicates that a user has selected an asset in the data layer. An asset can also be selected by the settings the selected properties value.

Parameters for ifef.widget.DataLayer

map:
: The map ID that the data layer is dependent on. For example: @{map}.

index:
: The order of the data layers. The greater the value or the layer the higher the layer.

formatType:
: If data comes from a REST service, then this parameter setting is optional. The formats that are supported are: GeoJSON, TopoJSON, EsriJSON, IGC, Polyline, WKT, GML, GPX, KML, and OSMXML. You can make a selection based on your service type.

geometryName:
: The parameter is required when the data is from the REST service and geometry field Name is not LOCATION. Default is “LOCATION”.

Styles:
: The user can configure multiple styles.

- Condition - you can set the condition that uses the style.
- Style:
  - fill: Set the fill style for vector features.

    type: "ol.style.Fill"

    parameters: same as the openlayer "ol.style.Fill" class API.

    color: Colors can be defined as strings as rgb (r,g,b) or rgba (r,g,b,a) formats, or in hex #rrggbb or #rgb format. The color names, 'red', 'blue' or 'green', can be used with the Canvas renderer. Default null; if null, the Canvas/renderer default black is used.

- stroke: Set the stroke style for vector features.

  type: "ol.style.Stroke"

  parameters: same as the openlayer "ol.style.Stroke" class API.

Table 14. Stroke parameters	
Parameter	**Description**
color:	Colors can be defined as strings as rgb (r,g,b) or rgba (r,g,b,a) formats, or in hex #rrggbb or #rgb format. The color names, 'red', 'blue' or 'green', can be used with the Canvas renderer. Default null; if null, the Canvas/renderer default black is used.
lineCap	Line cap style: butt, round, or square. Default is round.
lineJoin:	Line join style: bevel, round, or miter. Default is round.
lineDash:	Line dash pattern. Default is undefined (no dash).
miterLimit:	Miter limit. Default is 10.
width:	Line width.

- image: - A circle, icon, or a RegularShape.

  type: "ol.style.Circle", "ol.style.Icon" , "ol.style.RegularShape",

  parameters: same as the openlayer's "ol.style.Circle" ,"ol.style.Icon","ol.style.RegularShape" API

  if the type is ol.style.Circle:

Table 15. Style is ol.style.Circle	
Parameter	**Description**
fill:	Set fill style for vector features. (Object) Refer to openlayer API for more information.
stroke:	Set stroke style for vector features. (Object) Refer to the openlayer API for more information.
radius:	Radius of the circle.

  if the type is ol.style.Icon:

Table 16. Style is ol.style.Icon	
Parameter	**Description**
fill:	Set fill style for vector features. (Object) Refer to openlayer API for more information
stroke:	Set stroke style for vector features. (Object)
anchor:	Anchor. Default value is [0.5, 0.5] (icon center).
anchorOrigin:	Origin of the anchor: bottom-left, bottom-right, top-left or top-right. Default is top-left.

Table 16. Style is ol.style.Icon (continued)	
Parameter	**Description**
offset:	Offset, which, together with the size and the offset origin, define the sub-rectangle to use from the original icon image. Default value is [0, 0].
offsetOrigin:	Origin of the offset: bottom-left, bottom-right, top-left or top-right. Default is top-left.
opacity:	Sets the opacity of the icon. Default is 1.
scale:	Scale
rotation:	Sets the rotation in radians (positive rotation clockwise). Default is 0.
src:	The URL of the image source.
size:	Icon size in pixel. Can be used together with offset to define the sub-rectangle to use from the origin (sprite) icon image.

If the type is ol.style.IRegularShape: Then the resulting shape is a regular polygon when radius is provided, or a star when radius1 and radius2 are provided.

Table 17. Style is ol.style.IRegularShape	
Parameter	**Description**
fill:	Set fill style for vector features.(Object)
stroke:	Set stroke style for vector features. (Object)
points:	Number of points for stars and regular polygons. In the case of a polygon, the number of points is the number of sides.
radius:	The circumradius of a regular polygon.
radius1:	The inner radius of a star.
radius2:	The outer radius of a star.
angle:	The angle from the vertical of a shape in radians. A value of 0 will have one of the shape's point facing up. Default value is 0.
rotation:	The angle of rotation in radians (positive rotation clockwise). Default is 0.

– text: - Sets the text style for vector features.

  if the type is "ol.style.Text"

Table 18. Style is ol.style.Text	
Parameter	**Description**
font:	The font used as a CSS font value. .
offsetX:	The horizontal text offset in pixels. A positive value shifts the text to the right. Default is 0.

Parameter	Description
offsetY:	The vertical text offset in pixels. A positive shifts the text down. Default is 0.
scale:	Scale
rotation:	The angle of rotation in radians (positive rotation clockwise). Default is 0.
text:	The text content.
textAlign:	The text alignment. The values are: 'left', 'right', 'center', 'end' and 'start'. The default is 'start'..
textBaseline:	Text base line. Possible values: 'bottom', 'top', 'middle', 'alphabetic', 'hanging', 'ideographic'. Default is 'alphabetic'.
fill:	Set fill style for vector features. (Object).
stroke:	The stroke style for vector features.

Table 18. Style is ol.style.Text (continued)

Example

```
{"id": "overheadlineLayer",
     "module":   "ifef/widget/map/DataLayer",
     "container":   "map",
     "properties":   ["store", "creteria","selected"],
     "parameters":
     {
          "index":2,
          "map":"@{map}",
          "styles":[
          {
               "condition":   "STATUS==0",
               "style":   {
                    "stroke":
{

"type":"ol.style.Stroke",

                              "parameters":
{
                                   "color":
"#699037",

                                   "width":   2
                                              }

                             }
                          }
          },
          {
               "condition":   "STATUS==1",
               "style": {
                    "stroke":
{

"type":"ol.style.Stroke",

                              "parameters":
{
                                   "color":
"#FDBA1A",

                                   "width":
2
                                            }

                             }
                        }
```

```
          },{
               "condition":   "STATUS==2",
               "style": {
                    "stroke":
{

"type":"ol.style.Stroke",

                              "parameters":
{
                                   "color":
"#C32E14",

                                   "width":
2
                                            }

                              }
                         }
          }]
     }
},
{"id": "overheadlineModel",
     "module":   "ifef/model/Model",
     "properties":   ["store"],
     "parameters":
     {
          "storeModule":   "ifef/model/Store",
              "target":"/ibm/ife/sample/dno/api/overheadline",
              "idProperty":   "ASSET_ID"
     }
},
{"id": "bind1",
     "module":"ifef/behavior/Bind",
     "parameters":{
          "bindings":[
               {"source": "@{overheadlineModel}",
                "sourceProp": "store",
                "target": "@{overheadlineLayer}",
                "targetProp": "store"
               }
          ]
     }
},
```

Logical map widget

The logical map shows the relationship between assets and between an asset and a type of measurement.

The user can query the logical map via a keyword of the instance. The logical map can be extended to show related items, and the user can analyze the impact between instances.

In the illustration, the node in the logical map represents the instance, the line represents the relationship.

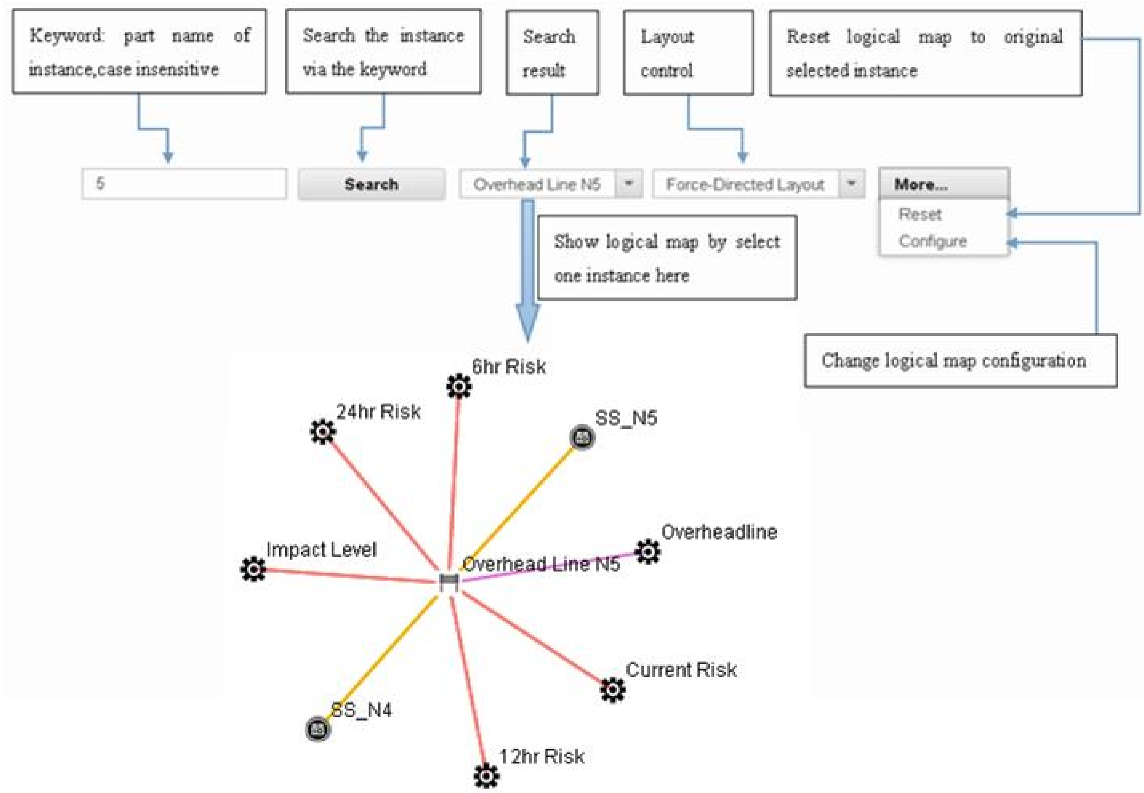


Figure 33. Logical map widget

The framework of a logical map is in three parts:

1. The logical map and its services. The data is queried from the model server via services and the model server interface.
2. The crawler and model server, The crawler generates the OWL file based on the configuration.
3. The user defined part. This includes the data model and its service, and the configuration for the crawler to generate the OWL file.

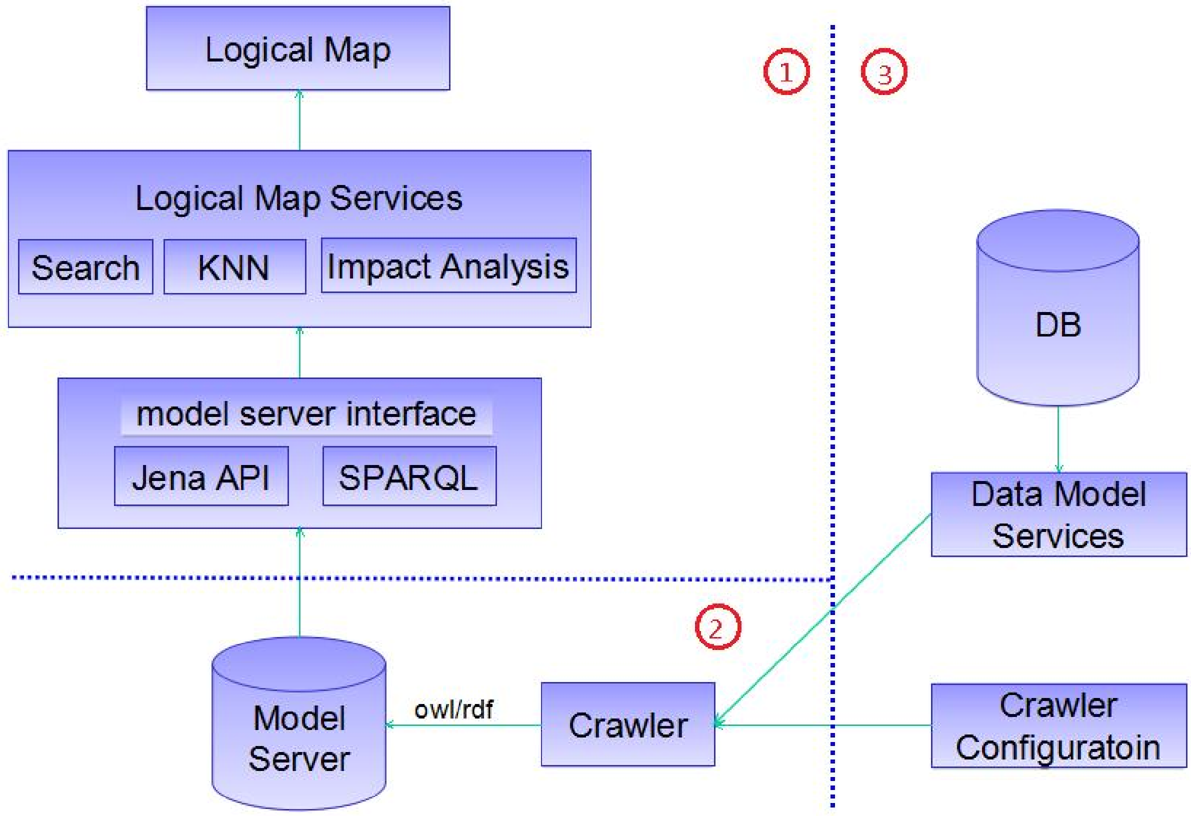


Figure 34. The framework of a logical map

Here is sample data generated by the crawler. In this example the OWL defines 2 kinds of instance types:

- `substation` and asset.
- `Current_Risk` a measurement type.

The OWL also defines two relationships:

- `Asset.hasMeasurement` references the measurement belonging to the asset.
- `Measurement.associatedToAsset` references the asset belonging to the measurement.

The two relationships are an inverse of each other, therefore in the rdf file, only one relationship needs to be defined. After execution of `inference.sh` from the Jena installation path the other relationship is automatically generated.

```
<rdf:Description rdf:about="http://ontology#substation">
  <rdf:type rdf:resource="http://www.w3.org/2002/07//owl#class"/>
  <rdfs:lable xml:lang="en">Substation</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://ontology#asset"/>
</red:Description>

<rdf:Description rdf:about="http://ontology#Asset.hasMeasurement">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:label xml:lang="en">Asset.hasMeasurement</rdfs#label>
  <owl:inverseOf rdf:resource="http://ontology#Measurement.associatedToAsset">
  <rdfs:subClassOf rdf:resource="http://ontology#Relationship"/>
</rdf:Description>

<rdf:Description rdf:about="http://ontology#Current_Risk">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:label xml:lang="en">Current_Risk</rdfs#label>
  <rdfs:subClassOf rdf:resource="http://ontology#Measurement"/>
  <rdfs:comment>Measurement</rdfs:comment>
</rdf:Description>

<rdf:Description rdf:about="http://ontology#Measurement.associationToAsset">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
```

```
  <rdfs:label xml:lang="en">Measurement.associationToAsset</rdfs#label>
  <owl:inverseOf rdf:resource="http://ontology#Asset.hasMeasurement">
  <rdfs:subClassOf rdf:resource="http://ontology#Relationship"/>
</rdf:Description>
```

The example rdf file defines that `Current_Risk_944` is a measurement `Measurement.associatedToAsset` of `substation_2596`. After execution of the `inference.sh` then `substation_2596` has the relationship `Asset.hasMeasurement` and references `Current_Risk_944`.

The rdf file defines two instances:

- substation_2596
- Current_Risk_944

These two instance have the relationship `Measurement.associatedToAsset`. The `Current_Risk_944` is the measurement of the asset `substation_2596`.

Note: The rdf:resource should refer to an existing id defined in rdf:ID, e.g. rdf:resource of Current_Risk_944 refers to the rdf:ID of substation_2596.

```
<otl:substation rdf:ID="substation_2596">
  <otl:Object.name>Anytown_North</otl:Object.name>
  <rdfs:label xml:lang="en">Anytown_North</rdfs:label>
  <otl:Object.item.Url>ibm/ife/sample/dno/api/substation/2596</otl:Object.item.Url>
</otl:substation>

<otl:Current_Risk rdf:ID="Current_Risk_944">
  <otl:Object.name>Current Risk</otl:Object.name>
  <rdfs:label xml:lang="en">Current Risk</rdfs:label>
  <otl:Object_description>Current Risk</otl:Object.description>
  <otl:Object.item.Url>ibm/ife/sample/dno/api/measurement/944</otl:Object.item.Url>
  <otl:Measurement.accosicatedToAsset rdf:resource="#substation_2596"/>
</otl:Current_Risk>
```

The example rdf file has the crawler configuration necessary to generate the OWL and RDF files as follows:

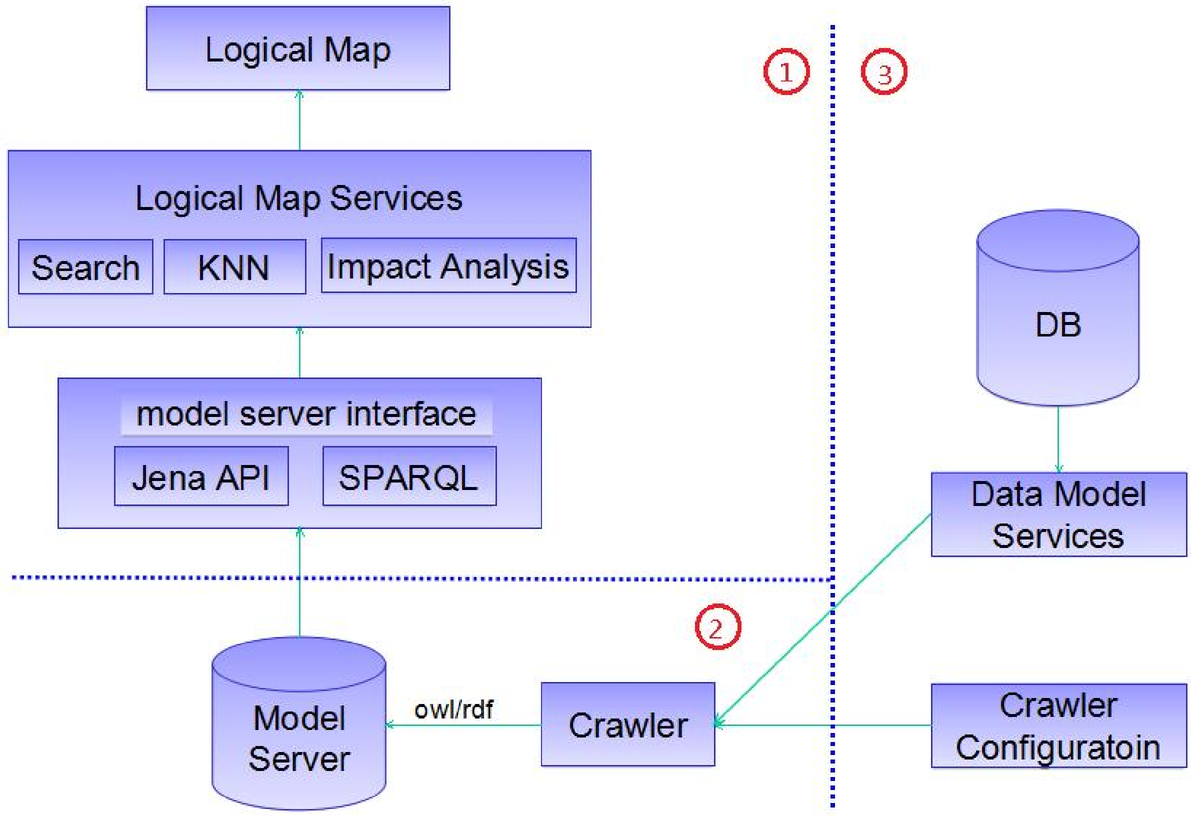


Figure 35. Crawler configuration ID:substation

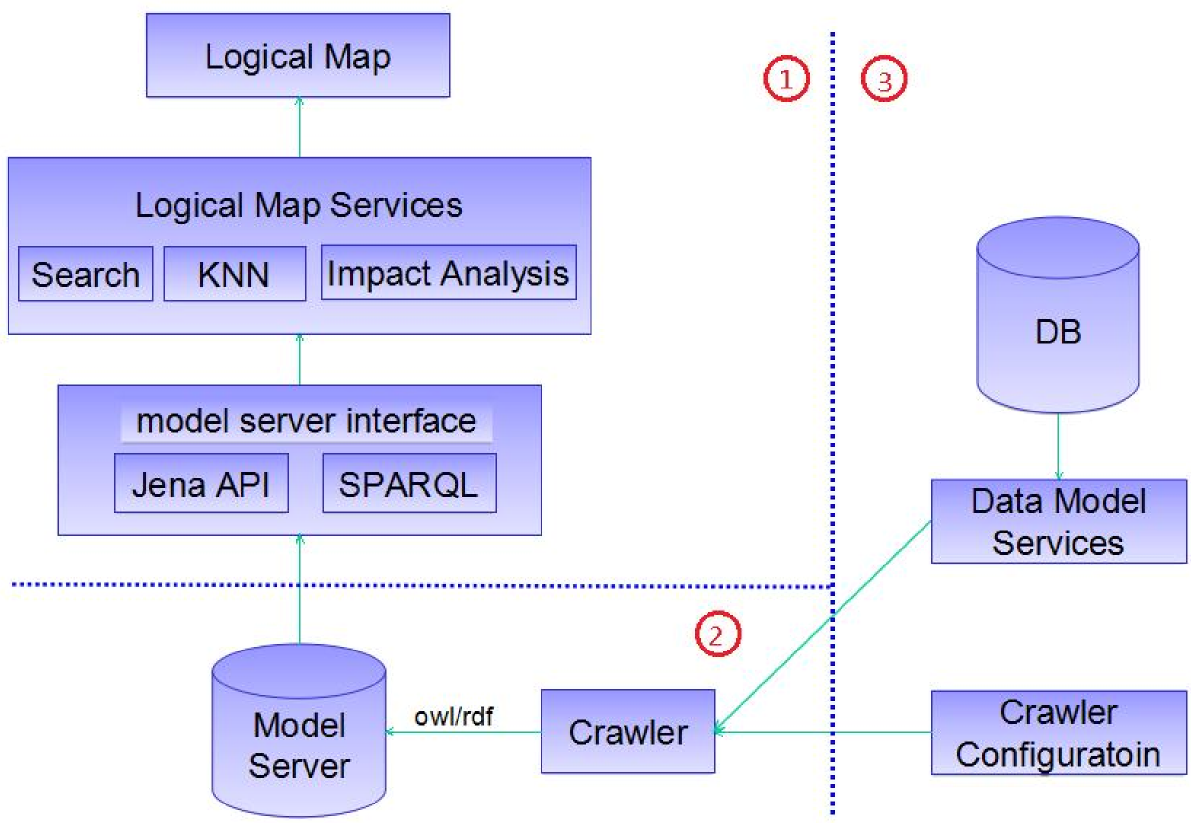


Figure 36. Crawler configuration ID:Current_Risk

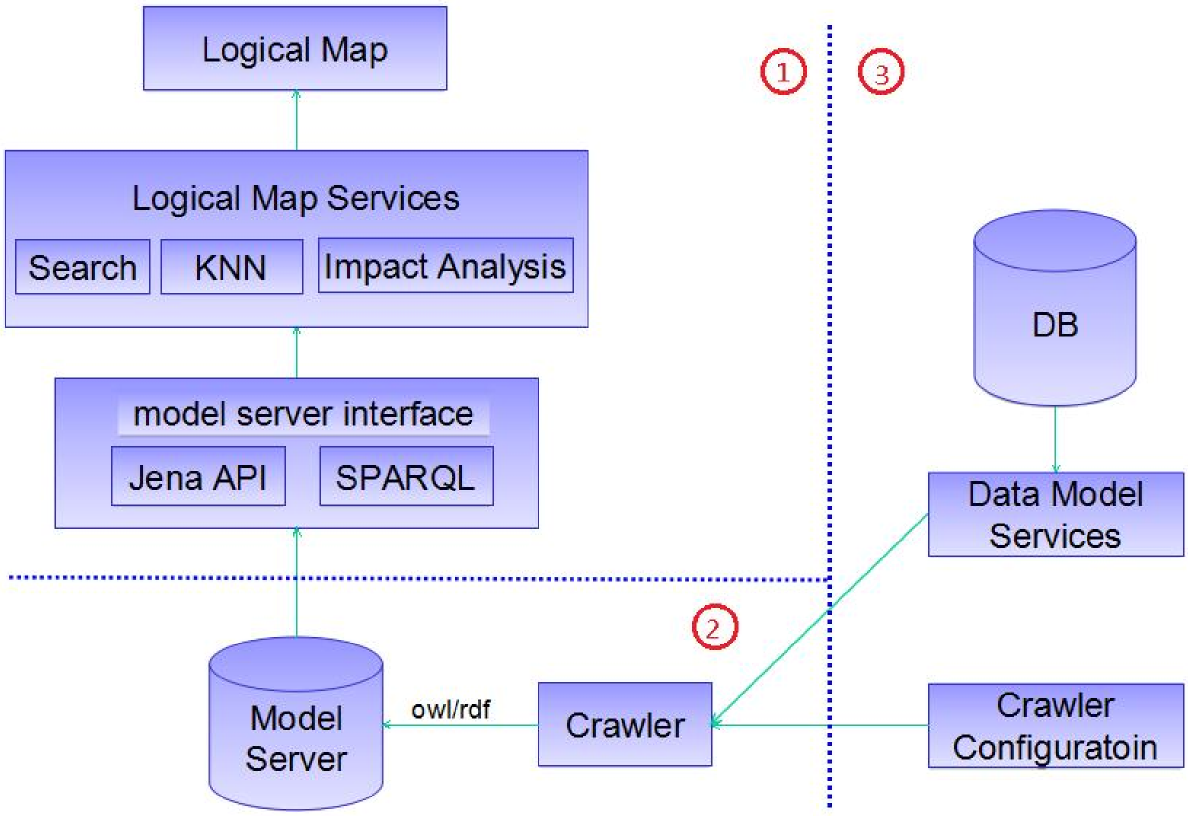


Figure 37. Crawler configuration ID:Measurement_associatedToAsset

Note: Explanatory notes:

- The red rectangle ID:"substation" should be the same value as `itermsUrl.asset_type`, otherwise the `rdf:resource` will refer to a none existing `rdf:ID`. The result in the logical map will not be correct.
- The values for `id`, `value`, `subClassOf`, `owl_template`, and `rdf_template` can all be edited.
- The value for `itemsUrl` is the service URL and is mandatory. It represents the result returned from the service.
- The `label` is used to search and is mandatory.
- The `owl_template` is the content of this type in the owl file.
- The `rdf_template` is the content of a instance of this type in the rdf file.
- The variable `@{…}` is changed to a real value.

The Service.properties file at the location: /opt/IBM/energy/crawler defines some of the properties used in the crawler.

Table 19. Type and parameter mapping

Property	Description
serviceBaseURL=http://ip:port	The service URL.
serviceUser=sysadmin	Service username.
servicePassword=passw0rd	Service password.
CLIENT_READ_TIMEOUT=0	Read_timeout.
CLIENT_CONNECT_TIMEOUT=0	Connect_timeout.

Table 19. Type and parameter mapping (continued)	
Property	**Description**
owlfile=d:/newfile.owl	New OWL file.
rdffile=d:/newrdffile.rdf	New rdf file.

ifef.widget.LogicalMap Property

The property for the logical map is:

layoutData:

The data that is used to draw in the logical map.

ifef.widget.LogicalMap Parameters

Th parameters for th logical map are:

keyFields:

Used to identify a record, json object, coming from a call of the model.

namespace:

The namespace of the data as an OWL/rdf file that is generated by the crawler. The namespace is defined in the configuration file of the crawler.

knnDepth:

The depth for the KNN service, K-NearestNeighbor.

impactAnalysisDepth:

The depth of the impact analysis service. It can be temporarily changed by the configuration dialog of the logical map.

instCfg:

The configuration for the instance. It can be temporarily changed by the configuration dialog of the logical map. The user can add or delete items to add or remove one kind of instance. The instType of newly added item here must be generated in OWL files and already imported to model server.

For example: Add one instance type:

```
{"instType":"undergroundline", "canBeOnMap":true, "checked":false, "label":"Underground
        Line", "instStyle":"/ibm/ife/logicalmap/test/logicalmap/icon/undergroundline.png"}
```

The instance type is defined in the configuration file of crawler:

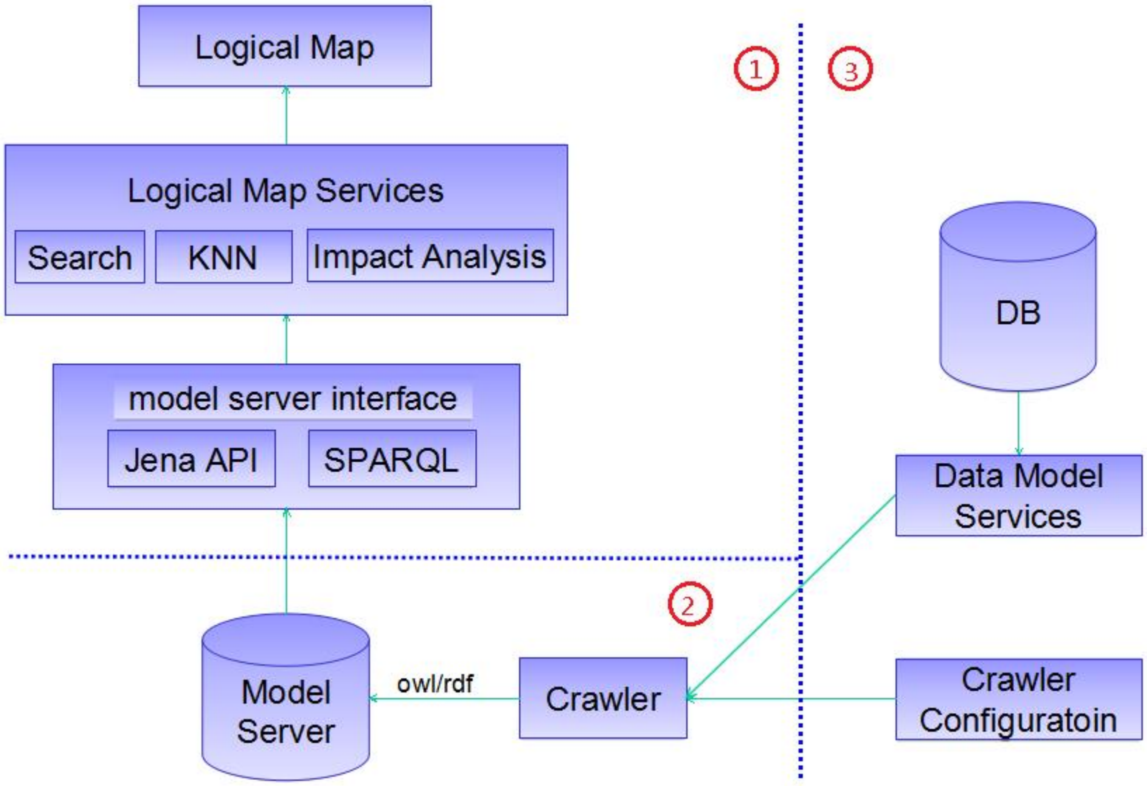


Figure 38. Configuration file of the crawler

- `canBeOnMap` - the instance can be displayed on map. If true, then there must be a “`Highlight On Map`” action for this instance.
- `checked` - Determines if this instance type is used to filter the search result.
- `instStyle` - The icon used for this instance type
- `label` - The text display in configuration dialog.

relationshipCfg:

The configuration for the relationship. It can be temporarily changed by the configuration dialog of logical map. The user can add or delete item here to add or remove one kind of relationship. The relationship must be generated in the OWL file and already be imported to the model server.

For example: Add one relationship.

```
{"relationship":"http://ontology#Asset.NewRelationship", "checked":true, "label":"New
Relationship", "relationshipStyle":"#ffa500"}
```

- `relationship` - the relationship defined in the OWL file.
- `checked` - determines if this relationship is used to filter the search result.
- `relationsihpStyle` - the color for this relationship. If the value is not in range of ColorPalette, then black(#000000) is used.
- `label` - the text displayed in the configuration dialog.

An sample configuration

```
{
                "id": "logicalMap",
                "module": "lmap/widget/logicalmap/LogicalMap",
                "container": "logicalMapTab",
                "properties":["layoutData"],
                "parameters": {
        "keyFields":
```

```
        ["ASSET_TYPE",
        "ASSET_ID"],
        "namespace":"http://ontology#",
        "knnDepth":1,
        "impactAnalysisDepth":6,
        "instCfg":[{"instType":"substation",
        "canBeOnMap":true, "checked":true,
"label":"@{nls.CfgDlgInstSubstation}",
"instStyle":"/ibm/ife/logicalmap/test/logicalmap/icon/substation.png"},
                   {"instType":"overheadline", "canBeOnMap":true, "checked":true,
"label":"@{nls.CfgDlgInstOverheadline}",
"instStyle":"/ibm/ife/logicalmap/test/logicalmap/icon/overheadline.png"},
                   {"instType":"wind_farm", "canBeOnMap":true, "checked":false,
"label":"@{nls.CfgDlgInstWindFarm}",
"instStyle":"/ibm/ife/logicalmap/test/logicalmap/icon/wind_farm.png"},
                   {"instType":"Measurement", "canBeOnMap":false, "checked":false,
"label":"@{nls.CfgDlgInstMeasurement}",
"instStyle":"/ibm/ife/logicalmap/test/logicalmap/icon/measurement.png"},
                   {"instType":"Default", "instStyle":"/ibm/ife/logicalmap/test/logicalmap/
icon/measurement.png"}
            ],
        "relationshipCfg":[{"relationship":"http://ontology#Asset.Connects",
"checked":true, "label":"@{nls.CfgDlgRealConnects}",
        "relationshipStyle":"#7fff00"},                            {"relationship":"http://
ontology#Asset.Connected",
        "checked":true, "label":"@{nls.CfgDlgRealConnected}",
"relationshipStyle":"#ffa500"},
                           {"relationship":"http://ontology#Asset.Contains",
        "checked":false, "label":"@{nls.CfgDlgRealContains}",
"relationshipStyle":"#6495ed"},
                           {"relationship":"http://ontology#Asset.Contained",
        "checked":false, "label":"@{nls.CfgDlgRealContained}",
"relationshipStyle":"#7b68ee"},
                           {"relationship":"http://ontology#Asset.hasMeasurement",
        "checked":true, "label":"@{nls.CfgDlgRealHasMeasurement}",
"relationshipStyle":"#f08080"},
                    {"relationship":"http://ontology#Measurement.associatedToAsset",
        "checked":true, "label":"@{nls.CfgDlgRealAssociatedToAsset}",
        "relationshipStyle":"#ff4500"},
                    {"relationship":"http://ontology#Asset.hasPrimaryMeasurement",
        "checked":true, "label":"@{nls.CfgDlgRealHasPrimaryMeasurement}",
"relationshipStyle":"#b22222"},
                    {"relationship":"http://
ontology#Measurement.Is_Primary_Measurement_Of",
        "checked":true, "label":"@{nls.CfgDlgRealIsPrimaryMeasurementOf}",
"relationshipStyle":"#ffd700"},
                    {"relationship":"http://www.w3.org/2000/01/rdf-schema#subClassOf",
        "checked":true, "label":"@{nls.CfgDlgRealSubClassOf}",
        "relationshipStyle":"#ffe4c4"},
                    {"relationship":"http://www.w3.org/1999/02/22-rdf-syntax-ns#type",
        "checked":true, "label":"@{nls.CfgDlgRealType}",
        "relationshipStyle":"#da70d6"}        ]                 }
    }
```

Related configuration for the logical map

There are three other configurations to support the logical map:

1. Logical map layouts - used to draw logical maps with different layouts.
   - Property -

     `rawLayoutData:` the data used to draw the logical map.
   - Parameter -

     The id of logical map.

   Configuration sample:

   ```
   {
   "id": "logicalMapLayouts",
   "module": "lmap/behavior/LogicalMapLayouts",
   "properties":["rawLayoutData"],
   "parameters": { "logicalMap":"@{logicalMap}" } }
   ```
2. Logical map previewcard helper - used as the helper class for more actions of logical map previewcard.

- Parameters:
  - logicalMap: the ID of logical map.
  - highlightOnMap: the common highlight on the map support.

  Configuration sample:

```
{
                "id":"logicMapPreviewCardHelper",
                "module":"lmap/widget/logicalmap/LogicMapPreviewCardHelper",
                "parameters":{
                                "logicalMap":"@{logicalMap}",
                                "highlightOnmap":"@{highlightOnMap.highlightOnMap}"
                }
    }
```

3. Logical map previewcard - Refer to "PreviewCard widget" on page 91 for information about the parameters.

   The property "NAME" is required to display the relationship or instance information without an item URL, for example: Instance type:substation, in the previewcard. For example: `{"label":"@{dno_nls.NAME}","name":"NAME", "isKey": true}`.

   A sample configuration is given:

```
"id":"logicalMapPreviewCard",
                "module": "ifef/widget/previewcard/PreviewCard",
                "properties":["data"],
                "parameters": {
                      "titleProperty": "NAME",
                      "position": "_position",
                      "properties":[
                           {"label":"@{dno_nls.Measurement_ID}", "name":"MEASUREMENT_ID",
"isKey": true},
                           {"label":"@{dno_nls.Asset_ID}", "name":"ASSET_ID", "isKey": true},
                           {"label":"@{dno_nls.FARM_ID}", "name":"FARM_ID", "isKey": true},
                           {"label":"@{dno_nls.Measurement_Type}",
"name":"MEASUREMENT_TYPE", "isKey": true},
                           {"label":"@{dno_nls.ID}","name":"ID", "isKey": true},
{"label":"@{dno_nls.NAME}","name":"NAME", "isKey": true},
                           {"label":"@{dno_nls.Serial_Number}","name":"SERIAL_NUMBER",
"isKey": true},
                           {"label":"@{dno_nls.UNIT}","name":"UNIT", "isKey": true},
                           {"label":"@{dno_nls.Description}","name":"DESCRIPTION", "isKey":
true},
                           {"label":"@{dno_nls.Create_Date}","name":"CREATE_DATE"},

{"label":"@{dno_nls.Installation_Date}","name":"INSTALLATION_DATE"},
                           {"label":"@{dno_nls.STATUS}",
                      "name":"STATUS",
                      "isKey": true,
                      "render":{
                           "templates":[
                                {
                                     "condition":"STATUS==0",
                                     "content":"<span
style='background-color:green'>${Acceptable}</
span>",

                                     "variables":{
                                          "Acceptable":"@{dno_nls.Acceptable}"
                                     }
                                },
                                {
                                     "condition":"STATUS==1",
                                     "content":"<span
style='background-color:orange'>${Caution}</span>",
                                     "variables":{
                                          "Caution":"@{dno_nls.Caution"}
                                     }
                                 },

                                 {
                                      "condition":"STATUS==2",
                                      "content":"<span
style='background-color:red'>${Critical}</span>",
                                      "variables":{
```

```
                                                    "Critical":"@{dno_nls.Critical}"
                                               }
                                           }
                                      ]
                                 }
                            },
                            {"label":"@{dno_nls.Location}","name":"LOCATION"},
                       ]"
                  "moreActions":[
             ]
             }
        }
```

List Widget

The List widget is the wrapper for the idx gridx widget. It provides the capability for a developer to create and configure list and column customizations.

ifef.widget.List Properties

store:

The data store that acts as the bridge to the REST service.

selected:

When a property is selected from a list, it is populated with the corresponding JSON object.

criteria:

A JSON object that describes the filter criteria. This property will be populated only if the `keyFields` parameter is specified (see below) and is updated when the selected property is changed.

ifef.widget.List Parameters

list level:

- `pageSize:` The initial size of the page, the number of the items that will be displayed on the initial page. For example: `"pageSize": 10` will show 10 items on the initial page.
- `paginationBar:` the pagination for the list. The quantity of items that show on each page. A value of 150 will display the next 150 items on that page and also on subsequent pages. The last value 0 displays all the items in the data store.
- `baseSort:` The sort sequence for the columns, ascend or descend. Specify descending as true or false to determine the sort order of the column.

```
Example:
"baseSort": [
        {
                "attribute": "ASSET_ID",
                "descending": true
},
{
                "attribute": "NAME",
                "descending": true
}
]
```

- `keyFields:` The fields used to generate the criteria and selectors based on the selected row in the list.

column level:

- `field:` The column field. The field that matches the field in the store that is queried from the database.
- `name:` The name for the column that will be displayed in the list. For example `"name": "ASSET_STATUS"`

  Note:

  The user can use the strings in the `nls` file to support globalization , for example:

"name": "@{dno_nls.STATUS}".

- sortable: Determines if the column can be sorted or not, true or false.

  For example: "sortable": true, mean that the column can be sorted.
- width: the width of the column in pixels. For example: "width": "80px".
- formatter: the format of the cell data for time or date.
  - type: specify the two data type that you want to format , 'number' or 'date'.

    number

    When number, you must also specify the parameter "places", this is the number of fixed significant digits without specifying the dot as a placement. For example:

    ```
    "formatter":
    {
         "type": "number",
         "places":2
    }
    ```

    date

  When date, you must also specify the parameter "formatLength". The available values are: long, short, medium and full. For example:

  ```
  "formatter":
  {
       "type": "date",
       "formatLength":"short"
  }
  ```
- style: the style for the cells. The value for this style can be a function that returns a style string, or simply be a style string for this column. For example:"style" :"background-color:#EE3D96;color:red" or "style" : "@{statusFormatter.style}".
- decorator: the function to decorate the raw data.

Configuration example

```
{
                "id": "measurementList",
                "module": "ifef/widget/list/List",
                "container": "measurementPane",
                "properties": ["store", "selected", "criteria", "selector"],
                "parameters":{
                                "selectorFields" : ["MEASUREMENT_TYPE"],
                                "keyFields" : ["MEASUREMENT_ID"],
                                "pageSize": 10,
                                "paginationBar":[10,50,100,0],
                                "baseSort": [
                                            {
                                                 "attribute":
"STATUS",

                                                 "descending": false
                                            },
                                            {

                                                 "attribute":
"NAME",

                                                 "descending":
true
                                            }

                                            ],
                                "column": [
                                          {
                                                 "name":
"@{dno_nls.STATUS}",

                                                 "field": "STATUS",
                                                 "sortable" :
```

```
true,
                                                  "style" :
"@{statusFormatter.style}",

                                                  "decorator" :
"@{statusFormatter.decorator}"


                                            },


{
                                                  "name": "@{dno_nls.Measurement_ID}",
                                                  "field": "MEASUREMENT_ID",
                                                  "sortable" : true
                                            },


{
                                                 "name": "@{dno_nls.NAME}",
                                                 "field": "NAME",
                                                 "sortable" : true
                                             },
                                             {
                                                  "name":
"@{dno_nls.UNIT}",


                                                  "field": "UNIT",
                                                  "sortable" : false
                                             }
          ]
     }
}
```

List Container Widget

The user can place one list directly in one page. However, where there are multiple lists that can change based on the selection made by the user, it is better to put all the lists in one List Container.

In the List Container widget you can specify the container for the lists. This is configured in a widget containment JSON file.

ifef.widget.ListContainer Parameters

selectLabel:

The label used for the drop down list.

style:

The cascading style sheet (CSS) used to define the display style.

Configuration sample

```
{
        "id": "listContainer",
        "module": "ifef/widget/list/ListContainer",
        "container": "listTab",
        "parameters": {
                        "selectLabel":"@{dno_nls.Select_Item}",
                        "style": "width:
        100%; height: 100%;"
        }
     },
     {
        "id": "overheadlineList",
        "module": "ifef/widget/list/List",
        "container": "listContainer",
           "properties": ["store"],
        "parameters": {
                       "keyFields":["ASSET_TYPE", "ASSET_ID"],
                       "title": "@{overheadline_filter.label}",
                       "pageSize": 100,
                       "paginationBar":[100,150,200,1000,0],
                       "baseSort": [
                           {
                            "attribute":
"STATUS",
```

```
                              "descending": true
                             }
                             ],
                        "column": [
                             {
                              "name":
"@{dno_nls.STATUS}",
                              "field": "STATUS"
                             },

{
                              "name":
"@{dno_nls.NAME}",
                              "field":
"NAME",
                              "sortable" : true
                             }
                      ]
     }
}
```

Line Chart Widget

The LineChart widget is used to show changes to a series of values over time.

The Line Chart can be enlarged or made smaller by zooming in or out, and the time period can be changed by dragging the chart to show different time periods. The line chart is used to show changes of one measurement reading over time for a specified asset.

ifef.widget.LineChart Parameters

title:

String, the title of the chart.

chartWidth:

Number, specifies the width of the chart.

chartHeight:

Number, specifies the height of the chart.

series:

Array, describes the number of lines and name for the specified series.

- `displayName:` String, the display name for the specified series.
- `property:` String, The field name for each of the series of the Y axis data. The field name must be a number.

X_property:

String, The field name for each time period used for the X axis. The field name must be a timestamp or period.

X_title:

The display name for the X axis.

Y_title:

The display name for the Y axis.

The properties for the Line Chart widget

store:

The data for line chart should be bound to the property `store` of the `LineChart` instance. This property changes when the user selects different measurement criteria.

Configuration sample

The configuration sample shows two series of data for the `displayName` `VALUE1` and `VALUE2`.

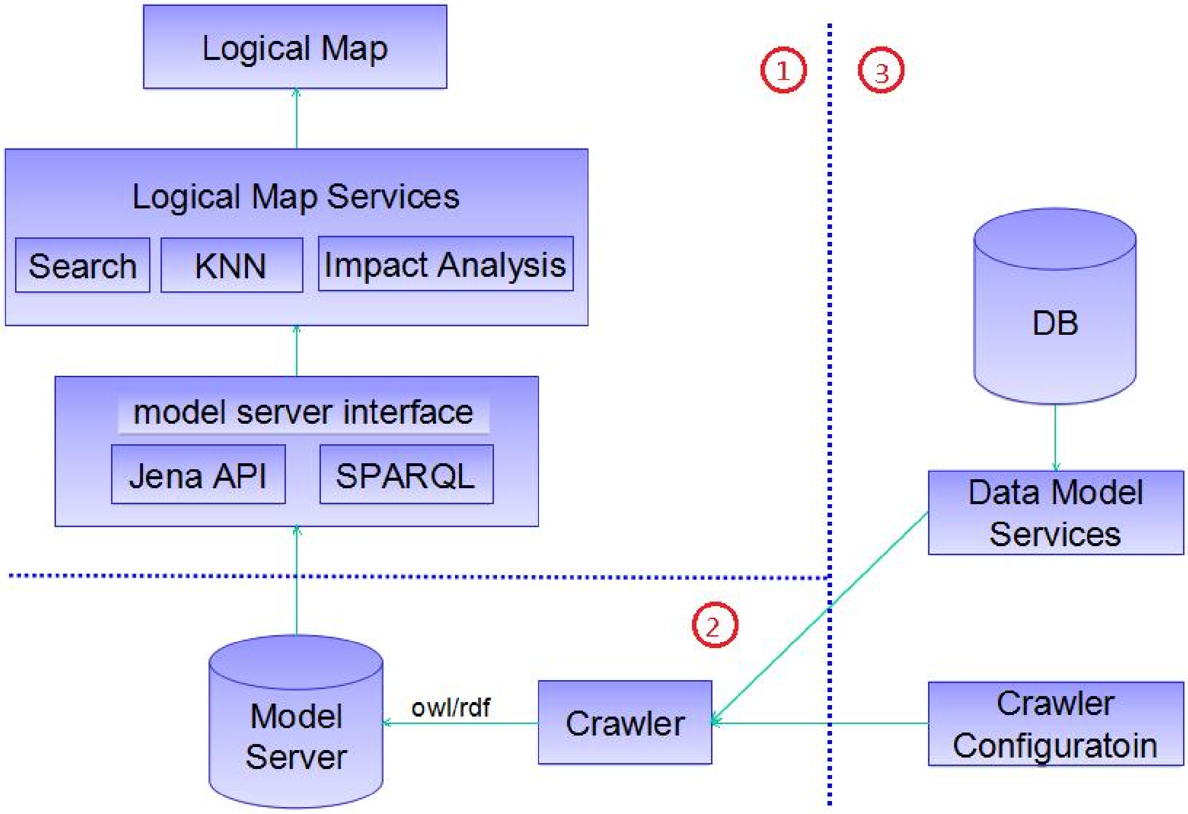


Figure 39. Example line chart

```
{
       "id": "readingLineChart",
       "module": "ifef/widget/chart/LineChart",
       "container": "readingLineChartTab",
       "properties": ["store"],
       "parameters": {
              "title": "@{dno_nls.Reading_Line_Chart}",
              "chartWidth":450,
              "chartHeight":200,
              "series": [{
                     "displayName": "@{dno_nls.VALUE1}",
                     "property": "VALUE1"
              },{
                     "displayName": "@{dno_nls.VALUE2}",
                     "property": "VALUE2"
              }],
              "X_property": "TIME",
              "X_title": "@{dno_nls.TIME}",
              "Y_title": "@{dno_nls.VALUE}"
       }
}
```

The data for the Line Chart should be provided as a JSON Array that must have a field that contains the timestamp value. The other fields which contain number value can be shown as a series of lines.

```
[{
       "id": 1,
       "TIME": "2013-01-01 08:58:00",
       "VALUE": 4
},{
       "id": 2,
       "TIME": "2014-01-01 08:59:00",
       "VALUE": 4
},{
```

```
        "id": 3,
        "TIME": "2015-01-01 09:00:00",
        "VALUE": 5
}]
```

Bar Chart widget

The BarChart widget is used to show a group of related values that can be grouped as a hierarchical relationship.

As a time-series value, a bar chart widget can be grouped to show year, Month, Day, Hour, or Minute.

As a population series a bar chart can be grouped to show population by Country, State, City, Town, or Region.

You can click on the bar chart to drill up and drill down the hierarchical levels.

The example bar chart is used to show measurements readings for maximum, minimum and average values for a specified asset. The bar chart is grouped by Year, Month, Day, Hour and Minute.

ifef.widget.LineChart Parameters

title:
String, the title of the chart.

chartWidth:

Number, specifies the width of the chart.

chartHeight:

Number, specifies the height of the chart.

legends:

Array, specifies the number of bars for each of the groups.

- `displayName:` String, the display name for the specified series of bars.
- `property:` String, The field name for each series of bars of Y axis data. The field name must be a number.

X_property:

String, The field name for each time period used for the X axis. The field name must be a number.

X_title:

The display name for the X axis.

defaultLevelIndex:

Number, specifies the level of the hierarchy that shows when the bar chart opens.

levels:

Array, describes the number of levels of the bar chart can be drilled down or up, and also the number of hierarchy levels of the data in the chart.

- `index:` String, the index number of the current level.
- `X_title:` String, The display name of the X axis.
- `modelSelector:` Widget instance, specifies which model selector instance is used for the current level. Model Selector is a type of REST store map that provides the specified store for the widget by a key. An example is as below:

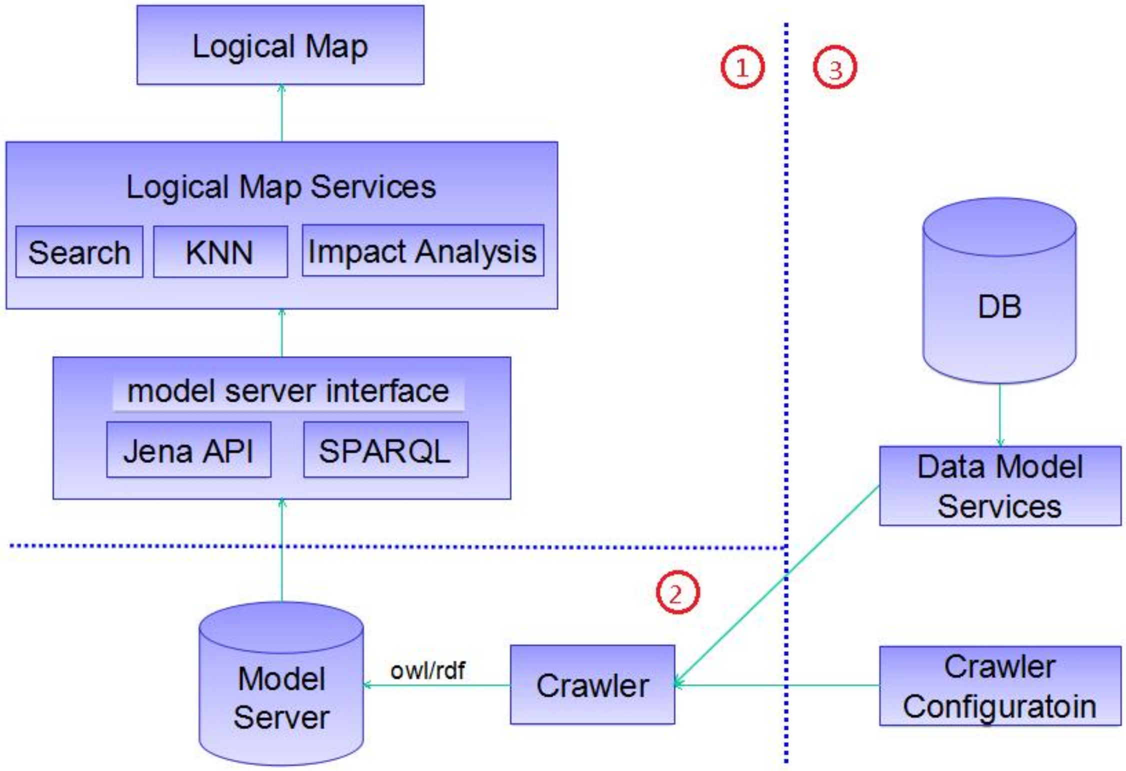


Figure 40. Model Selector instance for the current level of hierarchy

- `params:` Array, the fields that describe the hierarchy relationship of each level.
  - `displayName:` String, the display name for this field that shows as the drill up title.
  - `property:` String, specifies the field name of the data that is used to drill down or up and filter the data for Y axis.
  - `value:` Number, shows the default value for this field when the current level is the default level.

ifef.widget.BarChart Properties

store:

The data for the bar chart that should bound to the property "store" of the bar instance. This property changes when the user selects different measurement criteria.

Configuration sample

```
{
      "id": "readingBarChart",
      "module": "ifef/widget/chart/BarChart",
      "container": "statisticsBarChartTab",
      "properties": ["store","chartListColumns","chartListData"],
      "parameters":
                {
            "title": "@{dno_nls.Reading_Bar_Chart}",
            "chartWidth":
                450,            "chartHeight":
                200,            "legends":
                [{
                  "displayName": "@{nls.Chart_Label_MAX}",
                  "property": "MAX"
            },{
                  "displayName": "@{nls.Chart_Label_MIN}",
                  "property": "MIN"
            },{
                  "displayName": "@{nls.Chart_Label_AVG}",
                  "property": "AVG"
            }],
```

```
            "X_property": "TIME",
            "Y_title": "@{dno_nls.VALUE}",
            "defaultLevelIndex":0,
            "levels": [{
                  "index": "0",
                  "X_title": "@{nls.Chart_Label_YEAR}",
                  "modelSelector": "@{readingByYearSelector}",
                  "params": []
            },{
                  "index": "1",
                  "X_title": "@{nls.Chart_Label_MONTH}",
                  "modelSelector": "@{readingByMonthSelector}",
                  "params": [{
                      "displayName": "@{nls.Chart_Label_YEAR}",
                      "property": "YEAR",
                      "value": "2013"
                 }]
            },{
                  "index": "2",
                  "X_title": "@{nls.Chart_Label_DAY}",
                  "modelSelector": "@{readingByDaySelector}",
                  "params": [{
                       "displayName": "@{nls.Chart_Label_YEAR}",
                       "property": "YEAR",
                       "value": "2013"
                  },{
                       "displayName": "@{nls.Chart_Label_MONTH}",
                       "property": "MONTH",
                       "value": "1"
                }]
            },{
                  "index": "3",
                  "X_title": "@{nls.Chart_Label_HOUR}",
                  "modelSelector": "@{readingByHourSelector}",
                  "params": [{
                       "displayName": "@{nls.Chart_Label_YEAR}",
                       "property": "YEAR",
                       "value": "2013"
                 },{
                       "displayName": "@{nls.Chart_Label_MONTH}",
                       "property": "MONTH",
                       "value": "1"
                 },{
                       "displayName": "@{nls.Chart_Label_DAY}",
                       "property": "DAY",
                       "value": "1"
                 }]
            },{
                  "index": "4",
                  "X_title": "@{nls.Chart_Label_MINUTE}",
                  "modelSelector": "@{readingByMinuteSelector}",
                  "params": [{
                       "displayName": "@{nls.Chart_Label_YEAR}",
                       "property": "YEAR",
                       "value": "2013"
                 },{
                       "displayName": "@{nls.Chart_Label_MONTH}",
                       "property": "MONTH",
                       "value": "1"
                 },{
                       "displayName": "@{nls.Chart_Label_DAY}",
                       "property": "DAY",
                       "value": "1"
                 },{
                       "displayName": "@{nls.Chart_Label_HOUR}",
                       "property": "HOUR",
                       "value": "1"
                 }]
            }]
      }
}
```

Data sample

The data samples show a time-series value for year as the top level and month and day as subsequent levels.

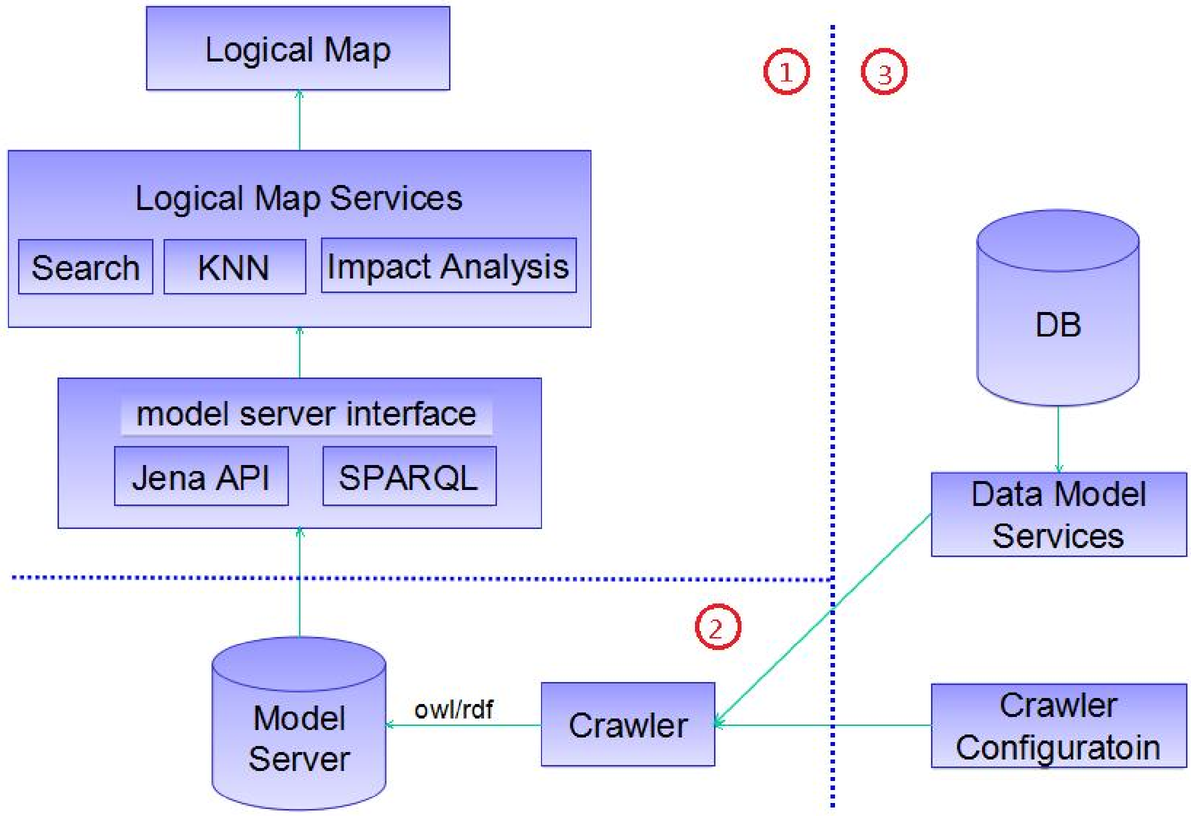


Figure 41. Hierarchy level: By Year

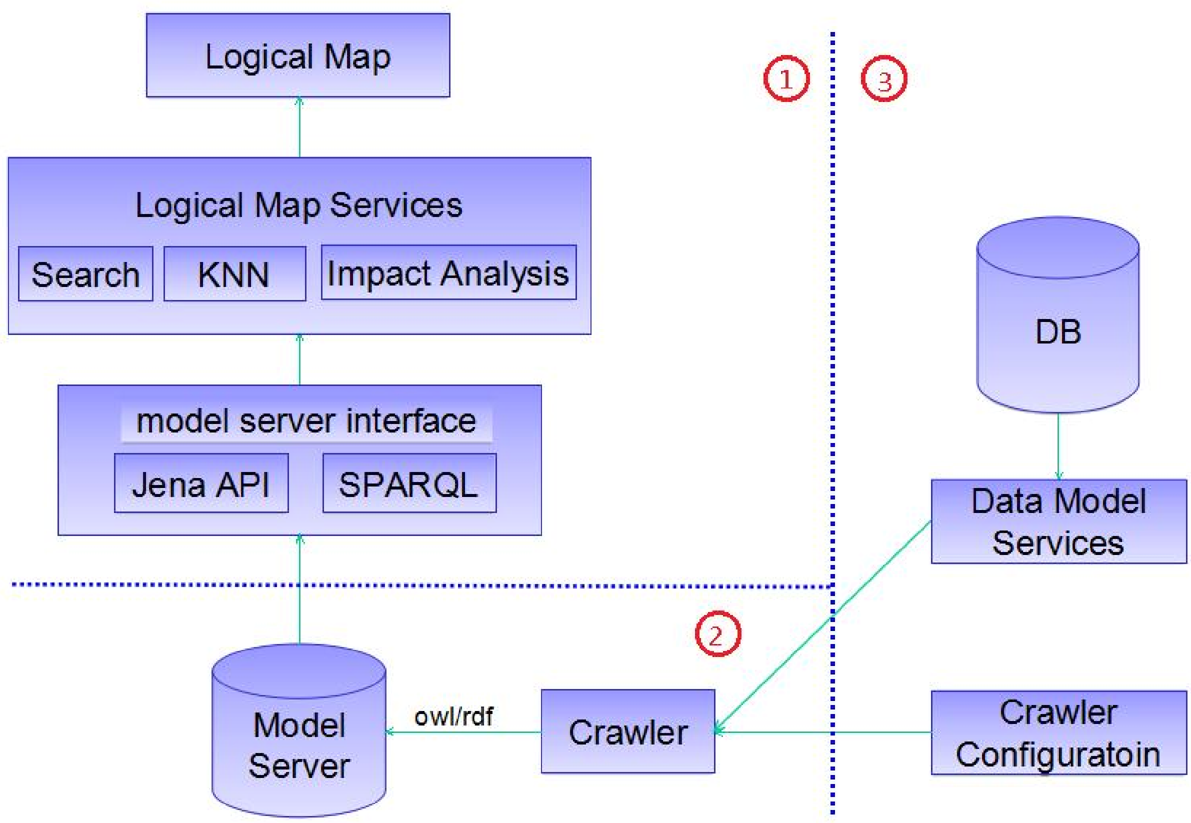


Figure 42. Hierarchy level: By Month

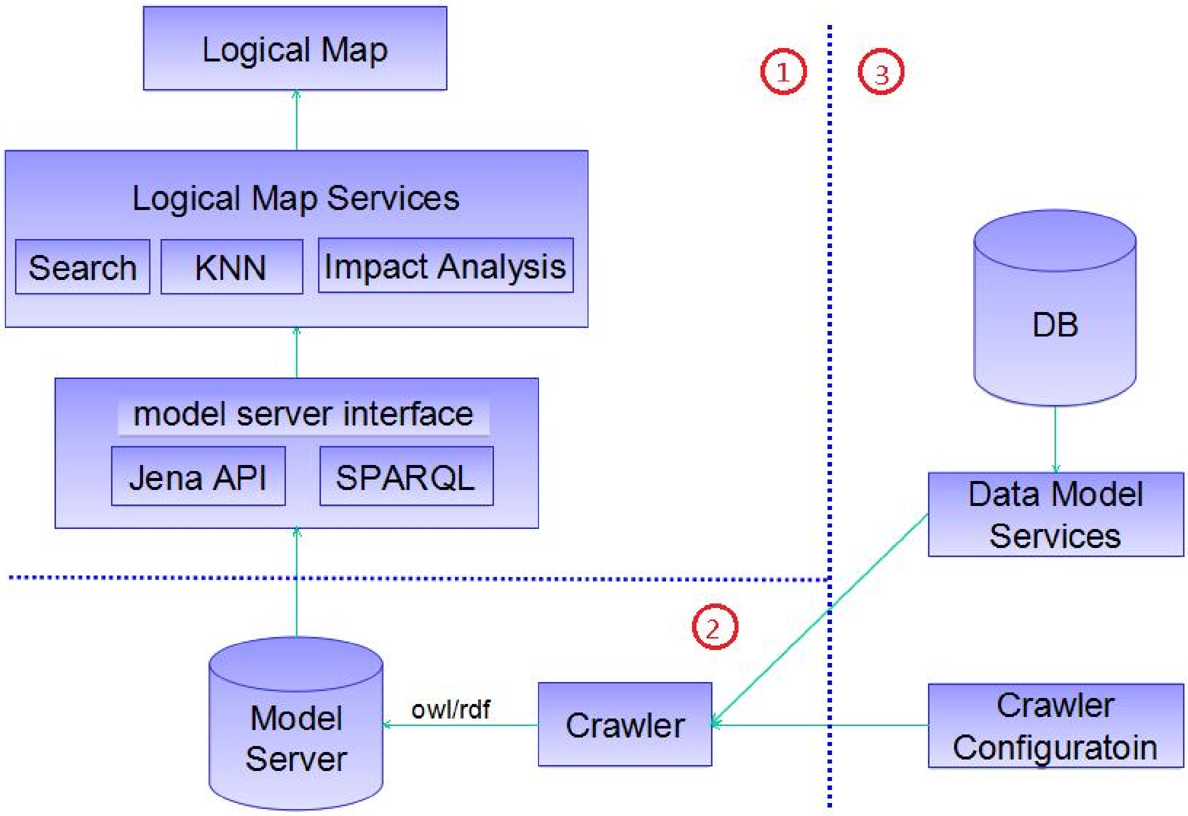


Figure 43. Hierarchy level: By Day

Header Button Widget

The Header Button widget provides the ability to show a pop-up dialog box that contains other widgets.

The Header Button is placed on the header bar.

ifef.widget.HeaderButton Parameters

label:

String, the label of the button.

icon:

The customized icon for the button to be shown for the label.

content:

The widget instance, specifies the widget that show in the pop-up dialog box when opened.

Properties

alertcount:

The alertcount is bound to the button, and shows as a number to the right of the label.

Configuration sample

In the configuration example, the icon can be customized, and the content changed to show the widgets that need to be included.

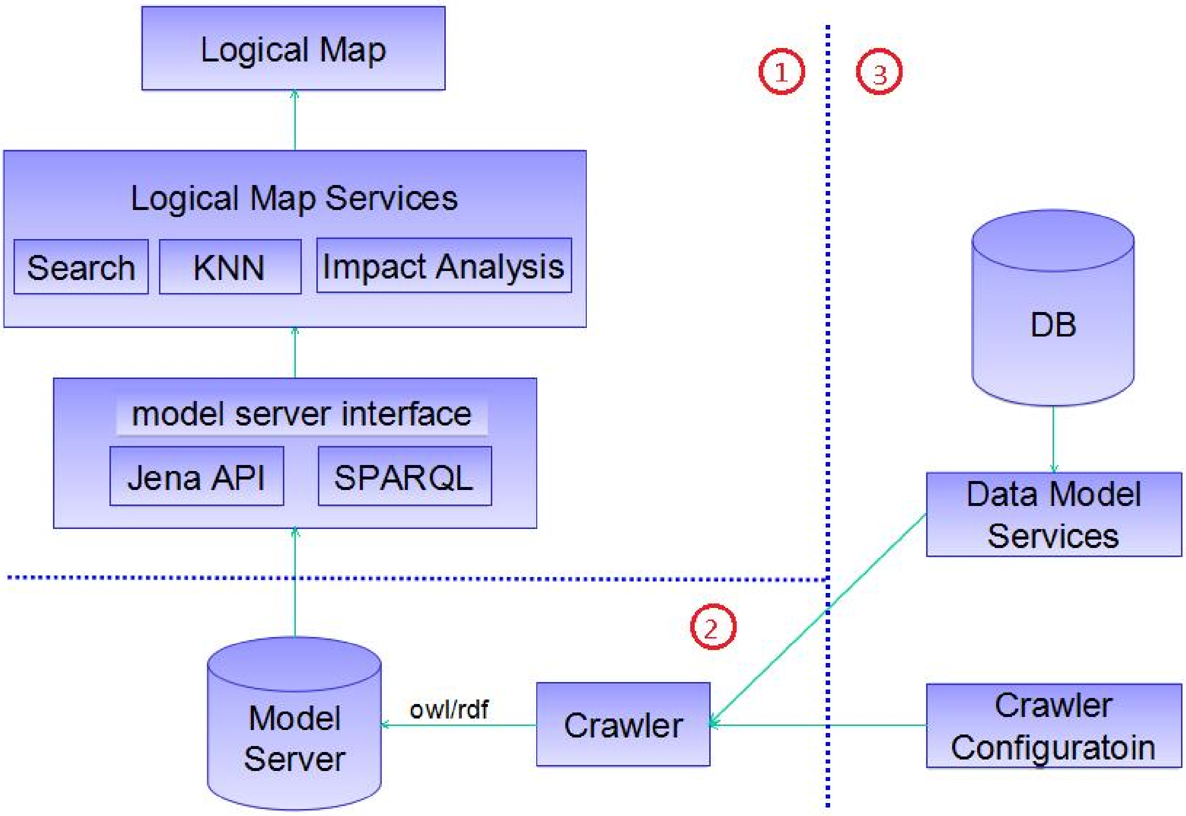


Figure 44. The header button showing the notifications button.

```
{
       "id": "notificationButton",
       "module": "ifef/widget/header/HeaderButton",
       "container": "header",
       "properties": ["alertcount"],
       "parameters":{
              "label": "Notification",
              "style": "float: right;",
              "buttonType": "info",
              "icon":
"http://findicons.com/icon/download/175313/notification_warning/16/png",
              "displayMode": "iconAndLabel",
              "content": "@{notificationContent}"
       }
}
```

Ready-to-use application template
An application template is a set of preconfigured configuration widgets that provide the basis of an application for the user to build upon. The template gives the user a start in building the application.

The REST service framework

This section introduces the mechanism and tasks to help the user to develop customized services. The REST service framework programming model provides an extension mechanism that lets the user to develop their own service quickly. These customized services can be used by the custom user interface (UI extension) for existing data.

The extension mechanism is a declarative programming model that lets the user to quickly create custom services. The services are shown in REST style and support general large-scale data handling. For example: pagination, filter by expression, and sorting.

Customizing the Asset Health data model

The application has several restrictions on the data model.

1. If you want to customize the existing SPSS models include the `*.str` file for the model and the `*.csv` file for the input for the model. Make a copy of the existing SPSS models and rename the new one. Create a new soft link for the commonly used models and then customize the new one. The reason: the upgrade of IBM Maximo Asset Performance Management for Energy & Utilities SaaS overwrites all the supported models.
2. IBM Maximo Asset Performance Management for Energy & Utilities SaaS provides reference files as the configuration files. These reference files can be customized for you own models.

   During an upgrade, the installer checks if the file exists without a template:

   - if the files exist without a template then do nothing,
   - if the files do not exist, then the template files are copied and renamed to remove template from the name. For example: The file `stream_model.template.cfg` is copied and renamed to `stream_model.cfg`. The file `stream_model.template.cfg` remains.

The Sample Application

The aim of the sample application is to help teach new developers how to incrementally develop the user interface based on UI framework provided by IFE.

There are two parts to the sample application:

- filter panel.
- corresponding map view.

The filter panel is used to control the data that is displayed in the map view. By enabling or disabling the filters, the map view changes accordingly.

Creating a UI Web project

About this task

You create a Dynamic Web Project called `ife_demo_web`.

Procedure

1. Start Eclipse.
2. Right-click the project explorer and select **Dynamic Web Project**.
3. In the **Project Name** field, type `ife_demo_web`.
4. In the **Target runtime** field, select **WebSphere Application Server V8.5 Liberty Profile**.
5. In the **Dynamic web module version** field, select **3.0**.
6. In the **Configuration** field, select **Default configuration for WebSphere Application Server V8.5 Liberty Profile**.
7. Click **Next**.
8. In the **Context root** field, type **ife_demo_web**.
9. In the **Content directory** field, type `WebContent` and enable the option: **Generate web.xml deployment descriptor**.
10. Click **Finish**.

Creating the main configuration and bootstrap files

You need to create the bootstrap file and place it into the WebContent directory of the project and create the configuration files to include different parts of the configuration items.

About this task

Different from traditional UI development, the UI Framework can parse JSON format configuration files dynamically at the runtime. The JSON format configuration contains the widget creation, containment, and the dependency relationship. The bootstrap file acts as the startup for the runtime.

Procedure

1. Create the bootstrap file `index.html` and save it to the `WebContent` directory for the project.
2. Create the configuration files to include the different configuration items, and save each one to the WebContent directory of the project.
   - `config.json` the entry configuration file.
   - `layout.json` the page layout configuration.
   - `map.json` the map view configuration.
   - `model.json` the model configuration.
   - `bind.json` the widget interaction configuration.

   The example shows the structure of the directory.

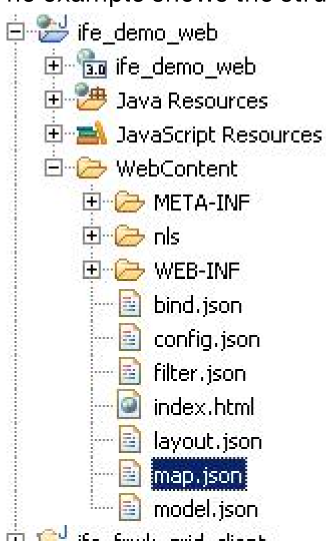


   Figure 45. The structure of the directory

3. Create the `UIMessage.js` message file in the `nls` folder for the supported localization languages.

   The example message file contains the localized language settings for French, Chinese, traditional Chinese, Japanese, and Brazilian Portuguese.

```
define({
 root : {
 Assets: "Assets",
 STATUS : "Status",
 NAME : "Name",
 UNIT : "Unit",
 ...
 },
 "fr" : true,
 "zh" : true,
 "zh-tw" : true,
 "ja" : true,
 "pt-br" : true
 });
```

For each enabled locale, there is corresponding sub-directory. In this example, the message files under `fr`, `zh`, `zh-tw`, `ja` and `pt-br` or `nls` will be created.

The result is:

Figure 46. The nls directory structure

Deploying the web application to the Websphere Liberty server

About this task

You need to deploy the ife_demo_web project to the Websphere Liberty server.

Procedure

1. Open the Server view, and right-click **Websphere Application Server V8.5 Liberty** from the local host and select **Add and Remove**.
2. Select **ife_demo_web** from the **Available** field and click **Add**.
3. Click **Finish**.

Launching the page

After you have deployed the sample application, you can launch the web page.

Procedure

1. Open your browser.
2. Type the URL as follows: https://<your ip>:<port>/ife_demo_web/index.html

   Where <your ip> and <port> are adjusted to your machine settings.

Chapter 6. Using Asset Performance Management application

The Asset Performance Management application shows how well a specific asset will provide its service in the future.

You have a direct visualization of the status of any asset from the reports and charts in the application. The Asset Performance Management application provides indices and numerical values that indicate the health and risk of asset and network failure. These reported values are:

- Health index
- Failure
- Criticality
- Risk

Health

Health is an index that is an aggregate score for the health of an asset and is calculated from the historical performance of the asset and the measured physical condition. The higher the value for the Health index, the less likely the asset will fail. Example factors that are used to calculate the heath index are age, manufacturer, and overload time. The Health index returns values in the range 0 to 100 where 100 is as new condition and 0 is very poor condition.

Failure

Failure is the probability that the network will fail. The higher the value, the more likely the network fails. Failure is calculated from the probability that an particular asset will fail and the impact that failure has on other assets in the network. Failure is calculated from four probabilities: The probability of failure of an individual asset. This is calculated as {(100 - Health index) * constant} The probability of failure of an asset downstream from the individual failing asset. The probability of failure of an asset upstream from the individual failing asset. The probability of the physical failure of a supporting asset. A supporting asset is one that gives physical support to the individual asset, for example, an overhead cable is physically supported by 2 poles. Failure returns values in the range 0 to 100 where 0 is no probability of failure and 100 is an imminent network failure.

Criticality

Criticality is a measure of the number of customers that are supported by an asset. Assets that support a greater number customers have a higher Criticality rating. As the number of downstream network nodes propagates, the Criticality rating of one asset is the summation of all Criticality rating of all downstream nodes, plus its own rating. Criticality is rated from 0 to 100 where 0 is no customers and 100 is all customers.

Risk

Risk is a measure of the risk to the business if a failure in the network occurs. The higher the value for the Risk the more risk there is to the business. Risk is a percentage value given by the product of the values for Failure and Criticality / 100. If Failure is 30 and Criticality is 65, then Risk is 19.5%.

If Criticality is much less in terms of customers, then if Failure is 30 and Criticality is 10, then Risk is 3%. The Risk returns values in the range 0 to 100 where 0 is no risk and 100 is a potential catastrophic risk.

Managing the custom analysis model

The **Custom Analysis Model** has the ability to download, upload, and delete a customized analysis model, manage the configuration of a model, and run an analysis for a configuration of a model.

Under the **Manage Model** tab you can upload your own customized SPSS streams and manage the folder hierarchy for a model.

Under the **Manage configurations** tab you can create parameters configurations for those SPSS models. One model can be used in more than one configuration using dofferent parameters..

Under the **Run new analysis** tab you can run an analysis for a particular configuration.

Prerequisites for an analysis

1. The data loading for Asset Performance Management must be complete.
2. The input parameters for the asset health models of each asset class are:
   - `Curve_Params_<asset>.csv` Defines the parameters for the degradation curve generation for the asset class.

     Note: Different subtypes can have different parameters. You can modify the parameter as necessary.
   - `AHI_Factor_<asset>.csv` Defines the parameters of factors that contributes to the asset health score.

     Note: Different subtypes can have different parameters. You can modify the parameter as necessary.
   - The files `AHI_Factor_<asset>.csv` and `Curve_Params_<asset>.csv` are in the location: `/opt/IBM/energy/AHI/SPSS_stream/data/ah_input`.
3. The required parameter configuration for analysis:
   - Is the configuration for the spss modeler server.

     The file includes configuration of:
     - The host name of server where the spss modeler runs
     - user name and password runs the modeler server
     - The location of the modeler batch.
   - `/opt/IBM/energy/AHI/SPSS_stream/conf/streamParams.cfg` Defines all the execution parameters that are needed when doing the analysis.

     Example content for the `streamParams.cfg` file.

```
dsname=IFEDB
dsuser=db2inst1
dspwd=db2inst1
-log /opt/IBM/energy/AHI/SPSS_stream/log/AHBacth.log
-appendlog
ana_year=20

# streams
[/opt/IBM/energy/AHI/SPSS_stream/stream/CircuitBreaker/CircuitBreaker_AHI.str]
csvFolder=/opt/IBM/energy/AHI/SPSS_stream/data/model_output
AHI_Factor_Weight=AHI_Factor_CircuitBreaker.csv
assetTable=CIM.CIRCUITBREAKER
asset_AHI_Factor_csv=AHI_factors_CB.csv
asset_AHI_csv=AHI_CB.csv
asset_Detail_csv=CircuitBreaker.csv
```

     Important: The first 6 lines are for common parameters:

     dsname
     is the datasource name

dsuser
is for db2 user

-log
is modeler batch option, defines the log file

-appendlog
is modeler batch option, append log to above log file

dspwd
is datasource password for db2 user

ana_year
is the analysis scope, default is 20 years from current year

Note: You can use plain text to modify `dspwd`, or you can encrypt the password with the file `encrypt.sh`.

- `/opt/IBM/energy/AHI/SPSS_stream/conf/stream_model.cfg` Defines the execution order of the models. You can edit this file to add and remove the models to be run.
- The utility models are not required to be changed.

  `/opt/IBM/energy/AHI/SPSS_stream/conf/stream_assetHealth_clearup.cfg` Defines the stream to clean up the asset health database.

  `/opt/IBM/energy/AHI/SPSS_stream/conf/stream_assetHealth.cfg` Defines the stream to move the analysis results into the database.

Uploading a custom analysis model and stream

The custom analysis model up-loader has a folder management feature that lets you manage the different models and streams that you wish to upload.

About this task

Before you upload an anaysis, create the folder hierarchy for the analysis: model and data stream.

Procedure

1. Click **Administration** > **Custom Analysis Model**
2. Click the **Manage Model** tab.
3. Click **Create folder** and type a name for the analysis model, for example `APM`.
4. Create the folder for the data, in the column **Create folder** click the and type a name for the data for example `data`.

   Figure 47. Create folder icon in a row
5. Type the folder name for model data.
6. Create the folder for the data stream, in the column **Create folder** click the and type a name for the data for example `stream`.

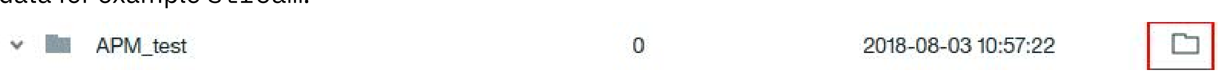
   

   Figure 48. Create folder icon in a row
7. Upload the data stream, click the **Upload** icon for the `stream` folder and select the str file from your source.

   You can add one file per upload. If you want more than one file, you must add each one separately.
8. Repeat for the data folder.

Setting the global parameters and configuring the analysis model

The Manage Configurations page is where you set the parameters for the stream and stream order.

About this task

You can organize the configurations in folders that you create.

Procedure

1. Click **Administration** > **Custom Anaysis Model** > **Manage Configurations**
2. Type the name and description for the configuration.

   This is the folder to manage the configuration parameters.
3. Under **Set Global Parameters**, click **Add new**.

   Global parameters are used for all data streams.
4. Complete the global parameters for the configuration for `Name` and `Command`:

   Table 20. Example global parameters

Name	**Command**
dsname	IoT4EUDB
dsuser	db2inst1
dspwd	pw4ibmioteusw
ana_year	20

   Click **Add new** for each new parameter.
5. Under **Configure Model**, Click **Import**.
6. Make a selection the streams for the configuration that you need to import. The streams are from the **Manage Models** page.
7. When you have completed the selection, click **Import**.
8. Add the stream that you want to add the parameters for. Click on each of the imported streams as required.

   The **Edit models** window opens.
9. Click **Add new**.
10. Type the parameters and commands for each of the required parameters.

    Table 21. Example stream parameters

Name	**Command**
csvFolder	/opt/IBM/energy/AHI/SPSS_stream/data/ model_output
AHI_Factor_Weight	AHI_Factor_CircuitBreaker.csv
assetTable	CIM.CIRCUITBREAKER
asset_AHI_Factor_csv	AHI_factors_CB.csv
asset_AHI_csv	AHI_CB.csv
asset_Detail_csv	circuitBreaker.csv
11. Click **Save** and **Save and back**

Running an analysis

You can either run an analysis from the **Manage Configuration** page or create a name for the analysis and select the configuration for the analysis you need to run.

Procedure

1. Click **Administration** > **Custom Analysis Model**.
2. In the **Manage Analysis** page, click **Run New Analysis**.
3. Type a name for the analysis and **Select configuration** from the drop down menu.
4. Click **Save and Run**.
5. When the analysis is complete, the time duration and log files are available.

Viewing and analyzing energy data

Use IBM Maximo Asset Performance Management for Energy & Utilities SaaS to provide data analysis, the calculation of failure and risk as well as providing an estimate of failure and risk.

Assets can be viewed on a geospatial map or as a list. The displayed assets can be filtered using criteria specified by the user.

Detailed reports can be displayed for individual assets, or groups of assets.

The user interface

The application user interface is composed of four parts:

- Filter selector - to filter assets for different criteria, asset class, type, score range, geography, and advanced.
- View selector - to select the content viewer between map, list, report, and matrix view.
- Content area - to visualize differing approaches of assets, including map, list, report and matrix views.
- Legend panel - to show the different asset classes and the score ranges.

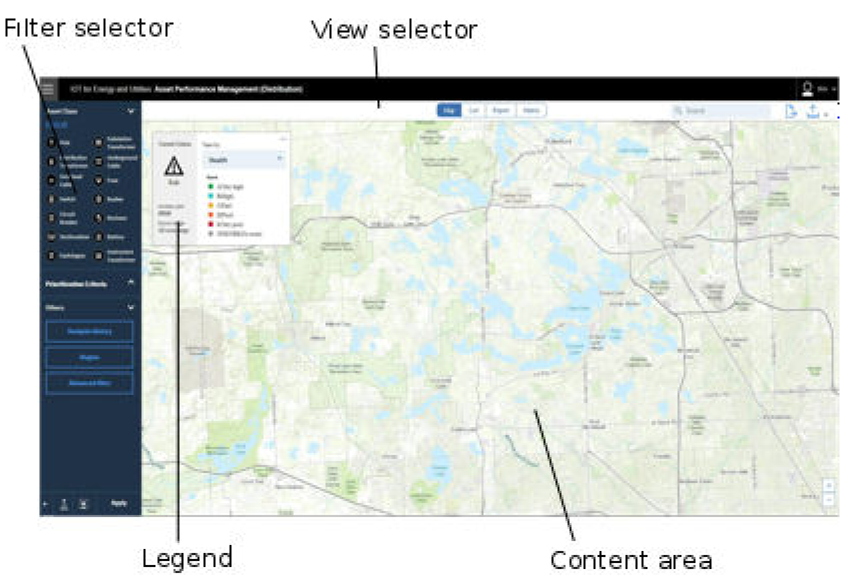


Figure 49. User interface for Asset Health

Logging on to the IBM Maximo Asset Performance Management for Energy & Utilities SaaS Asset Performance Management application

Log on to access the IBM Maximo Asset Performance Management for Energy & Utilities SaaS user interface.

Before you begin

Contact your local administrator to obtain your user ID and password. Your administrator is responsible for ensuring that you have the security access level that is appropriate to your role in your organization. Your administrator will also supply you with the web address URL for accessing the solution portal.

About this task

Use the following procedure to start a new browser session and access IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

Procedure

1. Enter the URL into the address field of the browser.

   Note: The fully qualified domain name is required in the URL, for example, `https://<App Node>/ibm`. If you use the IP address instead of the registered fully qualified domain name, some windows do not open correctly. Also, if you do not use the https protocol, the link is redirected to use the https protocol.
2. On the login page, enter your user ID and password.
3. Click **Log In**.
4. Click the menu icon and click **Asset Performance Management**.

Results

Only the pages, features, and data that you have permission to access are displayed. Contact your administrator if you require more access.

Navigating the user interface of IBM Maximo Asset Performance Management for Energy & Utilities SaaS

In IBM Maximo Asset Performance Management for Energy & Utilities SaaS, you can navigate to a specific page using the navigation bar.

The navigation bar contains four parts:

- Navigation segment
- Search box
- Transfer icon
- Download icon

Navigation segment

The navigation segment is part of the navigation bar. Maximo APM for E&U SaaS has four choices:

- **Map** - map view
- **List** - list view
- **Report** - report view
- **Matrix** - matrix view

Select the navigation segment for the view you need.

Search box

The search box is available in the list, report, and matrix views and is in the navigation bar. Here you can search for assets classes. The search box completes the search automatically.

Transfer icon

The transfer icon is part of the navigation bar. When you select an asset you can transfer the data for that asset to Asset Investment.

Down-load icon

The download icon is part of the navigation bar. When you click the icon, the report is downloaded.

Preview cards

When a user clicks on an asset or region, a preview card is displayed with additional information on that asset or region.

If a region containing multiple assets is selected, the preview card will display the average scores for the region, the total number of assets by class within the region, and available actions.

If a single asset is selected, the preview card will display the scores for that asset as well as available actions.

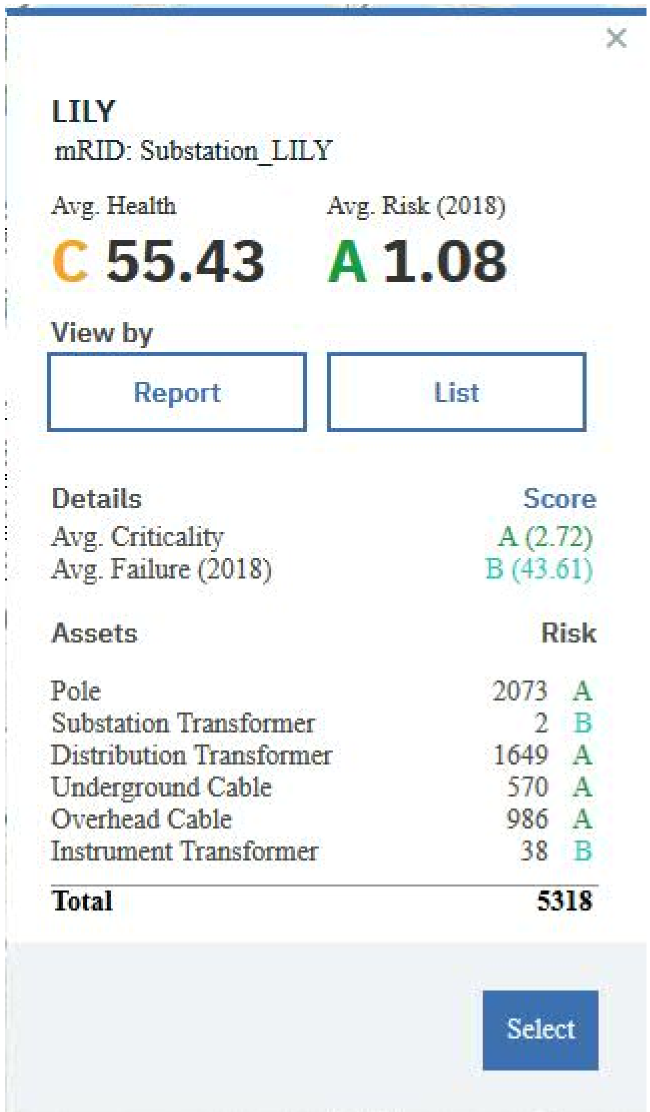


Figure 50. Preview card for a substation

Filter selector

The filters selector contains the controls to filter the visualization of assets on the main view.

You can select the filter control options that are available. The predefined filters are:

- **Asset class** - Shows the asset class that you want to view.

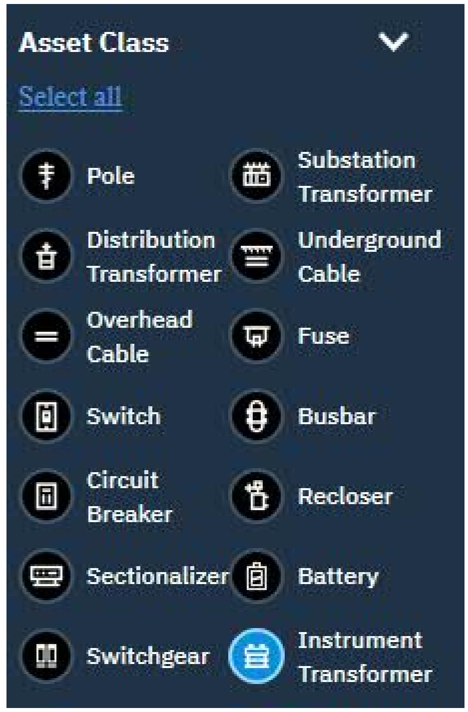


Figure 51. Asset class filter

- **Prioritization Criteria** - the filter criteria are:

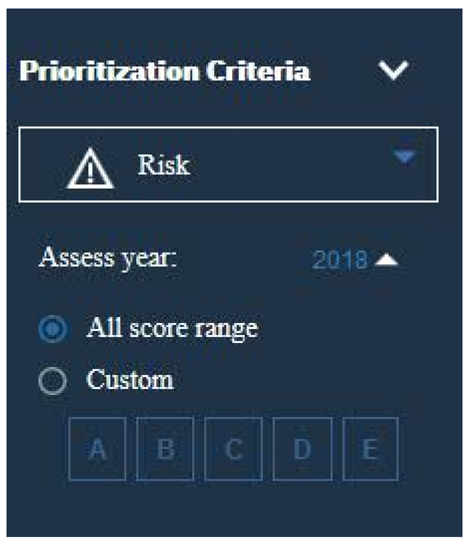


Figure 52. Prioritization criteria filter

  – **Score type** - The four score types are: Health, Risk, Failure and Criticality, where **Health** is the asset health index, **Failure** is probability of failure, **Criticality** is the seriousness of that failure and **Risk** is the measure of the risk to the business if a failure in the network occurs. When you select **Risk** you can also select the year.
  – **Asses year** - You can choose the present year, or a year in the future assess the risk of network or asset failure for that year.
  – **All score range** - The status filters for all score types.
  – **Custom** - The status filters for the score type you select. There are six states, A - Very low, B - Low, C - Poor, D - High, E Very high, and No score.
- **Others** - You can refine your selection to include:

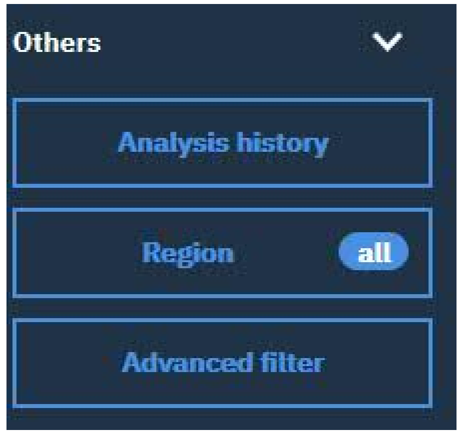


Figure 53. The other filter criteria

- **Analysis history** - Every time you run an SPSS analysis the generated results are on stored on the database. Analysis history lets you select the results from a previous analysis or select the system default which is the most recent analysis.
- **Region** - Filters the area map that you require.
- **Advanced** - You can add your own filters. The advanced filters are:
  1. **Filter scope –** You can upload custom files and select one or multiple of them to filter.
  2. **Edit advanced filter –** You can select an **asset class** and its **sub-component class**. You can also select **filter type**. The filter types are: **Attribute** and **Factor**.

  The properties of **Attribute** options are:

  - serialNumber
  - isActive
  - installationDate
  - removalDate
  - manufacturer
  - model
  - operatingVoltage
  - interruptingMedium
  - facillityId

  The properties of **Factor** options are:

  - Disorientation of the pole
  - Horizontal displacement of footings
  - Internal wood rot
  - Severity of burn marks
  - TTarget
  - Vertical displacement of footings
  - Wood Pole Remaining Strength Measurement

  The Relationship gives you the ability to set a value to a property. Depending on the property selected you can set a relationship type:

  - For string input:
    - is
    - is not

    - starts with
    - contains
  - For a boolean input:
    - is
  - For a date input:
    - after
    - no sooner than
    - before
    - no later than

  For a number input:
    - =
    - >
    - <
    - >=
    - <=

  The value relates directly to the property and relationship selected.
  - For a string, type the string value.
  - For a boolean value:
    - True
    - False
  - For a date value, select the date from the calendar.
  - For a number, type the number as a value.
- Time line- The period of time for the assets that you want to view.

Filtering assets

The assets displayed on the map or list can be filtered based on selected criteria.

About this task

IBM Maximo APM for Energy & Utilities has the following predefined filter options:

- Asset Class
- Score Range
- Type, where **Health** is the asset health, **Failure** is probability of failure, **Consequence** is the consequence of failure.
- Region
- Advanced

Additional filter criteria can be specified using the **Advanced** option.

To reduce the number of displayed assets to those meeting the desired criteria, do the following.

Procedure

1. Select the asset-classes that you need too assess. The icon for the selected asset class becomes light blue.
2. From the **Prioritization Criteria** select the score type.
3. Click **All score range** or select **Custom** to select the A,B,C,D, or E in the score range field.
4. Click the **Analysis history** button to select a previously saved analysis or to use the system default.

5. Click the **Region** button to open the region dialog window. Make the required selection of utility, substation and feeders.
6. Click the **Advanced filter** button to open the advanced dialog window.

   You can make a selection to your own criteria. When complete click **OK**.
7. Click **Apply**.

Results

The map or list view will display the assets meeting the selected filter criteria.

Creating a filter preset

You can create filter presets to be able to analyze similar filter selections.

About this task

You can use the filter criteria to create and save a preset. The preset save icon:

Procedure

1. Click the menu icon and click **Asset Performance Management** > **Asset Performance Management (Transmission)** or **Asset Performance Management** > **Asset Performance Management (Distribution)**.
2. Select the asset-classes that you need to create the preset for. The icon for the selected asset class becomes light blue.
3. From the **Prioritization Criteria**, select the score type.
4. Click **All score range** or select **Custom** to select the A,B,C,D, or E in the score range field.
5. Click the **Analysis history** button to select a previously saved analysis or to use the system default.
6. Click the **Region** button to open the region dialog window. Make the required selection of utility, substation, and feeders.
7. Click the **Advanced filter** button to open the advanced dialog window.

   You can make a selection to your own criteria. When complete click **OK**.
8. Click the preset save icon and type the name of the preset. If you are an administrator, you can select **Share this preset with public** to share the template with all users. You can also share the preset with specific user groups.
9. Click **Save**.

Results

You can use this saved preset to load to the application.

Loading a filter preset

A preset can be used to load filter criteria Asset Performance Management application.

About this task

After you have created a preset you can load the preset to the Asset Performance Management application. The preset load icon:

Procedure

1. Click the menu icon and click **Asset Performance Management** > **Asset Performance Management (Transmission)** or **Asset Performance Management** > **Asset Performance Management (Distribution)**.
2. Click the load preset icon and select the preset you want to use.

3. Click **Load**.

   You can now use this preset or edit the filter criteria to create a new analysis.
4. Click **Apply**.

Viewing the health status of assets in the map view

You can view the health status of assets classes in the map view. The map provides a visualized distribution of risk and failure.

About this task

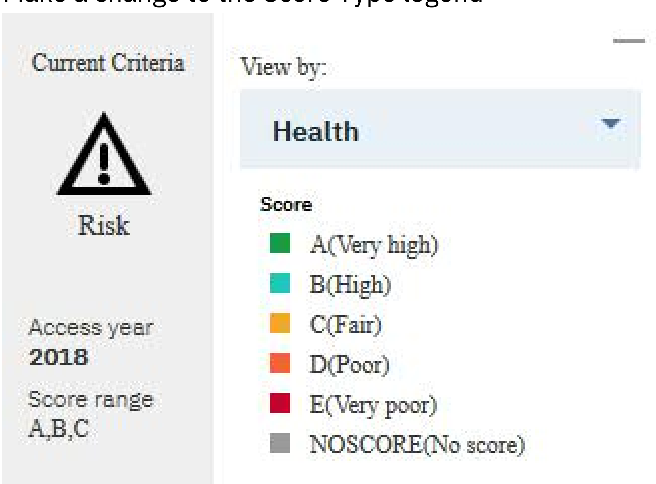
The map contains the following parts:

- Map area
- Asset class legend
- Score Type legend

Procedure

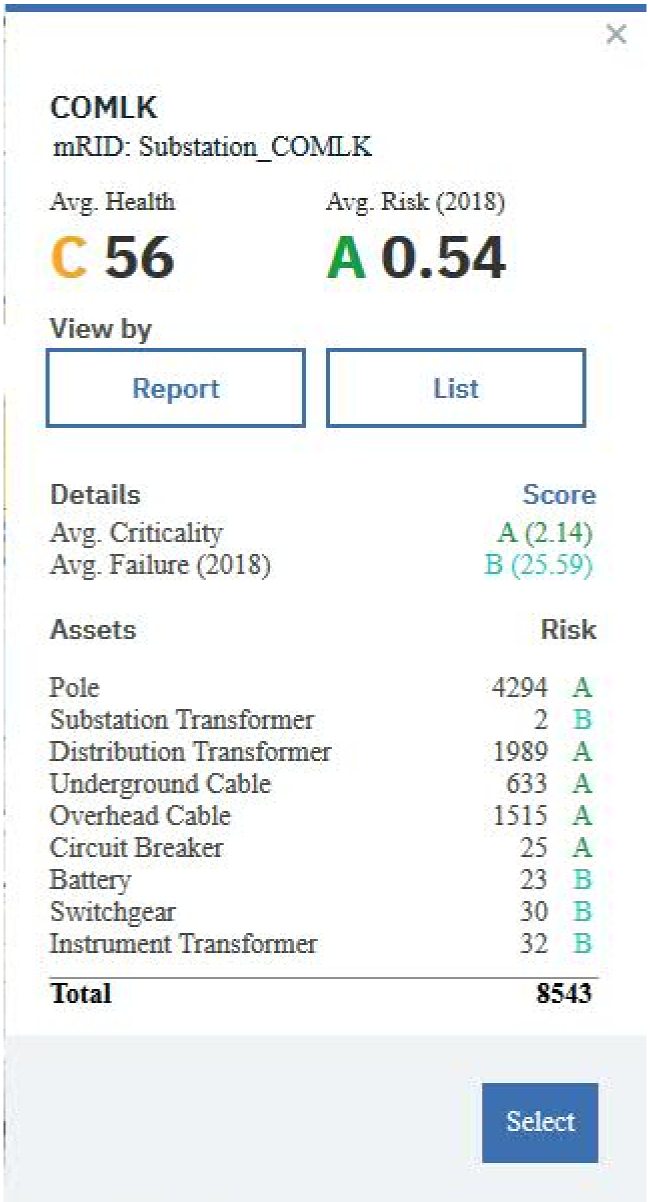
1. Sign on Maximo APM for E&U SaaS as a user.
2. In the Navigation segment click **Map**.

   The map is displayed.
3. Use the **Filter** selector to make a selection of an **Asset Class**, **Prioritization Criteria**, **Others**, and click **Apply**.
4. Make a change to the Score Type legend

   and view the change on the map.

   Note: If you choose the score type Risk or Failure then a time line shows. The Asset Health application estimates Risk and Failure for the future. If you choose a different year in the time line, the map shows the estimated values for that year in the map view.
5. Click on an area to view the preview card for the substation for that area.

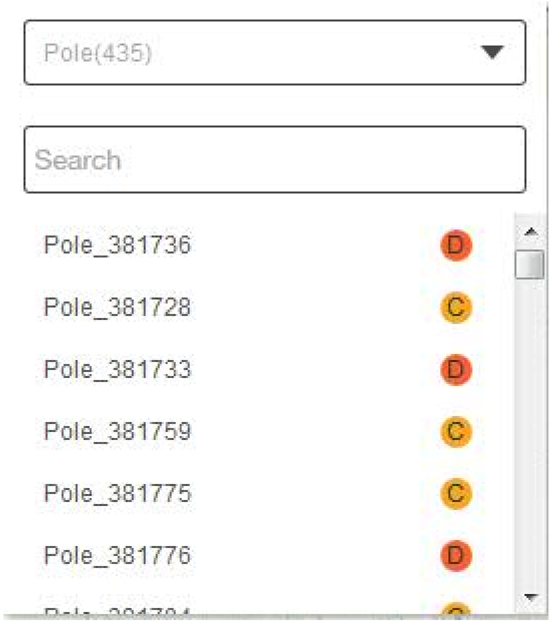


6. You have choices:
    a) To see the assets in the report view, click **Report**.
    b) To see the assets in the list view, click **List**.
    c) To see the assets in the map view, click **Select**.
7. Zoom in the map to see the details of an asset class.
8. Click the circle to open the details of the substation.
9. Click a grey circle that denotes a cluster which contains a number.

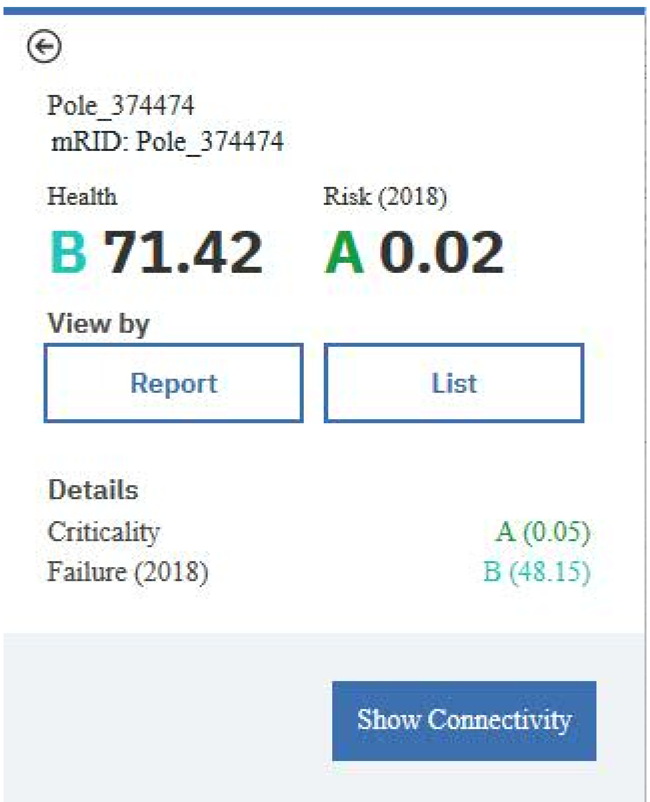
    A card opens that shows the details of the cluster.

    Note: The grey circle with the number indicates a cluster of assets. The number indicates the quantity of assets in that cluster.

    Note: You can search for a particular asset using the search box.



10. Click an asset from the list to open the details of that asset.



Results

In the map view you have a visualization of risk, failure, criticality and health for the different assets. By selecting different items in the legend and year in the time line, you can see the information for the different conditions.

Showing and Hiding the map street view

In IBM Maximo Asset Performance Management for Energy & Utilities SaaS you can view the location of an asset view in the street view. You can configure the application if you do not require this function.

About this task

Here you can change the configuration file to show or hide the map street view of Maximo APM for E&U SaaS.

Procedure

1. Find the Google map configuration file.
   - If you want to change distribution page open the file `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_ah_app.ear/ife_ah_web.war/config/modelDistribution.json`.
   - If you want to change transmission page open the file `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/apps/ife_ah_app.ear/ife_ah_web.war/config/modelTransmission.json`
2. To show the Google map street view, change the parameters as follows:

   ```
   "showStreetMap":true
   "googleStreetMapKey":yourMapsJavaScriptAPIKeyNumber
   ```

3. To hide the Google map street view, change the parameter as follows:

   ```
   "showStreetMap":false
   ```

4. For information on how to apply your Maps JavaScript API key, use the link: `https://developers.google.com/maps/documentation/javascript/get-api-key`

Viewing the physical location of a single asset in street view

You can view the physical environment of a single asset that has a point location in IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

With the **Street View** you can see the environment and conditions for the location of an asset. For example you can see the proximity of trees and branches to a pole with the overhead conductors. An assessment team can also view the location before an inspection.

The **Street View** of an asset is avaiable in the **Map** view.

You can also get to a single asset from the List view, and selecting **View on map**.

Procedure

1. In the **Map** view, click the icon for the single asset.
2. Click **Street View**.
3. The **Street View** opens at the closest location to the asset from the street or road.

   With your mouse you can rotate the field of vision through 360 degrees.

Viewing the health status of assets classes in the list view

Assets and their network health and risk values can be displayed as a list.

About this task

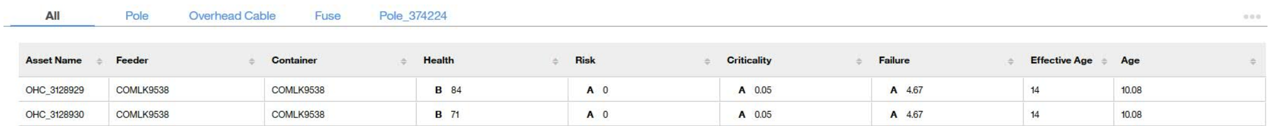
The List view contains the following parts:

- Main table

- The individual asset class tabs

A list view has these columns:

- Asset Name
- Feeder
- Container
- Health,
- Risk,
- Criticality,
- Failure,
- Effective age,
- Age.


All Pole Overhead Cable Fuse Pole_374224

Asset Name	Feeder	Container	Health	Risk	Criticality	Failure	Effective Age	Age
OHC_3128929	COMLK9538	COMLK9538	B 84	A 0	A 0.05	A 4.67	14	10.08
OHC_3128930	COMLK9538	COMLK9538	B 71	A 0	A 0.05	A 4.67	14	10.08

Figure 54. List view

When you view a single asset list, the year column also shows.

In this task you will select a different asset class tab and select a different year and review the results in the table.

Procedure

1. Sign on IBM Maximo APM for Energy & Utilities as a user.
2. Use the **Filter** selector to make a selection of the **Asset Class**, **Prioritization Criteria**, **Others**.
3. In the navigation segment bar click **List**.

   The list is displayed.
4. Click an asset class tab and view the result in the table.

   The filter selection you have made determines the content of each asset class tab, when you change the filter selector you change the items on display in the list view.
5. Select a different year in the time column to see how changes to the year changes the health of the asset.
6. Click an asset in the main table to open a hover menu with the items, **Open in new Tab**, **View by Map**, **View by Report**.

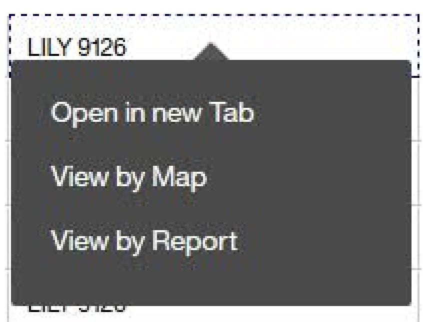
   

   Figure 55. Hover menu items
7. Click **Open in new Tab** to view a single asset list with its history.
8. Click on the **View by Map** or **View by Report** to open the map or report views.

Results

In the list view, you can see the details of every asset. You can open a specific asset in new tab, in the map view or in the report view to see more details.

Viewing the health status of multiple asset classes in the report view

The report view provides many visual charts. You can get a visualization of the health status of multiple asset classes.

About this task

In this task you select different tabs and in the report view see the results as visualizations. Both single and multiple asset class reports are available. The charts that are available for multiple asset classes are:

- Multiple asset class report showing the basic information for more than one asset class.

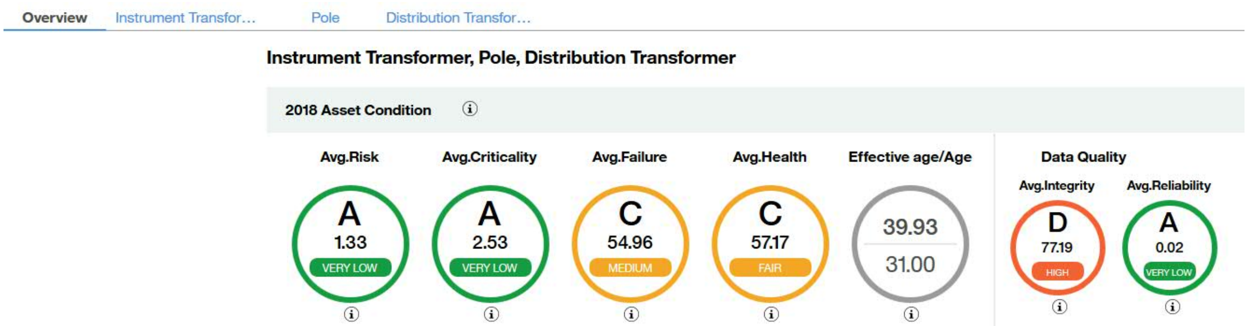


Figure 56. Multiple asset classes basic information

- The Average failure and risk over time shows the changes to average failure and risk over time.

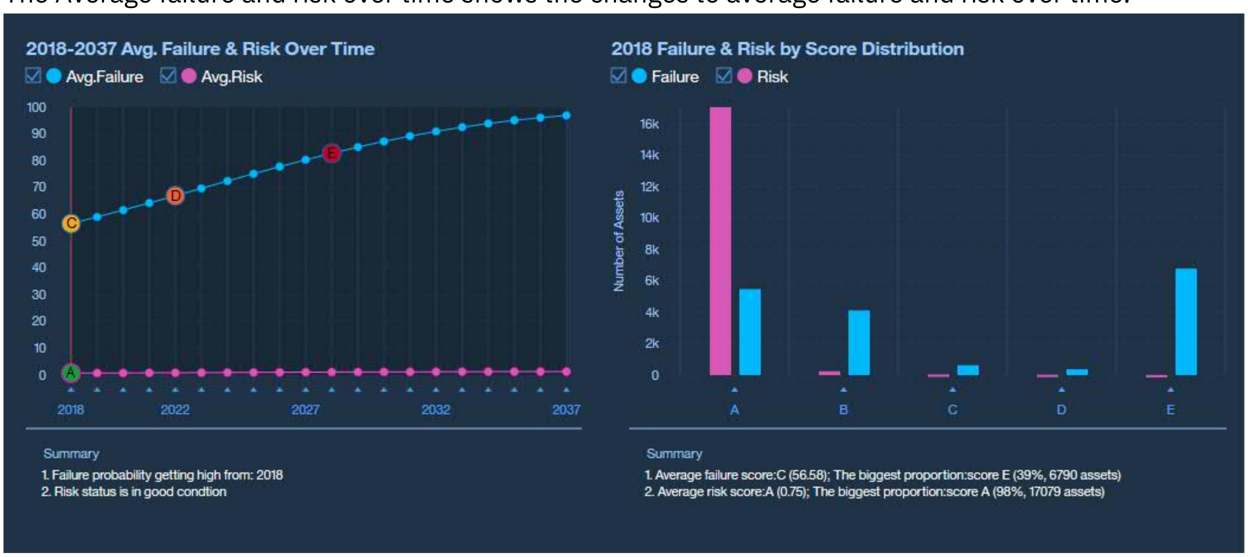


Figure 57. The changes to average failure and risk over time report

- Additional information report

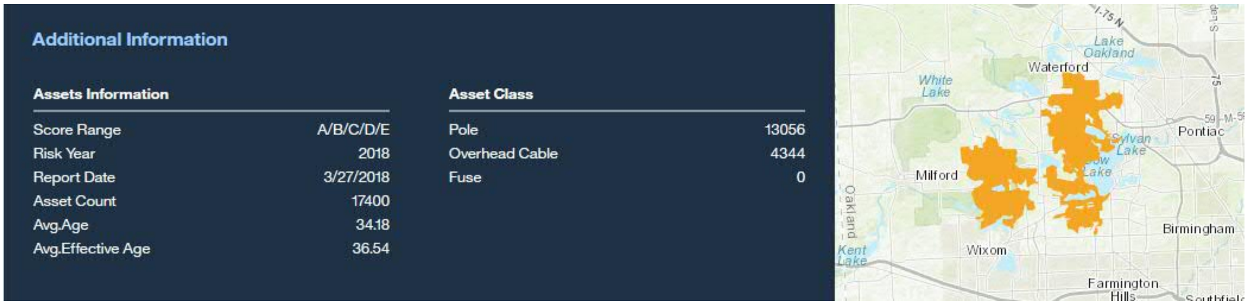


Figure 58. Aditional information report

- Asset Risk Distribution

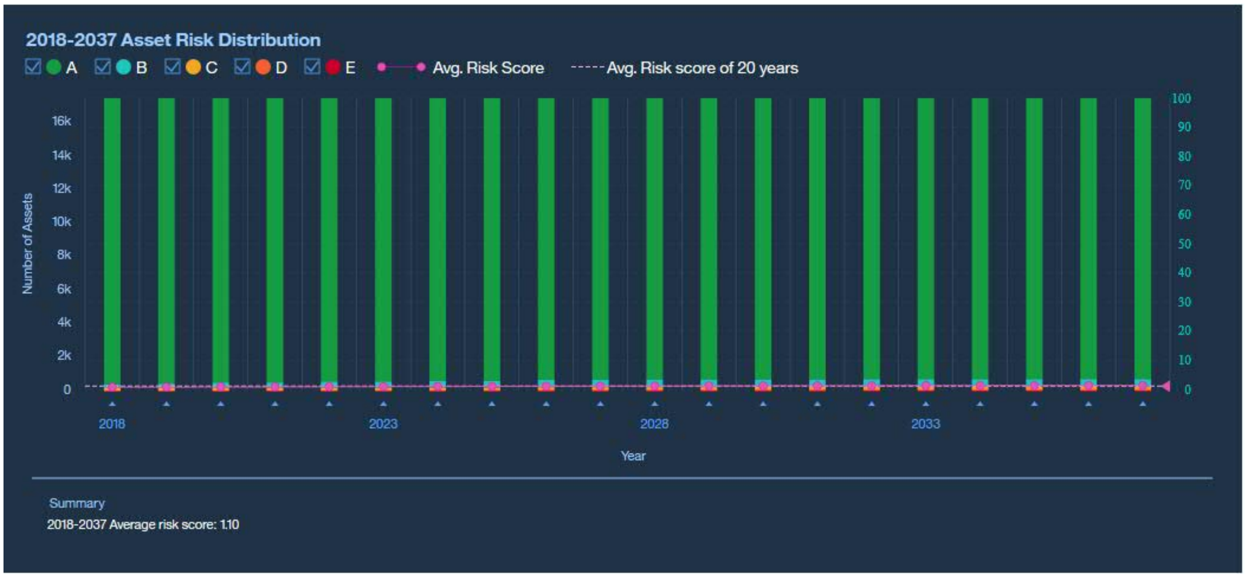


Figure 59. Asset risk distribution report

- Top 10 Highest Ranking - by region

Figure 60. Top 10 ranking by region

- Top 10 Highest Ranking by feeder

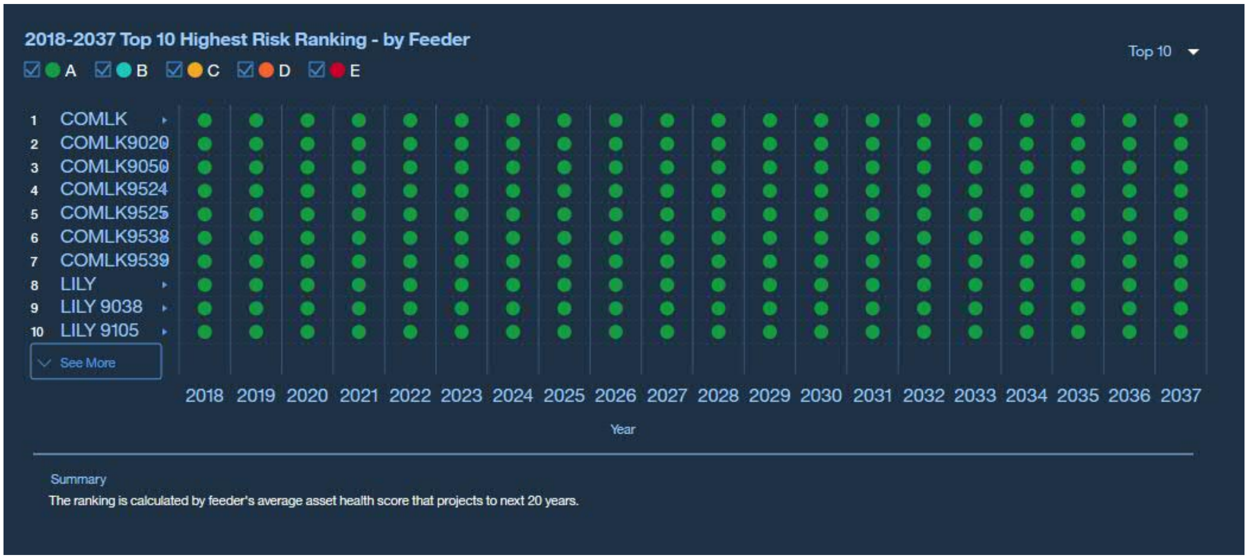


Figure 61. Top 10 ranking by feeder

Procedure

1. Sign on IBM Maximo Asset Performance Management for Energy & Utilities SaaS as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.

   When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **Report**.

   The report view opens.
4. See the reports available in the **Overview** report view.

Viewing the health status of a single asset class in the report view

The report view provides many visual charts. You can get a visualization of the health status of a selection of asset classes, a single asset class, and a single asset.

About this task

In this task you select different tabs and in the report view see the results as visualizations. Both single and multiple asset class reports are available. The charts that are available for a single asset class are:

- The overview report showing the basic information for a single asset class.

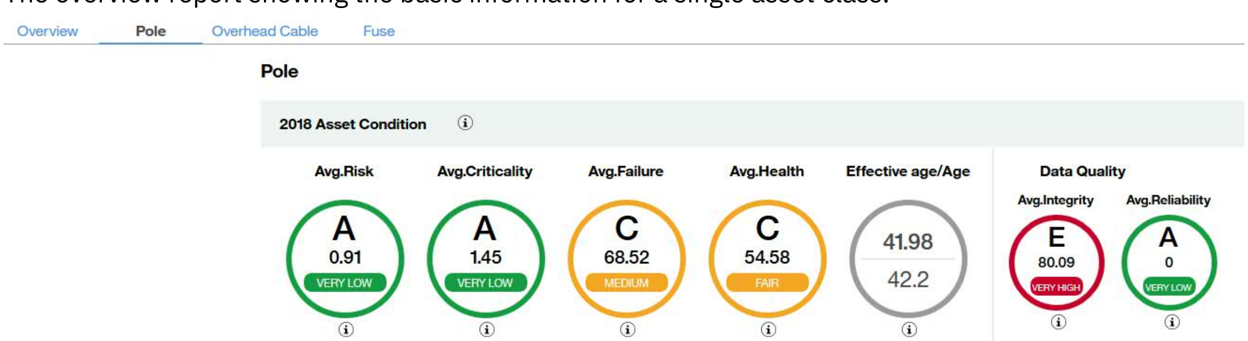


Figure 62. Overview showing a single asset class

- The basic information report shows the changes to average failure and risk over time.

Figure 63. The changes to average failure and risk

- Additional and location information.

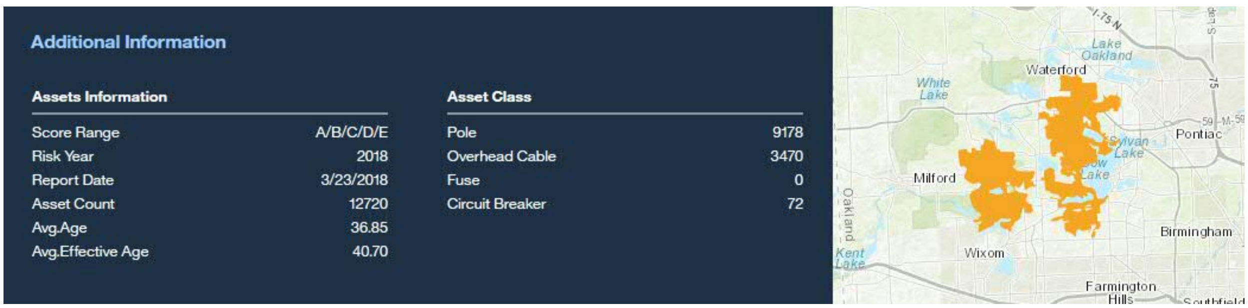


Figure 64. The additional information and map locator

- The distribution of average risk for all assets over time.

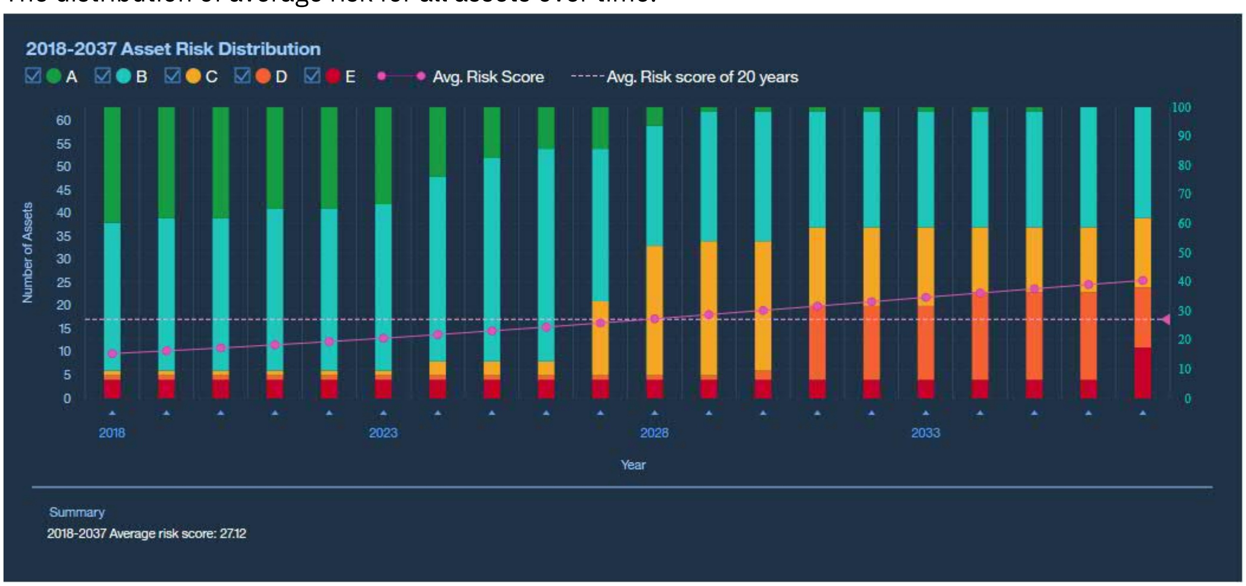


Figure 65. The score for average risk for an asset over time

- The level of average risk for a region changes over time.

Figure 66. Changes to ranking of the highest risk to multiple regions over time

- The level of average risk for a feeder changes over time.

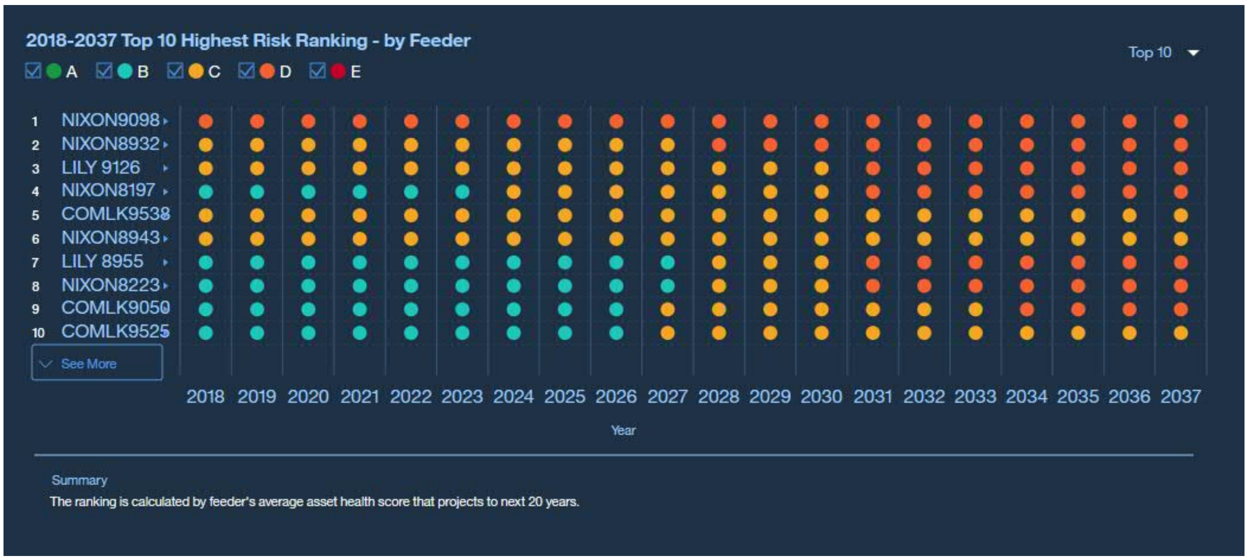


Figure 67. Chnages to the average risk for a feeder over time

- The degradation curve for an asset over time.

Figure 68. Degradation curve for an asset over time

- Asset Health Index by age distribution.

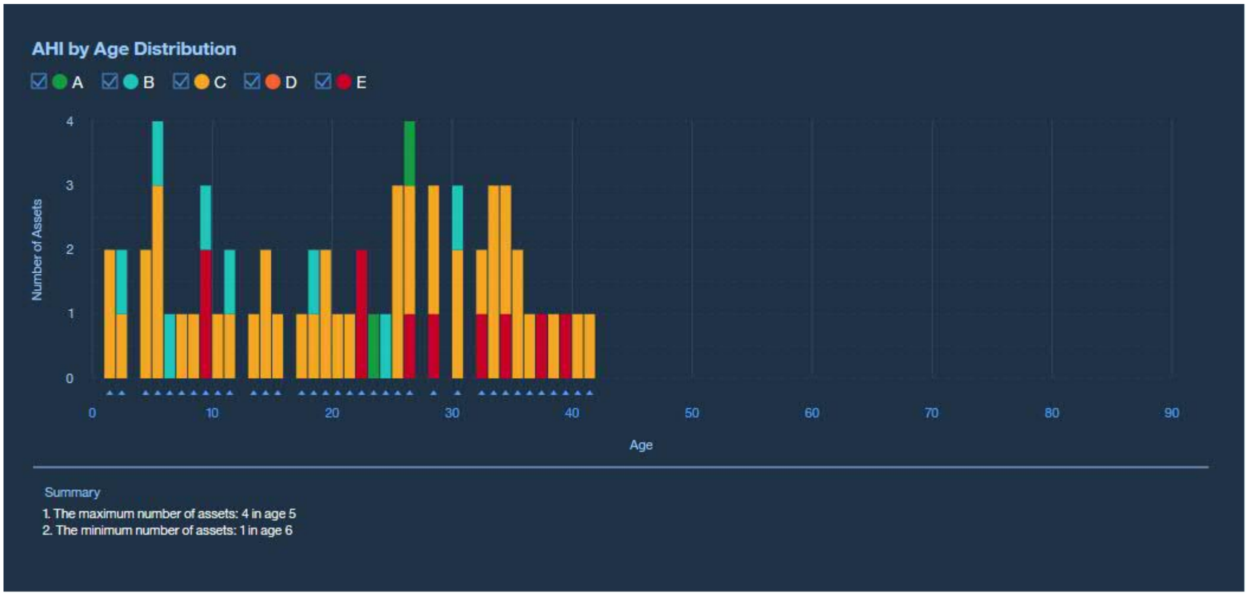


Figure 69. AHI age distribution

- Asset Health Index by effective age distribution.

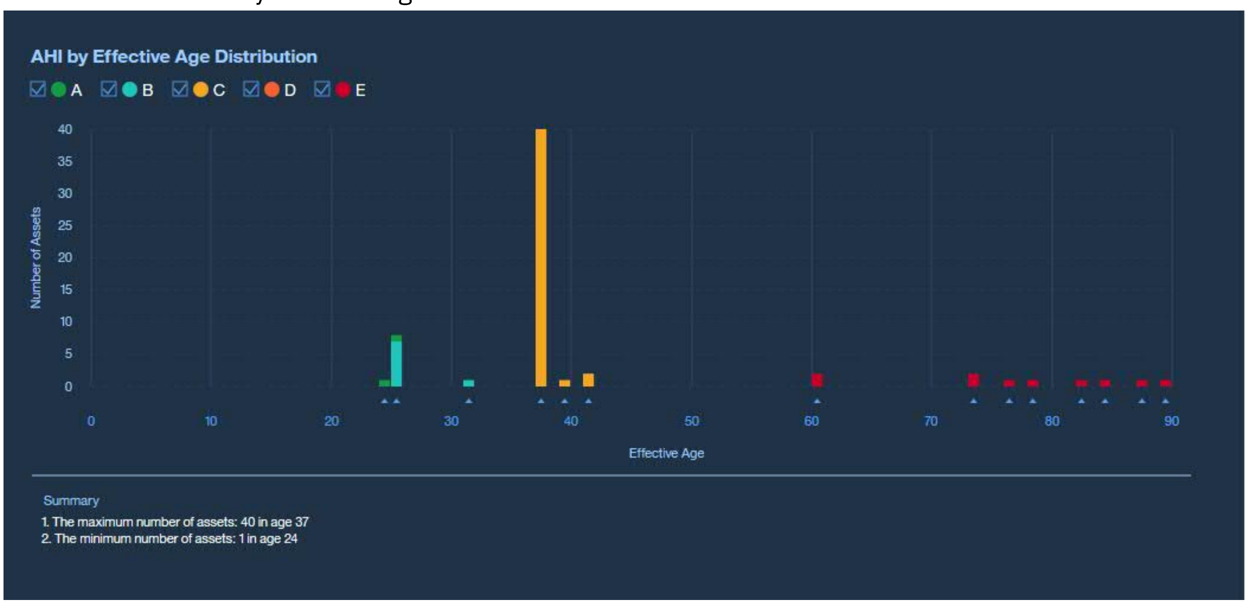


Figure 70. AHI effective age distribution

- Disolved Gas Analysis is in two parts. Part one shows the historical data of a substation and the trend over time. Part two shows the Du val method of diagnostic and the diagnostic result for a selected date.

Procedure

1. Sign on IBM Maximo Asset Performance Management for Energy & Utilities SaaS as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.

   When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **Report**.

   The report view opens.
4. Click an asset class tab and view the result in the report.

The filter select you have made determines the content of each asset class tab, when you change the filter selector you change the items on display in the report view.

5. See the reports available in the report view.

Results

In the report page, you can see the different charts available for the health status of an asset class.

Viewing the health status of a single asset in the report view

The report view provides many visual charts. You can get a visualization of the health status of a single asset or its subcomponent class.

About this task

In this task, you select different tabs and in the report view see the results as visualizations. Both single and multiple asset class reports are available. The charts that are available for a single asset class are:

- Single asset class report that shows the basic information for one asset.

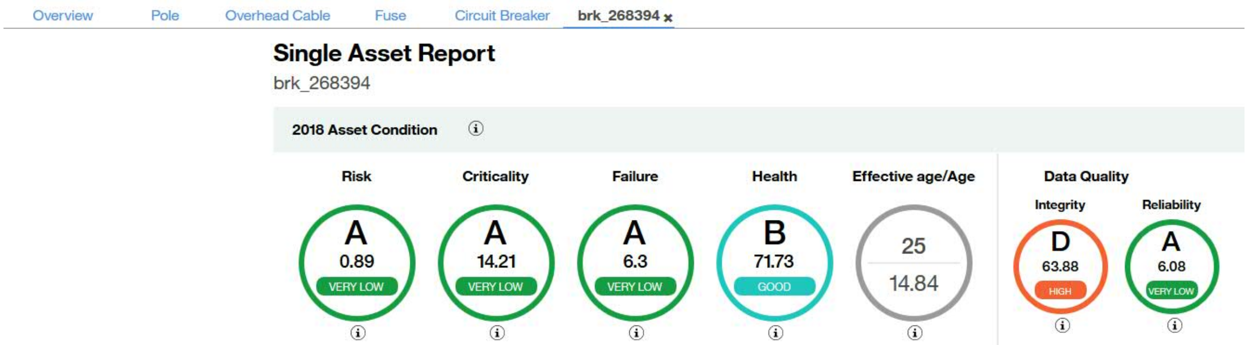


Figure 71. Single asset report overview information

- Basic Information

Figure 72. Basic Information report

- Multimedia

You can add images and video clips to the single asset report.

Figure 73. Multimedia images

- Health Breakdown

Figure 74. Health breakdown report

- Criticality Breakdown

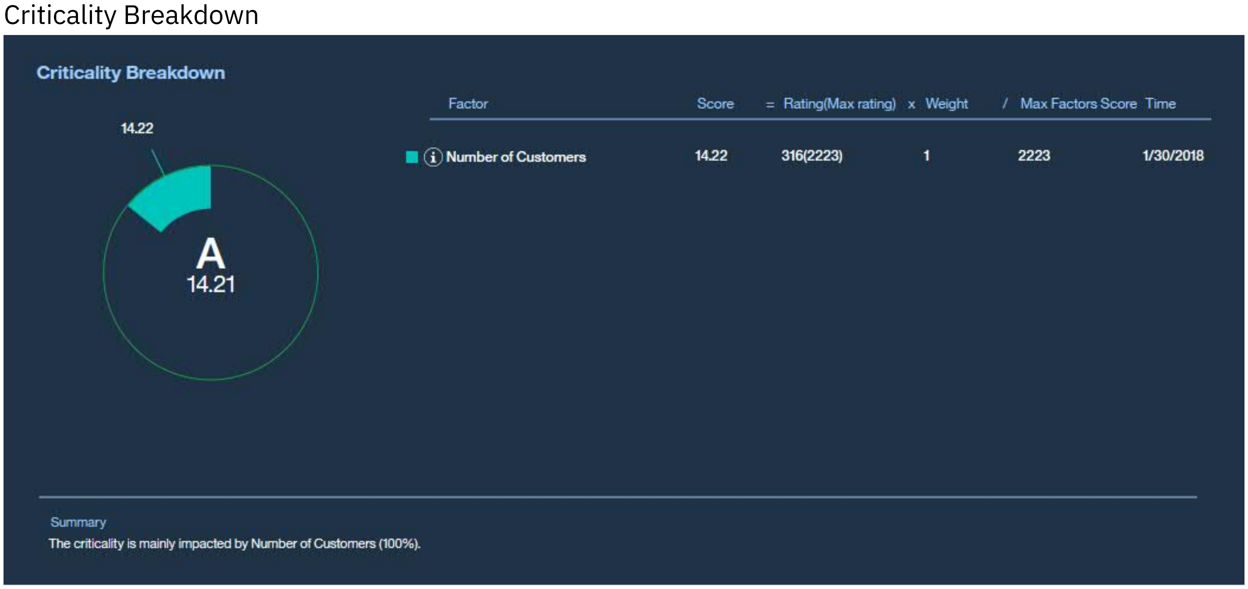


Figure 75. Criticality breakdown report

- Degradation Curve

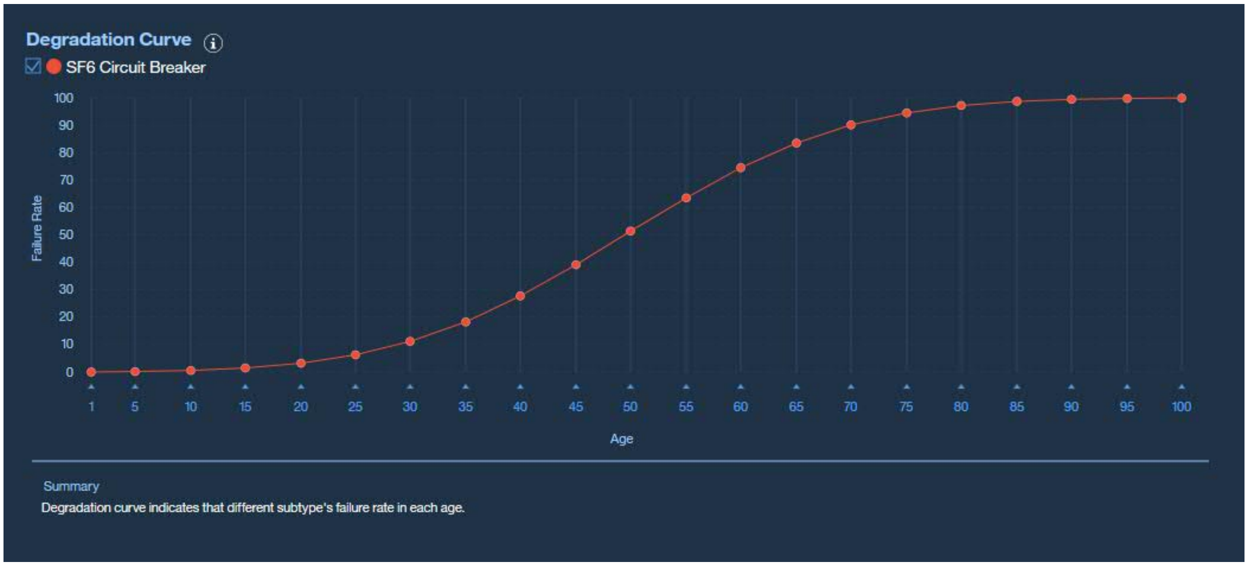


Figure 76. Degradation curve report

Procedure

1. Sign on IBM Maximo Asset Performance Management for Energy & Utilities SaaS as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.

   When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **List**.

   The list view opens.
4. Select the single asset from the **List** view and select **View by Report**.

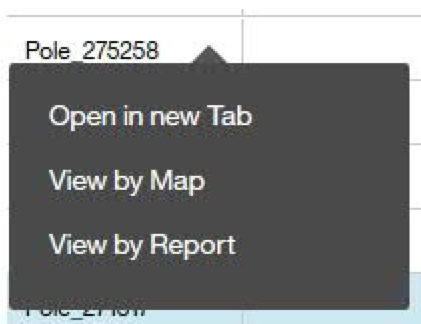
   

   Figure 77. Drop-down menu in the list view
5. See the reports available in the report view for a single asset.
6. If you see the reports for a subcomponent class of the single asset, click the drop-down menu in the **Single asset report** overview information or **Health breakdown** report and select a subcomponent class.

Viewing the health status of assets classes in the matrix view

Asset classes and their health status can be displayed as a matrix.

About this task

In this task, you select different tabs in the matrix and select different score type to see the detail information of an asset class.

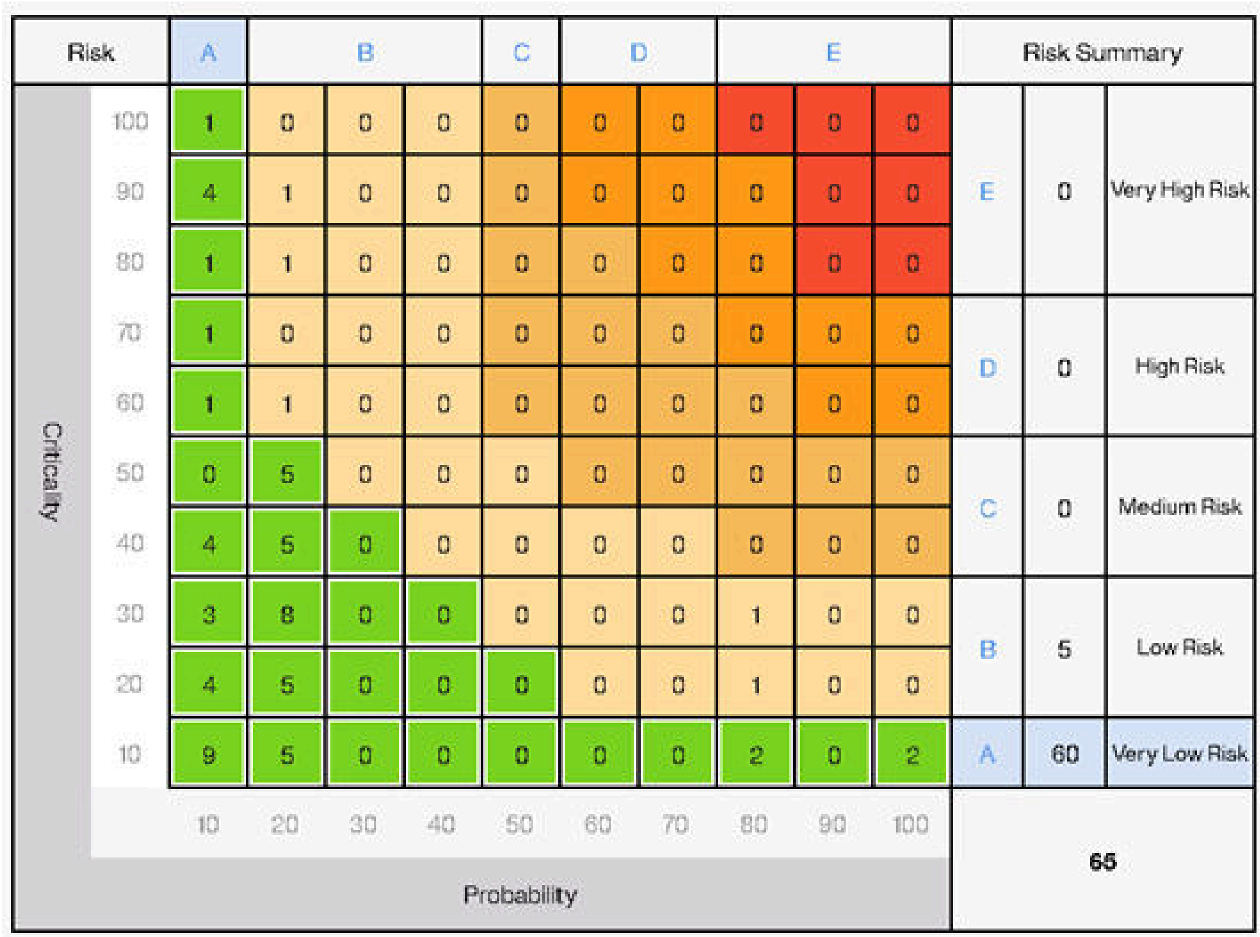


Figure 78. The health status of assets shown as a matrix

Procedure

1. Sign on Maximo APM for E&U SaaS as a user.
2. Use the **Filter** selector to make a selection of an **Asset Class**, **Score Range**, **Type**, and **Region** and click **Apply**.
3. In the navigation segment bar click **Matrix**.

   The matrix view opens.
4. Click an asset class tab and view the result in the matrix.

   The filter selection you have made determines the content of each asset class tab, when you change the filter selector you change the items on display in the matrix view.
5. Click different risk levels to highlight the results that correspond to that level of risk.

   The number in the results matrix indicates the number of qualified asset in that asset class.
6. View the preview panel to view the information details for the qualified asset class. You can also view the information by **Feeder** and **Region**.
7. Select a different year in the time line and observe the change to the information in the matrix and preview panel.

Exporting data for a single asset class

In IBM Maximo Asset Performance Management for Energy & Utilities SaaS you make an export of data in csv format for an asset class without having first to create a report.

Before you begin

You can export data directly from Asset Performance Management Transmission or Distribution, you do not need to create a filter before you make the export.

About this task

You can make a single selection for an asset class, and define the type of data you need to export directly from the database. The types of data you can export are:

- Asset master data.
- Asset health data.
- Asset measurement data.

Procedure

1. Click Asset Performance Management and select either **Transmission** or **Distribution**.
2. Click the **Export all data** icon.

   Figure 79. Exporting all data
3. Click **Export all data**.
4. Select the asset class that you want to export and the data type. Click **Next**.
5. Select the columns to export. The default selects all columns.

Results

You can export the csv file to your system and open it in the software of your choice.

Viewing analytics dashboards

When viewing a report, additional analytic data is available from IBM Maximo APM - Predictive Maintenance Insights.

Procedure

- When viewing a report, click **Advanced Analytics**.

Results

When viewing a single asset report, the Equipment Dashboard for that single asset is displayed. When viewing a multiple asset report, the site overview dashboard is displayed.

Work order management

You can integrate IBM Maximo Asset Performance Management for Energy & Utilities SaaS with IBM Maximo APM - Asset Health Insights On-Premises to create and view Maximo work orders from the report view of IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

Configuring work orders

You must configure the work order for use with IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

The configuration properties file for IBM Maximo Asset Performance Management for Energy & Utilities SaaS need to include the Maximo server IP address.

Procedure

1. Open the file: `opt/IBM/Websphere/liberty/list/servers/framework_server/lib/config.properties`.
2. Edit the file to include:

   ```
   MAXIMO_URL=http://IP address of the Maximo server
   ```
3. Save and close the file.
4. Restart the `frame_work` server for the URL to take effect.

   ```
   /opt/IBM/WebSphere/Liberty/bin/server stop framework_server
   /opt/IBM/WebSphere/Liberty/bin/server start framework_server
   ```

Viewing the work order history for an asset

You can view the work orders for an asset from the **Report** view for that asset.

Procedure

1. Sign on IBM Maximo APM for Energy & Utilities as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.

   When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **List**.

   The list view opens.
4. Select the single asset from the **List** view and select **View by Report**.

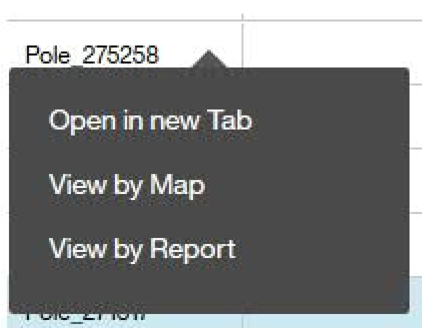
   

   Figure 80. Drop down menu in the list view
5. In the single asset report view the **Health Breakdown** report and click **View Work Orders**.

   The **Work Order History** opens.

6. To view the work order in Maximo, click **View**.

Creating a work order for an asset

You can create a work order for an asset from the **Report** view for that asset.

Procedure

1. Sign on IBM Maximo APM for Energy & Utilities as a user.
2. Use the **Filter** selector to make a selection of **Asset Class**, **Prioritization Criteria**, and **Others** and click **Apply**.

   When you select more than one asset class, you can receive a summarized report about multiple assets.
3. In the Navigation segment click **List**.

   The list view opens.
4. Select the single asset from the **List** view and select **View by Report**.

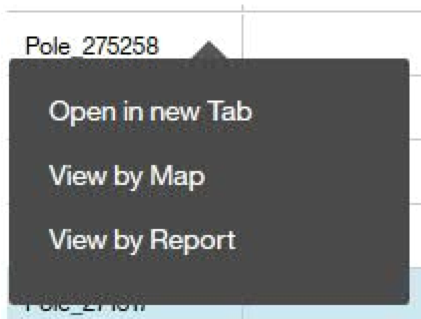
   

   Figure 81. Drop down menu in the list view
5. In the single asset report view the **Health Breakdown** report and click **Create Work Order**.

   You can create a single work order.

Asset Investment application

The Asset Investment application in IBM Maximo Asset Performance Management for Energy & Utilities SaaS helps asset managers to determine the best investment plan possible according to the utility objectives and constraints.

With Maximo APM for E&U SaaS, you can create an investment project for a particular asset class. The replacement costs for that asset are included when the investment project is set up. Various scenarios or models can be set to include the level of risk and failure, budget constraints, and planning duration.

Based on the asset health indexes, failure probability, criticality, risk and planning interval, you can review the future years for a single asset plan by map, list, or report view mode. You can then export the reports to whom it concerns. The default number of years to review is 20. The number of years can be changed by the user.

Creating an investment project

To create the various scenarios, you first must create the investment project in IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

About this task

The investment project includes the initial asset class, the subtype, and the replacement cost of those assets.

The results show an overview of the years of the report, the risk, the costs, and the number of asset replacements on an average, maximum, and minimum basis.

The yearly results can also be shown.

Procedure

1. Sign on Maximo APM for E&U SaaS as an administrator.
2. Click **Asset Investment**.
   - If this entry is the first project, click **Create Project**.
   - If there are multiple projects, you can create a project by clicking **Duplicate** on an existing project.
3. Click **Open**.
4. Type:
   a) The project name.
   b) Select the asset class to include.
   c) Select the asset subtype.
   d) Type the replacement cost.
5. Click **OK**.
   Maximo APM for E&U SaaS calculates the result and shows the results as a map view.

Viewing the results of the investment project in the map view

After the investment calculation finished in IBM Maximo Asset Performance Management for Energy & Utilities SaaS, you can view the results in a map view showing the assets as color circles on the map.

Before you begin

An investment project must be created and available for opening.

About this task

You can anticipate the replacement needs by cost, year, and location on the map view.

Procedure

1. Hover over the investment project you need and click **Open**.
2. The default map shows an overview of the results.

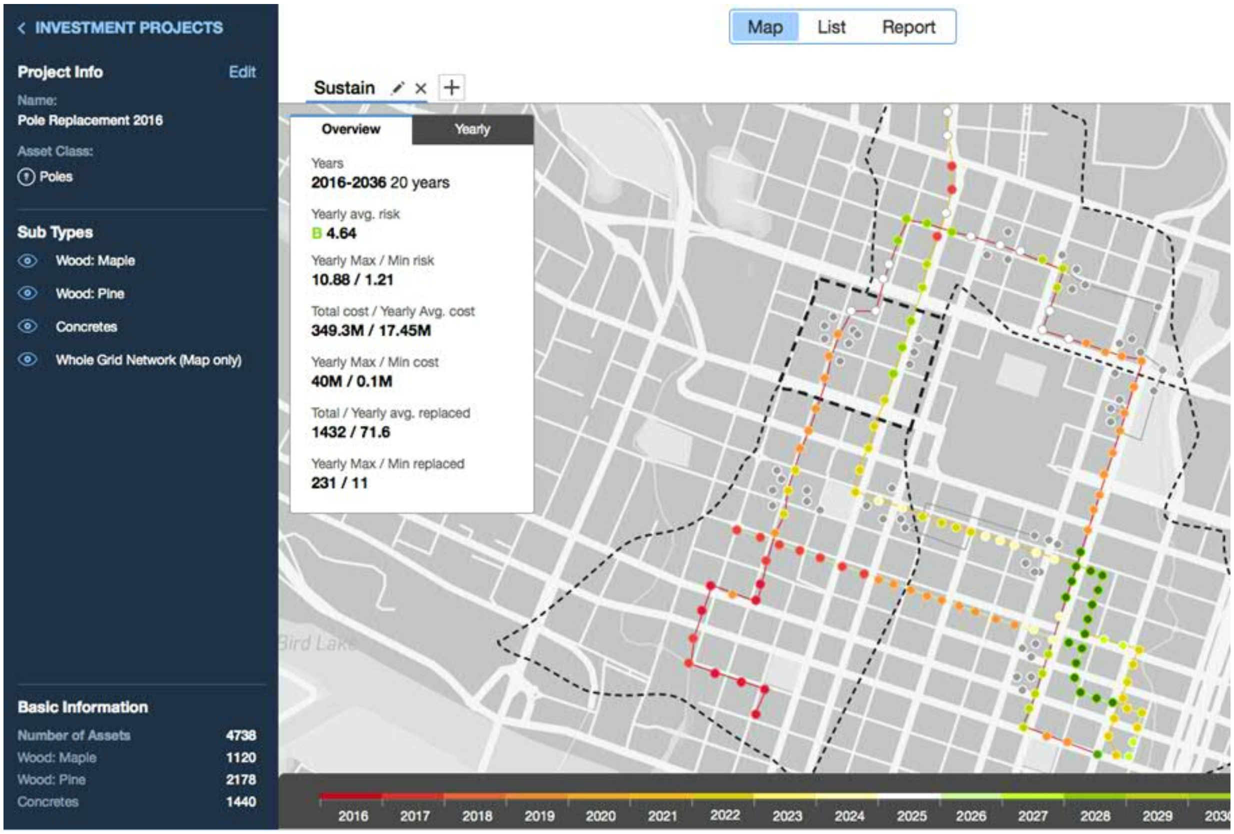


Figure 82. The map view of an investment project

3. To view the details of an individual asset, select a substation or feeder then zoom in until you see the specific asset and click.

   A chart opens that shows the replacement year, the risk before and after replacement, and the replacement cost.

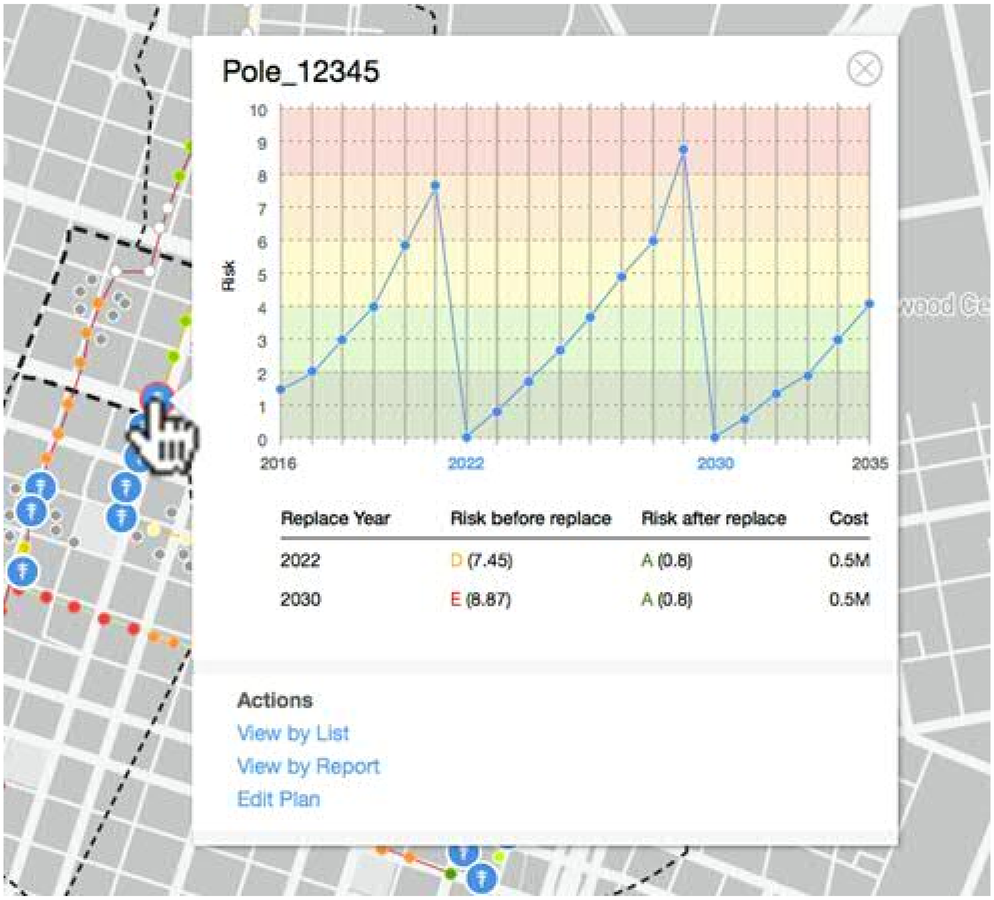


Figure 83. Asset details showing replacement year and risk

4. To see the yearly results, click the **Yearly** tab and move the slider on the timeline to view the year.
5. To select a range of years, in the **Yearly** tab, slide the **Select range** button and use the two slides in the timeline to select the range. The default is 20 years.

Viewing the results of the investment project in the list view

After the investment calculation finished in IBM Maximo Asset Performance Management for Energy & Utilities SaaS, you can view the results in a list view showing the assets as rows in the list.

Before you begin

An investment project must be created and available for opening.

About this task

You can anticipate the replacement needs by cost, year, risk, and failure in the list view.

Procedure

1. Hover over the investment project you need and click **Open**.
2. In the Navigation segment click **List**. The list is displayed.
3. Click the row containing the asset to view the details of that asset.

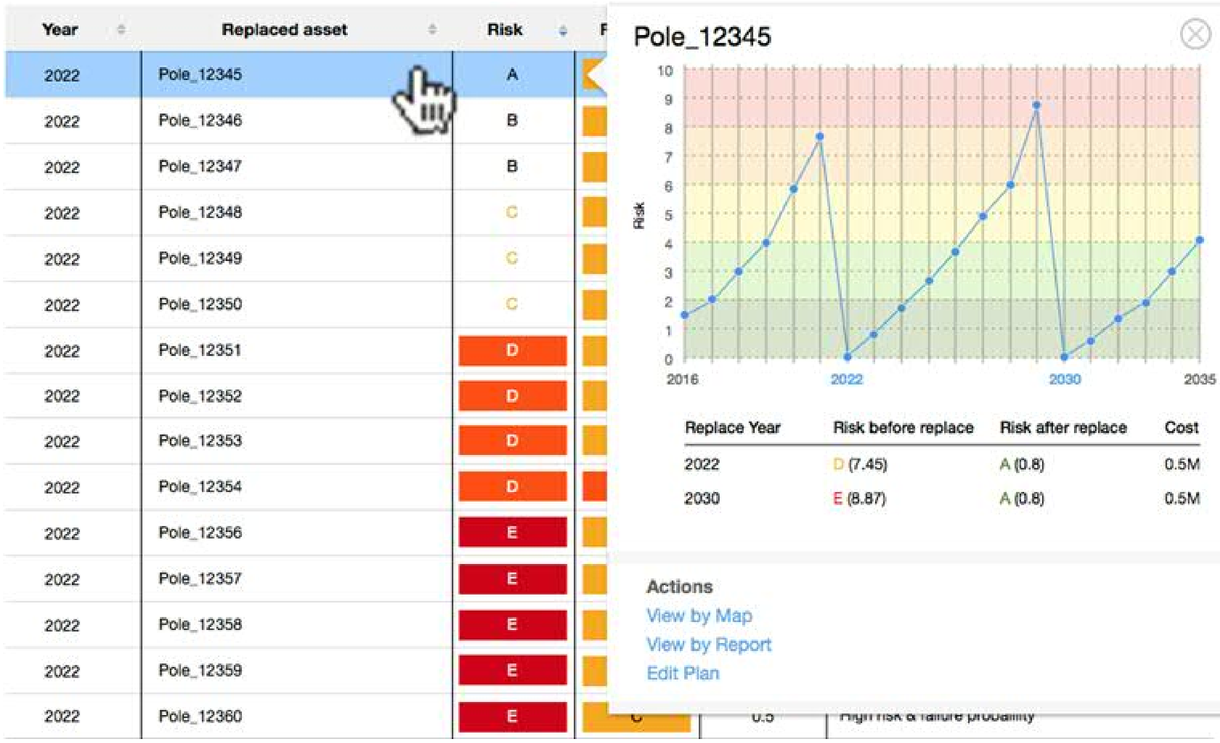


Figure 84. List view in Investment Planning

Creating a scenario

Using the Asset Investment Planning application in IBM Maximo Asset Performance Management for Energy & Utilities SaaS, you can create models for what if scenarios.

Before you begin

You must have an investment plan set-up in Maximo APM for E&U SaaS.

About this task

You can set up the plan duration and the failure threshold against either a budget or an acceptable level of risk.

Procedure

1. Click **Asset Investment**.
2. Click the investment project to work with.
3. Click the **Add** icon next to the **Sustain** tab.
4. Type the name of the scenario.
5. Select the orientation of the scenario.
   - **Budget orientated**
   - **Risk orientated**
6. Set the start year of the plan.

   The default is first year of asset health result.
7. Set the number for years of the plan.

   The default setting is 20 years.
8. Set the failure probability threshold in which assets must be replaced, for example 99%.
9. Click **OK**.

Results

You can now create a report that compares the difference between the project to sustain the assets and the scenario.

Comparing the results of a scenario in the report view

In IBM Maximo Asset Performance Management for Energy & Utilities SaaS you can compare the results of a scenario against the existing scenarios and against the sustain project.

Procedure

1. Open the investment project with the scenario you want to compare.
2. Click **Report**.
3. You can compare the scenario against the sustain for risk and budget.
4. You can compare **Risk**, **Failure probability**, **Replacement** and **Cost** for the different scenarios, click the each row in the visualization.

   The results are directly displayed in the comparison visualization.

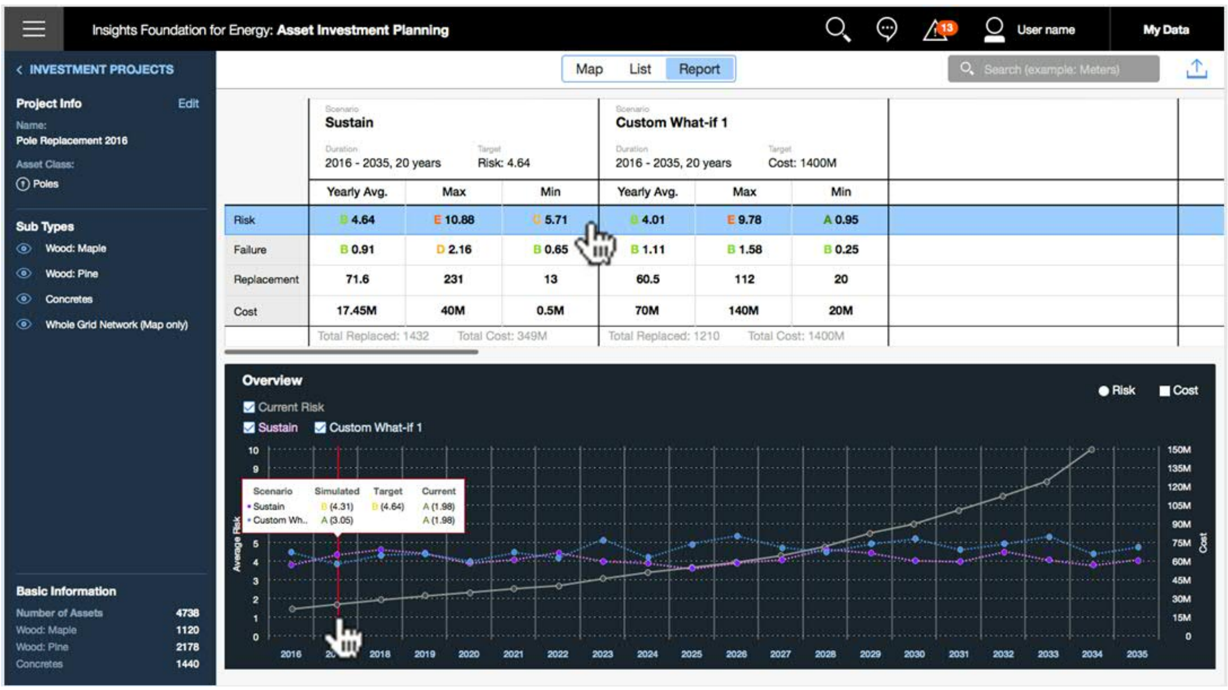


Figure 85. The comparison of results for Risk

Chapter 7. Using Connectivity Model

The Connectivity Model gives feedback to managers, data analysts and grid operators as to the accuracy of the phase and connectivity details of the network.

The Connectivity Model application ensures:

- The model is always current: The user is sure that the connectivity model represents the latest analytic results.
- The connectivity and reliability of information is quickly ascertained: The user can fully understand the state of assets in the network within 5 minutes or less.
- The network details are simplified: The user can focus on the assets and connections that are important to them.
- The analysis details can be reported and shared: The users can collect the necessary charts or maps, and download and share them with the key stakeholders.
- The reports for customer to phase and customer to transformer mapping are accurate.
- The ability to fix connectivity records without the time and expense of sending crews into the field.

Overview of the data flow

The data flow contains four parts: the data preparation for the Connectivity Model application, the Extract, Transform, and Load (ETL) process, the data validation, and the analysis of the data.

The details of the ETL module are described in this section.

You prepare you own data in `.csv` format and then load the `.csv` files into the HDFS.

After preparing the raw data, you run the ETL and validation module to generate the data for the analysis.

Preparing the data for the ETL module

Explains how you prepare the raw data for the ETL module on the Jupyter node as a .csv file.

The raw data is created as a .csv files, the raw data folder structure is shown as the following:

- connectivity
- electric_station
- exclude_time
- feeder_root
- meter
- meter_load
- meter_voltage
- overhead_cable
- scada
- scada_load
- substation_region
- transformer
- underground_cable

The prepared data includes master data and reading data. The master data contains the assets in the power grid, and the reading data contains the measurement readings, for example: voltage and load.

As the ETL module supports data in .csv format, by using the csv format, you can edit and update the master data and reading data. You can also incrementally add reading data to the ana store. You can define multiple csv files in each folder. The ETL module does not support sub-folders.

The format of the .csv files is given as follows:

Master data

The master data contains the data of the assets.

Table 22. Connectivity master data

Column name	Type	Description	Constraints
assetId	String	asset ID	Primary Key, Unique, Not NULL.
substation	String	ID of substation region to where electric station belongs.	Foreign Key, must be a valid substation in the substation region table.
type	String	The asset type.	
phase	String	The phase of the asset.	
numberOfPhases	String	The amount of phases.	
normalStatus	String	The normal status.	
node1	String	The node 1.	
node2	String	The node 2.	

Table 23. Electric station master data

Column name	Type	Description	Constraints
id	String	ID of the electric station	Primary Key, Unique, Not NULL.
substation	String	ID of substation region to where electric station belongs.	Foreign Key, must be a valid substation in the substation region table.
geometry	String	The geometry of the electric station.	Must be in a valid WKT POLYGON, in WSG84 projection.

Table 24. Exclude time master data

Column name	Type	Description	Constraints
feeder	String	ID of the feeder.	Primary Key. Must be a valid feeder in the feeder table or feeder_group in the feeder_group table
type	String	Type of analysis.	Primary Key. The candidate value include *Load*, *Voltage*, *Voltage_with_scada*.

Table 24. Exclude time master data (continued)			
Column name	**Type**	**Description**	**Constraints**
startTime	String	Timestamp, in the format of yyyy-MM-ddThh:mm:ss.sss.	Primary Key. The time range to be excluded from the analysis with the startTime being the start time of the range (inclusive).
endTime	String	Timestamp, in the format of yyyy-MM-ddThh:mm:ss.sss.	Primary Key. The time range to be excluded from the analysis with the endTime being the end time of the range (exclusive).

Table 25. Feeder root master data			
Column name	**Type**	**Description**	**Constraints**
feeder	String	ID of the feeder where the meter is connected	Foreign Key, must be a valid feeder in the feeder table.
rootAsset	String	The root asset.	

Table 26. Meter master data			
Column name	**Type**	**Description**	**Constraints**
id	String	ID of the meter	Primary Key, Unique, Not NULL.
substation	String	ID of substation region where the meter belongs to.	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the meter is connected	Foreign Key, must be a valid feeder in the feeder table.
transformer	String	ID of the transformer	Foreign Key, must be valid transformer in transformer table.
phase	String	The phase of meter	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	Geospatial geometry in WKT format, should be a point.	Must be a valid WKT POINT, in WSG84 projection.

Table 27. Overhead cable master data			
Column name	**Type**	**Description**	**Constraints**
id	String	ID of the overhead cable	Primary Key, Unique, Not NULL.
substation	String	ID of the substation region where the overhead cable belongs to.	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the overhead cable is connected.	Foreign Key, must be a valid feeder in the feeder table.
phase	String	The phase of the overhead cable.	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	The geometry of the overhead cable.	Must be a valid WKT MULTILINESTRING, in WSG84 projection.

Table 28. SCADA master data			
Column name	**Type**	**Description**	**Constraints**
assetId	String	ID of the feeder.	Primary Key, Unique, Not NULL.
measurement	String	Type of analysis	Primary Key Candidate value must include *Load*, *Voltage*, *Voltage_with_scada*
scadaId	String	ID of SCADA.	

Table 29. Substation region master data			
Column name	**Type**	**Description**	**Constraints**
id	String	ID of the substation region.	Primary Key, Unique, Not Null.
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection

Table 30. Transformer master data			
Column name	**Type**	**Description**	**Constraints**
id	String	ID of the transformer.	Primary Key, Unique, Not NULL.

Table 30. Transformer master data (continued)			
Column name	**Type**	**Description**	**Constraints**
substation	String	ID of the substation region where the transformer belongs to.	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the transformer is connected.	Foreign Key, must be a valid feeder in the feeder table.
phase	String	The phase of the transformer	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
kva	Int	The kilovolt-amps.	
voltage	Int	The voltage.	
geometry	String	Geospatial geometry in WKT format, should be a point.	Must be a valid WKT POINT, in WSG84 projection.

Table 31. Underground cable master data			
Column name	**Type**	**Description**	**Constraints**
id	String	ID of the underground cable.	Primary Key, Unique, Not NULL.
substation	String	ID of the substation region where the underground cable belongs to,	Foreign Key, must be a valid substation in the substation region table.
feeder	String	ID of the feeder where the underground cable is connected.	Foreign Key, must be a valid feeder in the feeder table.
phase	String	The phase number.	Must be a valid phase code from 1 to 7. For example: 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	The geometry of the underground cable.	Must be a valid WKT MULTILINESTRING, in WSG84 projection.

Reading data

The reading data contains the current record values, for example, the voltage values and the load values. The reading data is used for the analysis.

Note: These notes are for all reading data tables:

1. The timestamp must be in same time zone as the corresponding load table. For example both load tables and reading tables must either use UTC or use the local time zone.
2. The time interval between two timestamps must be fixed and be the same as time interval in corresponding load table. For example, if the time interval is 1 hour, then all readings in the meter load table and feeder load table should use 1 hour interval.
3. The timestamp should align with timestamp in the corresponding load table. For example, if the feeder load has a timestamp 9:00 then the meter load timestamp should be the same.
4. The timestamp represents the end edge. For example, if the timestamp is 10:00 and interval used is 1 hour, the kWh is the load between 9:00-10:00.

Table 32. Feeder load reading data

Column name	Type	Description	Constraint
scadaId	String	ID of the feeder	Derived
timestamp	String	Timestamp in the format `yyyy-MM-ddThh:mm:ss.sss`.	Expected in UTC
kwh1	Double	Load of phase A.	The value represents the aggregated active load in the past time interval. All values must be in units of kWh not MWh. The gap between feeder load and aggregated meter load should less than 10% otherwise the accuracy is affected. The kwh1 is load of phase A that maps to phase code 0b100=4.
kwh2	Double	Load of phase B.	
kwh3	Double	Load of phase C.	

Table 33. Meter load reading data

Column name	Type	Description	Constraint
meterId	String	ID of the meter	Primary Key, Foreign Key Must be a valid meter in the meter table
timestamp	String	Timestamp of reading data, in the format of `yyyy-MM-ddThh:mm:ss.sss`	

Table 33. Meter load reading data (continued)			
Column name	**Type**	**Description**	**Constraint**
kwh	Double	Load	Value represent aggregated active load of all 3 phases in past interval. All values in unit to KWH.

Table 34. Feeder voltage reading data			
Column name	**Type**	**Description**	**Constraint**
scadaId	String	ID of the SCADA tag.	Derived
timestamp	String	Timestamp in the format `yyyy-MM-ddThh:mm:ss.sss`.	Expected in UTC
volt1	Double	Phase A voltage	The value represents the average voltage in the past time interval. The value must be normalized to the same voltage level. For example, if some meters are 120v, and others are 240v, when the feeder voltage is 1kV, the values is normalized to the minimal voltage level, that is: Value * 120 / 240. When ch1volt, ch2volt, ch3volt represent the voltage of 3 phases, ch1volt maps to phase A, ch2volt maps to phase B, and ch3volt maps to phase C.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	

Table 35. Meter voltage reading data			
Column name	**Type**	**Description**	**Constraint**
meterId	String	ID of the meter	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table, or feeder group in the feeder_group table.
timestamp	String	Timestamp yyyy-MM-ddThh:mm:ss.sss.	
volt1	Double	Phase A voltage	The value represents the average voltage in the past interval. The value must be normalized to the same voltage level. For example, if some meters are 120v, and others are 240v, when the feeder voltage is 1kV, the values is normalized to the minimal voltage level, that is: Value * 120 / 240. Channels ch1, ch2 and ch3 need not be the same value. For a single phase meter, only one channel should have a value. For two phases meters, only two channels should have values. For 3 phases meters, all channels must have values.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	

The ETL process

The ETL is the major process of the ETL module and prepares the required data for the next procedure of the analysis.

The ETL module has three stores:

- The raw store stores the raw data from the user.
- The tmp store stores the results of ETL.
- The ana store stores the data required by analysis. After the optional validation, the data under `tmp` store should be moved to the ana store so that the analysis can be executed.

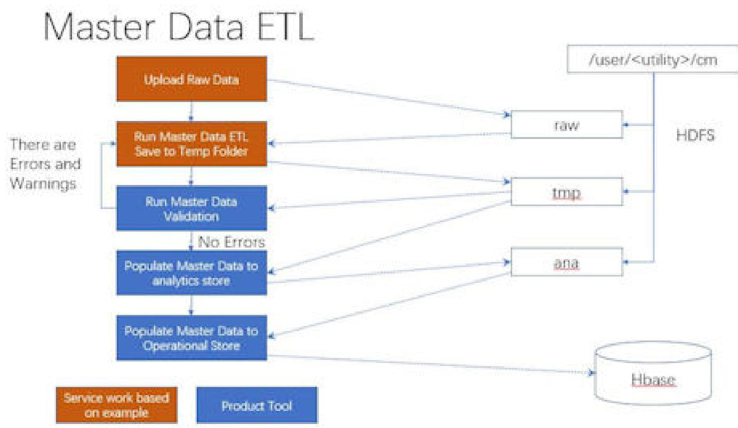


Figure 86. Master data in the ETL module

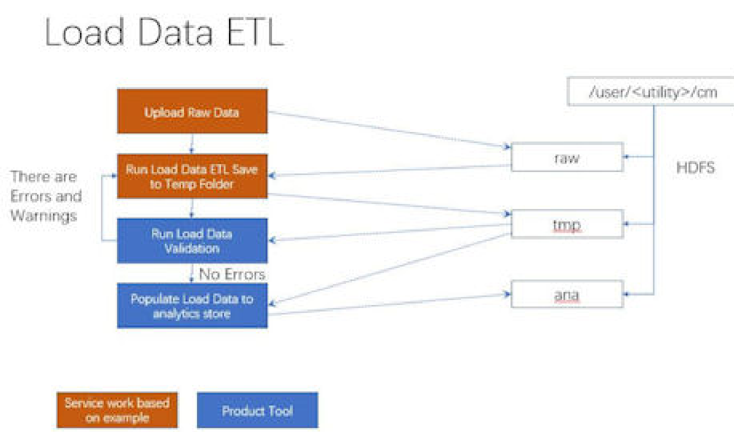


Figure 87. Load data to the ETL module

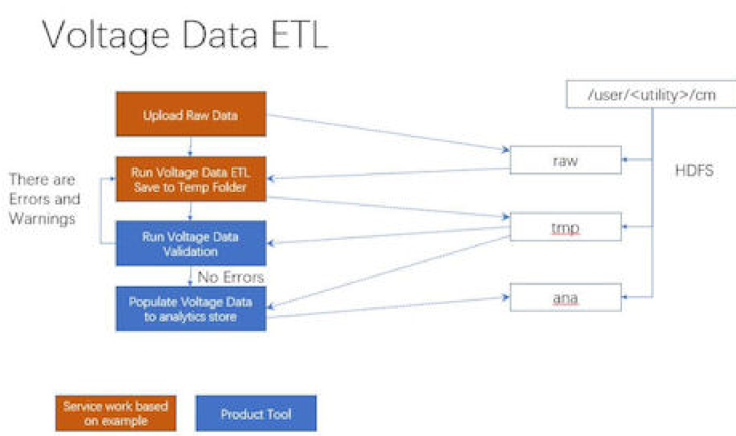


Figure 88. Voltage data in the ETL module

The ETL process includes the group feeder ETL, the exclude time range ETL, the master data ETL, and the reading data ETL. You can find the files on the Jupyter node:

The configure file: `/home/<utility_id>/conf/`

The python source code: `/home/<utility_id>/etl`

The shell files: `/home/<utility_id>/etl`

The ETL results are written to a tmp store, and not into ana store immediately. After the validation procedure, the data is moved from the tmp store to the ana store.

The output files of ETL are in parquet format, use the pyspark to read and show the data of the parquet format. For example, create a `readTable.py` file in the `/home/<utility_id>/etl` and run this python file.

```
etl_utility = ETLUtil()
(sc,sqlContext) = etl_utility.init("readTable")
sqlContext.read.parquet(hdfs:///user/cm_sample/cm/tmp/feeder_load).show()
```

The results are:

feeder	timestamp	kwh1	kwh2	kwh3	year	week
NIXON8943	2012-01-05 04:00:...	31329.39453125	29201.60546875	38259.37109375	2012	1
NIXON8943	2012-01-07 22:00:...	27076.27734375	29999.46484375	41325.609375	2012	1
NIXON8943	2012-01-10 02:00:...	35938.890625	27460.6328125	42180.26953125	2012	2
NIXON8943	2012-01-06 09:00:...	33440.50390625	31753.73046875	38686.05859375	2012	1
NIXON8943	2012-01-08 03:00:...	34617.5	33180.6171875	38329.6484375	2012	1

Figure 89. The results of the ETL process

Group feeder ETL

The group feeder ETL is used to support multiple feeders that belong to one group. The group feeder ETL defines the feeders for the group and how the results are mapped from the SCADA raw data.

Table 36. The output data format of the group feeder ETL

Column name	Type	Description	Constraints
feeder_group	String	ID of feeder group	Primary Key. Used to group feeders when they share the same SCADA reading
type	String	The type of analysis.	Primary Key. The candidate value includes *Load*, *Voltage*, *Voltage_with_scada*. **Note:** all feeders in same group are to be analyzed in a single batch. .
feeder	String	ID of the feeder.	Primary Key

Exclude time range ETL

The exclude time range is used to support the analysis when the you want to exclude data in a specific time range.

You specify the time range in the raw data as a csv file. And the exclude time range ETL generates the exclude time range table for the analysis.

The input data:

- `/user/<utility>/cm/raw/exclude_time`
- `/user/<utility>/cm/raw/scada`
- `/user/<utility>/cm/tmp/feeder_group`

The output data: `/user/<utility>/cm/tmp/exclude_time`

Table 37. The output data format of the exclude time range ETL			
Column name	**Type**	**Description**	**Constraints**
`feeder`	String	ID of feeder	Primary Key Must be a valid feeder in the feeder table or feeder_group in the feeder_group table.
`type`	String	Type of analysis.	Primary Key Candidate value includes *Load*, *Voltage*, *Voltage_with_scada*.
`startTime`	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss.	Primary Key The time range to be excluded from the analysis with the startTime being the start time of the range (inclusive).
`endTime`	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss.	Primary Key. The time range to be excluded from the analysis with the endTime being the end time of the range (exclusive).

Master data ETL

The master data ETL process converts raw data into the master data for the user interface. The master data ETL contains:

- substation region
- electric station
- feeder
- lateral
- transformer
- meter

The details are:

for a substation region:

The input data: `/user/<utility_id>/cm/raw/substation_region`

The output data: `/user/<utility_id>/cm/tmp/substation_region`

Table 38. The output data format of the substation region			
Column name	**Type**	**Description**	**Constraints**
`substation`	String	ID of substation region	Primary Key, Unique, Not Null

Table 38. The output data format of the substation region (continued)			
Column name	**Type**	**Description**	**Constraints**
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection.

electric station:

The input data: `/user/<utility>/cm/raw/electric_station`

The output data: `/user/<utility>/cm/tmp/electric_station`

Table 39. The output data format of the electric station

Column name	Type	Description	Constraints
substation	String	The ID of the substation region that the station belongs to.	Foreign Key, must be a valid substation in the substation region table
electricStation	String	ID of the electric station.	Primary Key, Unique, Not NULL
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection.

feeder:

The input data:

- `/user/<utility>/cm/raw/transformer`
- `/user/<utility>/cm/raw/overhead_cable`
- `/user/<utility>/cm/raw/underground_cable`

The output data: `/user/<utility>/cm/tmp/feeder`

Table 40. The output data format of the feeder

Column name	Type	Description	Constraints
feeder	String	The ID of the feeder.	Primary Key, Unique, Not NULL
substation	String	The substation ID.	Foreign Key, must be a valid substation in the substation region table
geometry	String	Geospatial geometry in WKT format, should be a polygon.	Must be valid WKT POLYGON. In WSG84 projection.

lateral:

The input data: `/user/<utility>/cm/raw/lateral`

The output data: `/user/<utility>/cm/tmp/lateral`

Table 41. The output data format of the lateral			
Column name	**Type**	**Description**	**Constraints**
substation	String	The substation ID.	Foreign Key, must be a valid substation in the substation region table
feeder	String	The ID of the feeder.	Foreign Key, must be a valid feeder in the feeder table
lateral	String	ID of the lateral	Primary Key, Unique, Not NULL
phase	Int	The phase of the lateral	Must be a valid phase code, from 1 to 7 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	Geospatial geometry in WKT format, should be a line.	Must be valid WKT LINESTRING. In WSG84 projection.
out	String	The cable type.	

transformer:

The input raw data: `/user/<utility>/cm/raw/transformer`

The output data: `/user/<utility>/cm/tmp/transformer`

Table 42. The output data format of the transformer			
Column name	**Type**	**Description**	**Constraints**
substation	String	The ID of the substation region where the feeder belongs to.	Foreign Key, must be a valid substation in the substation region table
feeder	String	The ID of the feeder where the transformer connected belongs to.	Foreign Key, must be a valid feeder in the feeder table
lateral	String	The ID of the lateral.	
transformer	String	The ID of the transformer.	Primary Key, Unique, Not NULL
phase	Int	The phase of the transformer.	Must be a valid phase code, from 1 to 7 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
geometry	String	Geospatial geometry in WKT format, should be a point.	Must be valid WKT POINT. In WSG84 projection.
out	String	The transformer type.	

Table 42. The output data format of the transformer (continued)			
Column name	**Type**	**Description**	**Constraints**
`kva`	String	The kVA level.	
`voltage`	String	the voltage level.	

meter:

The input data: `/user/<utility>/cm/raw/meter`

The output data: `/user/<utility>/cm/tmp/meter`

Table 43. The output data format of the meter			
Column name	**Type**	**Description**	**Constraints**
`substation`	String	The ID of the substation region where the feeder belongs to.	Foreign Key, must be a valid substation in the substation region table
`feeder`	String	The ID of the feeder where the transformer connected belongs to.	Foreign Key, must be a valid feeder in the feeder table
`lateral`	String	The ID of the lateral.	
`transformer`	String	The ID of the transformer where the meter is connected.	Primary Key, Unique, Not NULL
`phase`	Int	The phase of the meter.	Must be a valid phase code, from 1 to 7 0b100=4=A, 0b010=2=B, 0b001=1=C, 0b110=6=AB.
`geometry`	String	Geospatial geometry in WKT format, should be a point.	Must be valid WKT POINT. In WSG84 projection.

Reading data ETL

The reading data ETL process uses the reading data, including voltage data and load data and appends the year and week of the reading data timestamp. The year and week data is used as the partition for the incremental saving of the output of the reading data ETL.

In reading data ETL, you can use the command parameters to control the time range of the reading data ETL. The reading data is incrementally saved as the figure shows:

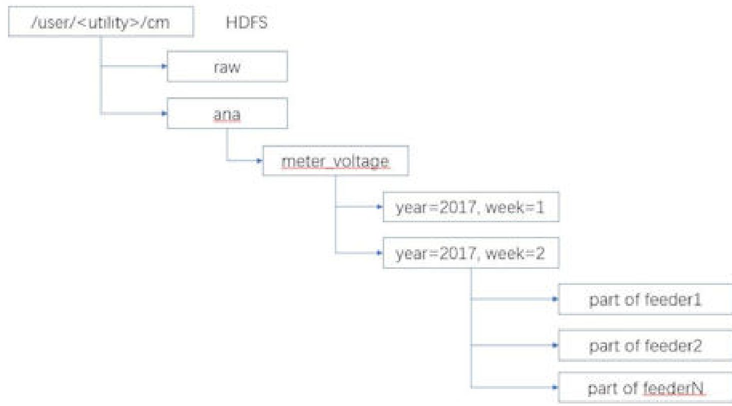


Figure 90. Incremental saves

Note: These notes are for all ETL reading data tables.

1. The timestamp must be in same time zone as the corresponding load table. For example both load tables and reading tables must either use UTC or use the local time zone.
2. The time interval between two timestamps must be fixed and be the same as time interval in corresponding load table. For example, if the time interval is 1 hour, then all readings in meter load table and feeder load table should use 1 hour interval.
3. The timestamp should align with timestamp in the corresponding load table. For example, if the feeder load has a timestamp 9:00 then the meter load timestamp should be the same.
4. The timestamp represents the end edge. For example, if the timestamp is 10:00 and interval used is 1 hour, the kWh is the load between 9:00-10:00.

The ETL reading data files are in the structure:

- feeder_load
- meter_load
- feeder_voltage
- meter_voltage

The input raw data: `/user/<utility>/cm/raw/<etl reading data>`

The output data: `/user/<utility>/cm/tmp/<etl reading data>`

To run both the voltage and load reading data ETL, use the command

```
/home/<utility_id>/etl/py_ReadingDataETL.sh
```

To run the voltage reading data ETL, use the command

```
/home/<utility_id>/etl/py_VoltageDataETL.sh
```

To run the load the reading data ETL, use the command

```
/home/<utility_id>/etl/py_LoadDataETL.sh
```

Table 44. The output data format of the feeder load ETL reading data			
Column name	**Type**	**Description**	**Constraint**
feeder	String	ID of the feeder	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table or feeder_group in the feeder_group table.
timestamp	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss .	
kwh1	Double	Load of phase A.	The value represents the aggregated active load in past interval. All values are in units of kWh (not MW h) Gap between feeder load and aggregated meter load should less than 10%, otherwise will impact accuracy. The kwh1 is load of phase A that maps to phase code 0b100=4.
kwh2	Double	Load of phase B.	
kwh3	Double	Load of phase C.	

Table 45. Meter load ETL reading data			
Column name	**Type**	**Description**	**Constraint**
feeder	String	ID of the feeder.	Primary Key, Foreign Key. The feeder should be a valid feeder in the feeder table, or feeder_group in the feeder_group table.
meter	String	ID of the meter	Primary Key, Foreign Key Must be a valid meter in the meter table
timestamp	Timestamp	Timestamp, yyyy-MM-ddThh:mm:ss.sss	

Table 45. Meter load ETL reading data (continued)			
Column name	**Type**	**Description**	**Constraint**
kwh	Double	kWh load	Value represent aggregated active load of all 3 phases in past interval. All values are in units of kWh.
year	Int	Year of the timestamp.	
week	Int	Week of the timestamp.	

Table 46. Feeder voltage ETL reading data			
Column name	**Type**	**Description**	**Constraint**
feeder	String	ID of the feeder	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table, or feeder group in the feeder_group table.
timestamp	Timestamp	Timestamp yyyy-MM-ddThh:mm:ss.sss.	
volt1	Double	Phase A voltage	The value represents the average voltage in past time interval. The value must be normalized to the same voltage level. For example, if some meters are 120v, and others are 240v, when the feeder voltage is 1kV, the values is normalized to the minimal voltage level, that is: Value * 120 / 240 When ch1volt, ch2volt, ch3volt represent the voltage of 3 phases, ch1volt maps to phase A, ch2volt maps to phase B, and ch3volt maps to phase C.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	
year	Int	Year of the timestamp.	
week	Int	Week of the timestamp	

Table 47. Meter voltage ETL reading data			
Column name	**Type**	**Description**	**Constraint**
feeder	String	ID of the feeder	Primary Key, Foreign Key The feeder should be a valid feeder in the feeder table, or feeder group in the feeder_group table.
meter	String	ID of the meter	Primary Key, Foreign Key must be valid meter in meter table
timestamp	String	Timestamp yyyy-MM-ddThh:mm:ss.sss.	
volt1	Double	Phase A voltage	The value represents the average voltage in the past interval. Channels ch1, ch2 and ch3 need not be the same value. For a single phase meter, only one channel should have a value. For two phases meters, only two channels should have values. For 3 phases meters, all channels must have values.
volt2	Double	Phase B voltage	
volt3	Double	Phase C voltage	
year	Int	Year of the timestamp.	
week	Int	Week of the timestamp.	

Validation of the ETL process (optional process)

The validation module gives information about the data from the ETL process and is an optional process. You can make any necessary corresponding changes to the input data before validation.

The data after ETL procedure is saved in the tmp store on HDFS. You can choose to run this validation procedure.

Any errors must be corrected otherwise it might affect the analysis results. You can also define customized validation rules in this module.

The files are located on the Jupyter node:

- The configure file: `/home/<utility>/conf/`
- The python source code: `/home/<utility>/bin`
- The shell files: `/home/<utility>/bin`

Master data validation

Validation of the master data changes according to the asset.

For different assumptions of the data, the validation module gives different validation levels, warning and error.

The input data of validation is from: /user/<utility>/cm/tmp/<asset entity>.

Voltage data validation

Validation of the voltage data focuses on the timestamps and the voltage values levels.

The timestamps describe the process variation of the voltage data that is required for the phase analysis in the analysis module.

The level of the voltage values describes the voltage level of different meters in the feeders. In practice, the voltage drop between the meters and their feeder should not be large.

The input data of validation is from: `/user/<utility>/cm/tmp/feeder_voltage` or `/user/<utility>/cm/tmp/meter_voltage`.

- Basic validation
- Timestamp validation - the validation of the fixed time interval and the validation of timestamp alignment between meters and their feeder. The fixed time interval is a basic assumption that the SCADA device provides a fixed time interval. The timestamp alignment of the feeder and the meters are not aligned. This is the basis of the analysis as it compares the variation trend of the feeder and the variation trend of meters in the feeder.
- Voltage data level validation - the validation of the voltage data three parts: the feeder voltage level, the meter voltage level, and the voltage level between the meter and feeder. In practice, the voltage level of the feeder and the meter must be the same. The population standard deviation value is used as the index to verify the voltage level. The user can design different indices to check the voltage level.

Load data validation

Validation of the load data focuses on the timestamps and the load values level.

The timestamps describe the variation process of the load data that is used in the phase analysis in the analysis module. The load values level describes the load level of different meters in feeders. In practice, the load gaps between meters and their feeders should not be large.

The input data of validation is from: `/user/<utility>/cm/tmp/feeder_load` or `/user/<utility>/cm/tmp/meter_load`

- Basic validation
- Timestamp validation - the validation of the fixed time interval and the validation of timestamp alignment between meters and their feeder. The fixed time interval is a basic assumption that the SCADA device provides a fixed time interval. The timestamp alignment of the feeder and the meters are not aligned. This is the basis of the analysis as it compares the variation trend of the feeder and the variation trend of meters in the feeder.
- Load data level validation - the validation of the load data level is in three parts: the feeder load level, the meter load level, and the load level between meter and feeder. In practice, the load level of the feeder and the meter must be the same. The population standard deviation value is used as the index to verify the load level. The user can design different indices to check the load level.

Preparing the data for the operational store

After ETL or ETL with validation, you move the data from the tmp store to the ana store, and then populate the operational store.

Move data from the tmp store to the ana store

After ETL or ETL with Validation, the you move the data from the tmp store to ana store on the data on HDFS.

The input raw data: /user/<utility>/cm/tmp/*

The output data: /user/<utility>/cm/ana/*

The master data will overwrite from .tmp to ana.

Populate the operational store

After the ETL procedure, the python script py_PopulateOperationalStore writes the master data into HBase.

The input raw data: /user/<utility>/cm/raw/feeder_voltage.

The analysis process of the data

Four analyzes are provided by the Connectivity Model application that ensures that the required data is made available.

The four analyzes are:

- Load based meter phase analysis
- Voltage based meter phase analysis
- Voltage with SCADA based meter phase analysis
- Lateral transformer phase analysis

Log files and the Analysis result

The log files are created on the Jupyter node under the directory of /home/<utility>/cm/logs/ once the scripts are run. Each type of analysis and persistence has its own sub-directory to keep its log file. The image shows the corresponding sub-directories.

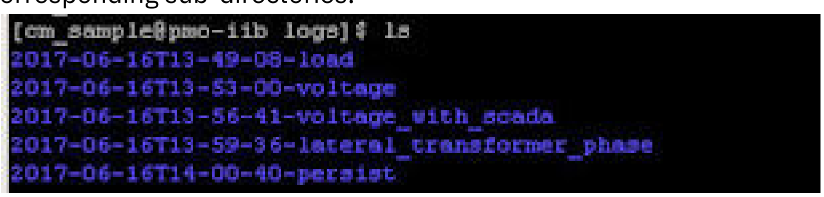


Figure 91. Sample log file directory structure

Each sub-directory name includes two parts: the timestamp and type.

You can check all the analysis results on HDFS: hdfs://user/<utility_id>/cm/<JOB_ID>/data/<type>

The following image shows the directory structure for the utility_id with JOB_ID of utility_id_sample_job.

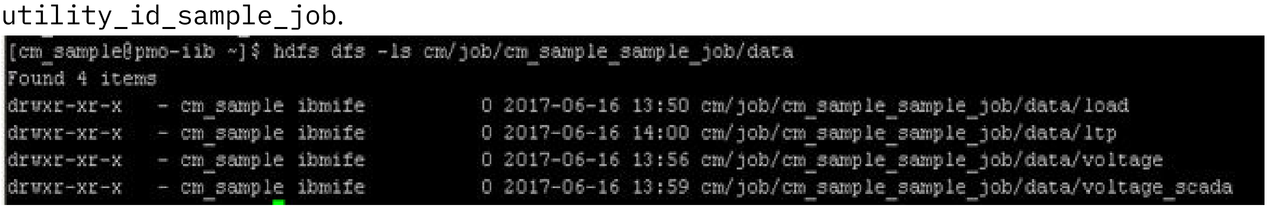


Figure 92. The directory structure on HDFS

The format of the analysis result for load, voltage and voltage with SCADA is described in the table:

Table 48. The result format for load, voltage, and voltage with SCADA		
Column name	**Type**	**Description**
feeder	String	ID of the feeder
meter	String	The ID of the meter
analysis_time	Timestamp	The timestamp when the analysis was run.
phase	Integer	The phase of the meter
confidence	Double	The confidence value

The format of the analysis result for a lateral transformer phase is described in the table:

Table 49. The result format for lateral transformer phase		
Column name	**Type**	**Description**
feeder	String	ID of the feeder.
lateral	String	ID of the lateral.
transformer	String	ID of the transformer
analysis_time	Timestamp	The timestamp when the analysis was run.
lateral_phase	Integer	The phase of the lateral.
transformer_phase	Integer	The phase of the transformer.
phase_match	Boolean	If the phase is matched or not.

Configuring the Connectivity Model application

Before you load the data for anaysis, the SMTP server, the ETL locations and the time duration for the ETL must be configured on the Jupyter node.

The items that can be configured are described as follows:

Table 50. The configurable elements, descriptions and sample			
Item	**Description**	**Relevant section**	**Sample**
smtp_type	The communication protocol to smtp server; the valid values could be: tls, ssl.	"Automating the data flow" on page 195	tls
smtp_server	The SMTP server address. If a different smtp_type is specified, then the smtp_server should be changed accordingly.	"Automating the data flow" on page 195	smtp.gmail.com:587
smtp_login	The login account used to verify the SMTP server.	"Automating the data flow" on page 195	

Table 50. The configurable elements, descriptions and sample (continued)			
Item	**Description**	**Relevant section**	**Sample**
smtp_password	The corresponding password of the login account.	"Automating the data flow" on page 195	
cm_mail_to	Email addresses to receive the automation flow mail. If there are multiple addresses, then separate each one with a comma.	"Automating the data flow" on page 195	
etl_input_path	The ETL input path where the raw csv data is stored.	"Loading the master data for the Connectivity Model application" on page 190, "Loading the reading data for the Connectivity Model application" on page 191	hdfs:///user/ cm_sample/cm/raw
etl_staging_path	The ETL intermediate path where the parquet files are stored.	"Loading the master data for the Connectivity Model application" on page 190, "Loading the reading data for the Connectivity Model application" on page 191	hdfs:///user/ cm_sample/cm/tmp
etl_output_path	The ETL final path where the parquet files are moved to after verification.	"Loading the master data for the Connectivity Model application" on page 190, "Loading the reading data for the Connectivity Model application" on page 191	hdfs:///user/ cm_sample/cm/ana
etl_connectivity_used	The indicator to judge whether the connectivity data is available. By default it is FALSE. If you have `connectivity.csv` table available, you must change the default value to "TRUE".	"Loading the master data for the Connectivity Model application" on page 190	

Table 50. The configurable elements, descriptions and sample (continued)			
Item	**Description**	**Relevant section**	**Sample**
master_etl_spark_defaults_conf	The SPARK default configuration file for the master data ETL. The file name must be specified if the resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file `/usr/hdp/current/spark-client/conf/spark-defaults.conf`	"Loading the master data for the Connectivity Model application" on page 190	
reading_etl_spark_defaults_conf	The SPARK defult configuration file for the reading data ETL. The file name must be specified if the resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file `/usr/hdp/current/spark-client/conf/spark-defaults.conf`	"Loading the reading data for the Connectivity Model application" on page 191	
analysis_load_spark_defaults_conf	The SPARK default configuration file for load analysis. The file name must be specified if the resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file `/usr/hdp/current/spark-client/conf/spark-defaults.conf`	"The analysis process of the data" on page 184	

Table 50. The configurable elements, descriptions and sample (continued)			
Item	**Description**	**Relevant section**	**Sample**
analysis_load_ duration	The duration of the load analysis in days. The value is changed if the load reading has a different duration.	"The analysis process of the data" on page 184	30
analysis_load_ until_time	The end of duration time for the load reading data used for the load analysis. The value is changed if thenew load reading data has a different time value.	"The analysis process of the data" on page 184	2017-10-09T00:00:00
analysis_load_ resource_guard	The resource guard used to control the resource consumption when executing the load analysis. The default value of 0 that indicates there is no limits to the resource consumption. If there are many items in the reading data, then the value should be set based on the number of the meters belonging to the feeder. For example, if there are 5000 meters on one feeder, then a number 20% greater than 5000 should be set.	"The analysis process of the data" on page 184	6000
analysis_voltage_ feeders	The feeder list file that contains all of the feeders to be analyzed by the voltage analysis.	"The analysis process of the data" on page 184	../conf/ feeder_voltage.lst

Item	Description	Relevant section	Sample
analysis_voltage_ with_scada_spark_ defaults_conf	The SPARK default configuration file for the voltage with SCADA analysis. The file name must be specified if resource allocation parameters need to be adjusted. By default it is empty and the system level configuration is used. If you need to adjust the parameters, please refer to the sample file `/usr/hdp/current/spark-client/conf/spark-defaults.conf`	"The analysis process of the data" on page 184	
analysis_voltage_ with_scada_duration	The duration of the voltage with SCADA analysis in days. The value must be changed if the voltage reading has a different duration.	"The analysis process of the data" on page 184	30
analysis_voltage_ with_scada_until_time	The end of duration time for voltage reading data used to do the voltage with SCADA analysis. It must be changed if the new voltage reading data has a different time value.	"The analysis process of the data" on page 184	2017-10-09T00:00:00
analysis_voltage_ with_scada_feeders	The feeder list file that contains the details of all the feeders to be analyzed by voltage analysis.	"The analysis process of the data" on page 184	../conf/feeder_voltage_ with_scada.lst
analysis_lateral_ transformer_phase_ feeders	The feeder list file that contains all details of the feeders to be analyzed by lateral transformer phase analysis.	"The analysis process of the data" on page 184	../conf/feeder_lateral_ transformer_phase.lst

Table 50. The configurable elements, descriptions and sample (continued)

Encrypting the SMTP password

You should provide and encrypt the SMTP password in the `tenant.cfg` file.

Procedure

1. Log into the Jupyter node.

2. Edit the /home/<utility_id>/conf/tenant.cfg file to provide the plain text value for the smtp_password item.
3. Go to the /home/<utility_id>/automation directory.
4. Run the command:

```
./encrypt.sh ../conf/tenant.cfg smtp_password aes
```

   The plain text password in the tenant.cfg is now encrypted.

Loading data to the Connectivity Model

When you have completed the configuring of the Connectivity Model, you must load your data into HDFS (Hadoop Distribution File System). It should be executed on Jupyter node.

Loading the master data for the Connectivity Model application

The following steps are give as an example of how to do the extracting and loading of the master data.

About this task

In this example set of steps, the **utility_id** is **utility1**, the work folder is /home/utility1, and the master data is in the /home/utility1/raw folder on the Jupyter node.

Procedure

1. Log into the Jupyter node.
2. To upload the source data to the HDFS, type the command:

```
hdfs dfs -put /home/<utility_id>/raw /user/utility_id/cm
```

3. To support multiple feeders that belong to one group, use the command:

```
/user/<utility_id>/cm/tmp/feeder_group
/user/<utility_id>/cm/raw/scada
```

   Note: The group feeder ETL defines the feeders for the group and how the results are mapped from the SCADA raw data.
4. To generate feeder_group in the tmp folder, use the command:

```
/home/<utility_id>/etl/py_GroupFeederETL.sh
```

5. To exclude a specific time range from the data, use the command:

```
/user/<utility_id>/cm/raw/exclude_time
```

6. To generate exclude_time in the tmp'll make the chage for steps 7 and 8 folder, use the command:

```
/home/<utility_id>/etl/py_ExcludeTimeRangeETL.sh
```

7. To translate the master data from the .csv format to the Parquet file format type the command:

```
py_MasterDataETL.sh
```

   For example: You replace the <utility_id> by utility1 in this example.

```
/home/<utility_id>/etl/py_MasterDataETL.sh
```

8. As an optional step for validation of the master data, type the command:

```
/home/<utility_id>/etl/py_MasterDataValidation.sh
```

Loading the reading data for the Connectivity Model application

The following steps are give as an example of how to do the extracting and loading of the reading data.

Before you begin

You must complete the "Loading the master data for the Connectivity Model application" on page 190 before you can load the reading data to the Connectivity Model application.

About this task

In this example set of steps, the **utility_id** is **utility1**, the work folder is `/home/utility1`, and the master data is in the `/home/utility/raw` folder.

Procedure

1. Log into the Jupyter node.
2. Load the reading data from the .csv format to the Parquet format and generate mapping parquet files, type the command:

   ```
   py_LoadDataETL.sh
   ```

   Example 1, to load all data and overwrite the load data in the Parquet folder: you replace the <utility_id> with actual utility1 in this example.

   ```
    /home/<utility_id>/etl/py_LoadDataETL.sh
   ```

   Example 2, to load specific data and to append the existing load data in Parquet folder: you replace the <utility_id> with utility1 and the <filename> with the actual file name in this example.

   ```
   /home/<utility_id>/etl/py_LoadDataETL.sh
   ```

   The system writes to `LoadDataETL.out` and system errors write to `LoadDataETL.err`.
3. To extract, transfer, and load the voltage reading data from the input .csv format into standard parquet format and generate mapping parquet files, type the command:

   ```
   py_VoltageDataETL.sh <csvPath> <overwrite>
   ```

   Example 1, extract, transfer and load all voltage data, and overwrite the existing voltage data in the Parquet folder, you replace the <utility_id> with actual value.

   ```
   /home/<utility_id>/etl/py_VoltageDataETL.sh
   ```

   Example 2, extract, transfer, and load specific voltage data and to append the existing voltage data in parquet folder, you replace the <utility_id> with utility1 and the <filename> with actual file name in this example.

   ```
   /home/<utility_id>/etl/VoltageDataETL.sh
   ```

   The system writes to `VoltageDataETL.out` and system error write to `VoltageDataETL.err`.
4. After you load the reading data, you can validate it.
   - To validate load reading data, type the command:

     ```
     /home/<utility_id>/etl/py_LoadDataValidation.sh
     ```

   - To validate the voltage reading data, type the command:

     ```
     /home/<utility_id>/etl/py_VoltageDataValidation.sh
     ```

   - To specify a time duration for the reading data, the format is:

```
/home/<utility_id>/etl/py_<data_type>DataValidation.sh -s 2012-01-01 -e 2012-02-01
```

Populating the master to the operational store

After you have loaded the master and reading data, you need to populate the data to the operational store.

Procedure

1. Log into the Jupyter node.
2. Move the data in the tmp store to the ana store with the command:

   ```
   /home/<utility_id>/etl/py_MoveTmpDataToAnaStore.sh
   ```

3. To populate the master data to the `operational` store run the command:

   ```
   /home<utility_id>/etl/py_PopulateOperationalStore.sh
   ```

Administration of the Connectivity Model application

Before you can start an analysis in the Connectivity Model application, the tenant environment must be available and the variable `JOB_ID` must be set that is used to group all the following analysis tasks:

- Run an analysis on the Connectivity Model application.
- Populate the results in the operational store so that user interface can show the analysis results.

Tenant environment must be setup for the execution of the analysis.

From the Jupyter node, type the commands:

```
su – cm_sample
```

and for Kerberos authentication

```
kinit -kt /etc/security/keytab/cm_sample.keytab cm_sample@PMQ.IBM.COM
```

The `JOB_ID` is usually set as a date:

```
JOB_ID=2017-06-10
```

The corresponding configuration items for the analysis in the tenant configuration file should be adjusted according to the reading data which can be different each time when the analysis is run.

```
###
### load analysis items
###
analysis_load_duration=30 #days
analysis_load_until_time=`date +%Y-%m-%dT00:00:00`
#the end time of the analysis, change this value if
necessary to synchrnize with the current reading data
analysis_load_feeders=../conf/feeder_load.lst
#feeders to be analyzed

###
### voltage analysis items
###
analysis_voltage_duration=30 #days
analysis_voltage_until_time=`date +%Y-%m-%dT00:00:00`
#the end time of the analysis, change this value if necessary to synchrnize
with the current reading data
analysis_voltage_feeders=../conf/feeder_voltage.lst #feeders to be analyzed
analysis_voltage_sample_minutes=60 #analysis algorithm sample minutes
analysis_voltage_iterations=2
#analysis algorithm iterations

###
### voltage with scada analysis items
###
```

```
analysis_voltage_with_scada_duration=30 #days
analysis_voltage_with_scada_until_time=`date +%Y-%m-%dT00:00:00`
#the end time of the analysis, change this value if necessary to
synchronize with the current reading data
analysis_voltage_with_scada_feeders=../conf/feeder_voltage_with_scada.lst #feeders to be
analyzed
analysis_voltage_with_scada_sample_minutes=60
#analysis algorithm sample minutes
analysis_voltage_with_scada_iterations=2
#analysis algorithm iterations

###
### lateral transformer phase analysis items
###
analysis_lateral_transformer_phase_feeders=../conf/feeder_lateral_transformer_phase.lst
#feeders to be analyzed
```

Optimize the hardware configuration items to maximize performance.

By default, the following three items are not specified as the hardware capability is unknown in the deployment environment.

Specify these values of you environment so that your system resources can be fully utilized.

Note: The values can not exceed the largest configuration value of your system; otherwise the analysis will fail as the resource cannot be allocated.

```
cpu_cores=
driver_memory=
executor_memory=
```

Running the supplied Connectivity Model analyzes

Some environment variables are provided so that the configuration items in the tenant configuration file are not required to be modified each time the data is changed.

When the scale of the reading data is increased, a tuning process adapts the parameters, `cpu_core`, `driver_memory` and `executor_memory` to maximize the running of the analysis.

The environment variables for export are:

```
CPU_CORES
DRIVER_MEMORY
EXECUTOR_MEMORY
```

Export the environment variables for each analysis type:

Load analysis

If you do not want to use the default settings in the `tenant.cfg` file for load analysis, you can replace the settings by exporting the following environment variables:

```
ANALYSIS_LOAD_DURATION
ANALYSIS_LOAD_UNTIL_TIME
ANALYSIS_LOAD_FEEDERS
```

For example:

```
export ANALYSIS_LOAD_DURATION=90
export ANALYSIS_LOAD_UNTIL_TIME=2017-06-18T00:00:00
```

From the Jupyter node, type the command:

```
./bin/run_load_analysis.sh
```

Voltage analysis

If you do not want to use the default settings in the `tenant.cfg` file for voltage anaysis, you can replace the settings by exporting the following environment variables:

```
ANALYSIS_VOLTAGE_DURATION
ANALYSIS_VOLTAGE_UNTIL_TIME
ANALYSIS_VOLTAGE_FEEDERS
```

For example:

```
export ANALYSIS_VOLTAGE_DURATION=30
ANALYSIS_VOLTAGE_UNTIL_TIME=2017-1009T00:00:00
ANALYSIS_VOLTAGE_FEEDERS=../conf/feeder_voltage.lst
```

From the Jupyter node, type the command

```
./bin/run_voltage_analysis.sh
```

Voltage with SCADA analysis

If you do not want to use the default settings in the `tenant.cfg` for voltage with SCADA analysis, you can replace the settings by exporting the following environment variables:

```
ANALYSIS_VOLTAGE_WITH_SCADA_DURATION
ANALYSIS_VOLTAGE_WITH_SCADA_UNTIL_TIME
ANALYSIS_VOLTAGE_WITH_SCADA_FEEDERS
```

for example:

```
export ANALYSIS_VOLTAGE_WITH_SCADA_DURATION=30
ANALYSIS_VOLTAGE_WITH_SCADA_UNTIL_TIME=2017-10-09T00:00:00
ANALYSIS_VOLTAGE_WITH_SCADA_FEEDERS=../conf/feeder_voltage_with_scada.lst
```

From the Jupyter node, type the command:

```
./bin/run_voltage_with_scada_analysis.sh
```

Lateral transformer phase analysis

If you do not want to use the default settings in the `tenant.cfg` for lateral transformer phase analysis, you can replace the settings by exporting the following environment variable.

```
ANALYSIS_LATERAL_TRANSFORMER_PHASE_FEEDERS
```

For example:

```
export ANALYSIS_LATERAL_TRANSFORMER_PHASE_FEEDERS=../conf/feeder_lateral_transformer_phase.lst
```

From the Jupyter node, type the command:

```
./bin/run_lateral_transformer_phase_analysis.sh
```

Populating the analysis results to the operational store

After you have run the analysis is run, the results must be populated into the operational store so that user interface can show the results of the analysis.

Procedure

1. Log into the Jupyter node and go to the tenant home, for example:
   `/home/cm_sample`
2. Type the command:

```
./bin/persist_result.sh <type>
```

The *<type>* can be:

- *raw*: Populates the analysis result for the review purpose and user interface does not get the result until `persist` is called with the aggregate specified.
- *aggregate*: Submits the raw result so that the user interface shows the results.
- *all*: Both raw and aggregate results show in the user interface.

3. Verify the analysis results in the user interface. See the section "Using the connectivity model application" on page 205

Automating the data flow

To simplify the use of the Connectivity Model application, three flow scripts are delivered with IBM Maximo Asset Performance Management for Energy & Utilities SaaS

Before you begin

Before executing the flow scripts, the configuration items listed in the "Configuring the Connectivity Model application" on page 185 must be complete.

The incoming master and reading data must be prepared and be in the `/home/<utility>/staging` directory. The files must be in a .zip format and the names must start with `master_data_*.zip` and `reading_data_*.zip` respectively. For example:

```
master_data_2017-12-19.zip
```

and

```
reading_data_2017-12-19.zip
```

Procedure

1. Log into the notebook node as a tenant user with access rights to HDFS and Hbase and start the master data flow.
   a) Open the `/home/<utility>/automation` directory.
   b) Run the script: `./master_data_flow.sh`

   The output example for the zip file `master_data_2017-12-19.zip`:

Figure 93. The example output

   Where `master_data_2017-12-19.log` is the log directory, `master_data_2017-12-19.report` contains the quality report, and `master_data_2017-12-19.success` indicates that the flow was completed without errors.

2. From the notebook node, start the reading data flow.
   a) Open the `/home/<utility>/automation` directory.
   b) Run the script: `./cm_automation/reading_data_flow.sh`.

   Note: Export the `ANALYSIS_LOAD_UNTIL_TIME` and `ANALYSIS_VOLTAGE_UNTIL_TIME` environment variables with a suitable time for the reading data before running the script if the latest time in the reading data is not yesterday.

   The output example for the zip file `reading_data_2017-12-19.zip`:

Figure 94. The reading data output

Where `reading_data_2017-12-19.log` is the log directory, `reading_data_2017-12-19.report` contains the quality report, and `reading_data_2017-12-19.success` indicates that the flow was completed without errors.

3. From the notebook node, start the analysis flow.
   a) Open the `/home/<utility>/automation` directory.
   b) Run the script:

   ```
   ./cm_automation/analysis_flow.sh
   ```

   Note: Export the `ANALYSIS_LOAD_UNTIL_TIME` and `ANALYSIS_VOLTAGE_UNTIL_TIME` environment variables with a suitable time if necessary.

   The output example for the zip file `analysis_flow_2018-01-08.zip`:

Figure 95. The analysis flow output

   Where `analysis_flow_2018-01-08.log` is the log directory, `analysis_flow_2018-01-08.report` contains the quality report, and `analysis_flow_2018-01-08.success` indicates that the flow was completed without errors.

   c) To enable or disable voltage or load analysis open analysis_flow.sh in a text editor.
      1) Open the file `analysis_flow.sh` in a text editor.
      2) Edit the **tasks** part of this file.

   - To enable the load or voltage analysis remove the comment symbol #.
   - To disable the load or voltage analysis add the comment symbol #.

   Examples:

   To enable voltage analysis:

   ```
   "voltage_analysis" "run_voltage_analysis"
   #"load_analysis" "run_load_analysis"
   ```

   To enable load analysis:

   ```
   #"voltage_analysis" "run_voltage_analysis"
   "load_analysis" "run_load_analysis"
   ```

4. Schedule the flows with crontab script.

   The three flows can be run separately, or scheduled with crontab for a specified time.
   a) Log in as a tenant user.
   b) Run the command:

   ```
   crontab -e
   ```

   c) Put the contents into a text editor.

   Note: For the format of the crontab file, please refer to the Linux crontab guide.

An example crontab file:

```
#specify time zone to be used. Right now it is UTC timezone
CRON_TZ=UTC
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
# specify the necessary environment variables
#PYTHON_LIB=
#SPARK_HOME=

#uncomment the following env variables to adjust
the time used in the flow if needed
#ANALYSIS_LOAD_UNTIL_TIME=2016-09-01
#ANALYSIS_VOLTAGE_UNTIL_TIME=2016-08-31
#LOAD_UNTIL_TIME=2016-09-01
#VOLTAGE_UNTIL_TIME=2016-08-31

#master_data_flow & reading_data_flow
#master data flow scheduled at 14:00 every day, UTC timezone
00 14 * * * $HOME/automation/master_data_flow.sh &
#reading data flow scheduled at 15:00 every day, UTC timezone
00 15 * * * $HOME/automation/reading_data_flow.sh &

#analysis_flow scheduled at 18:00 every Saturday, UTC timezone
00 18 * * sat  $HOME/automation/analysis_flow.sh &
```

Automating the data flow - Quality Reports

Several quality reports are generated after the data flow automation has been executed:

These reports are useful tools that verifies the quality of the data and analysis result.

- Master data quality report
- Reading data quality report
- Analysis result report

The email address that are that are configured in the item `cm_mail_to` receive the reports as an attachment.

The following sections introduce each report and describes how to interpret them.

Master data quality report

The Master data quality report shows the different types of master data in the form of radar charts and tables. The values in the tables are indexed to measure the data quality: The larger of the value, the better the quality of the data.

Feeder

A feeder has three quality attributes defined:

- distinct_feeder: The percentage of the feeders with unique names. Each feeder should have a unique name.
- substation: The percentage of feeder that have defined data for substations.
- geometry: The percentage of feeders that have the geometry defined.

The example shows a data quality chart and table for a feeder:

	distinct_feeder	substation	geometry
0	100.0	100.0	0.0

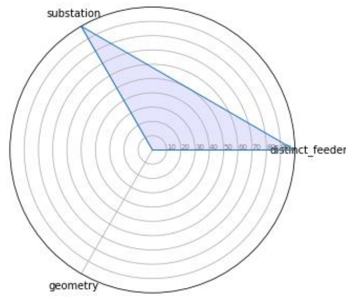


Figure 96. Feeder quality report

Lateral

A lateral has seven quality attributes defined:

- distinct_lateral: The percentage of the laterals with unique names. Each lateral should have a unique name.

  Note: If two feeders have the same name, then the laterals that belong to those two feeders can also have duplicated names. The value of distinct_lateral will not be 100.
- substation: The percentage of laterals that have defined data for substations.
- feeder: The percentage of laterals that have defined data for feeders.
- phase: The percentage of laterals that have defined data for phase.
- nPhases: The percentage of laterals with single phase.
- geometry: The percentage of laterals that have geometry defined.
- ou: The percentage of laterals that have ou defined.

The example shows a data quality chart and table for a lateral:

	distinct_lateral	substation	feeder	phase	nPhases	geometry	ou
0	100.0	100.0	100.0	100.0	100.0	0.0	100.0

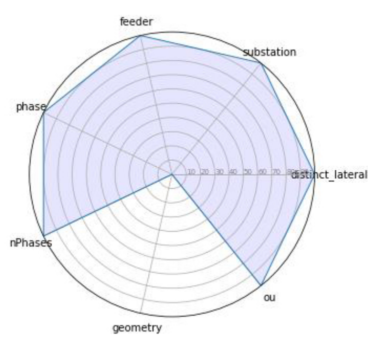


Figure 97. The lateral data quality

Transformer

A transformer has nine quality attributes defined:

- Distinct_transformer: The percentage of the transformers with unique names.

  Note: If two laterals have the same name, then the transformers that belong to those two laterals can also have duplicated names. The value of distinct_transformer will not be 100. This also applies for duplicated feeder names.
- Substation: The percentage of transformers that have defined data for a substation.

- Feeder: The percentage of transformers that have defined data for a feeder.
- Lateral: The percentage of transformers that have defined data for a lateral.
- phase: The percentage of transformers that have defined phase data.
- nPhase: The percentage of transformers with single phase.
- geometry: The percentage of transformers that have defined geometry data.
- ou: The percentage of transformers that have defined ou data.
- kva: The percentage of transformers that have defined kVA data.
- has_meters: The percentage of transformers that have meters defined.

The example shows a data quality chart and table for a transformer:

	distinct_transformer	substation	feeder	lateral	phase	nPhases	geometry	ou	kva	has_meters
0	99.89833	100.0	100.0	95.599129	100.0	100.0	100.0	95.599129	100.0	78.329702

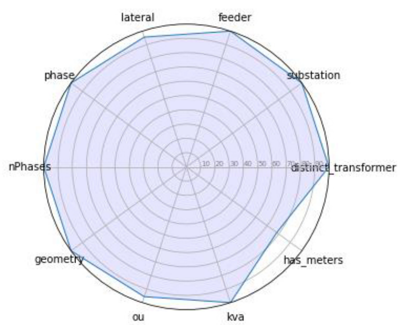


Figure 98. The transformer data quality

Meter

A meter has nine quality data attributes defined:

- distinct_meter: The percentage of the meters with unique names.

  Note: If two transformers have the same name, then the meters that belong to those two transformers can also have duplicated names. The value of distinct_meter will not be 100. This also applies for duplicated lateral and feeder names.
- substation: The percentage of meters that have defined data for a substation.
- feeder: The percentage of meters that have defined data for a feeder.
- lateral: The percentage of meters that have defined data for a lateral.
- transformer: The percentage of meters that have defined data for a transformer.
- phase: The percentage of meters that have defined phase data.
- nPhase: The percentage of meters with single phase.
- geometry: The percentage of meters that have defined geometry data.
- meter_transformer_distance: The percentage of meters with a suitable distance to the corresponding transformers without being excluded by the spatial filtering.

The example shows a data quality chart and table for a meter:

	distinct_meter	substation	feeder	lateral	transformer	phase	nPhases	geometry	meter_transformer_distance
0	99.24569	100.0	100.0	94.12069	100.0	100.0	100.0	100.0	99.353448

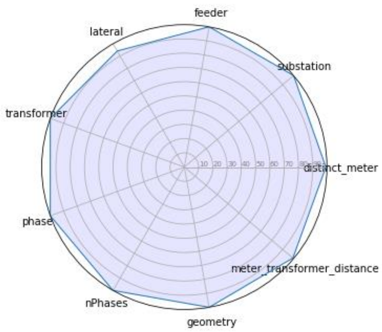


Figure 99. The meter data quality

Reading data quality report

Reading data quality report shows the different types of reading data in the form of radar charts and tables. The values in the tables are indexed to measure the data quality: The larger of the value, the better the quality of the data.

Feeder load

Feeder load has six quality data attributes defined:

- feeder_coverage: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- timestamp_coverage: The percentage of the timestamp coverage.
- timestamp_duplication: The percentage of timestamps that have duplicated values.
- kwh1: The percentage of valid kWh values on phase 1.
- kwh2: The percentage of valid kWh values on phase 2.
- kwh3: The percentage of valid kWh values on phase 3

The example shows a data quality chart and table for feeder load:

	feeder_coverage	timestamp_coverage	timestamp_duplication	kwh1	kwh2	kwh3
0	45.833333	97.777778	21.221064	99.98295	99.983562	99.424462

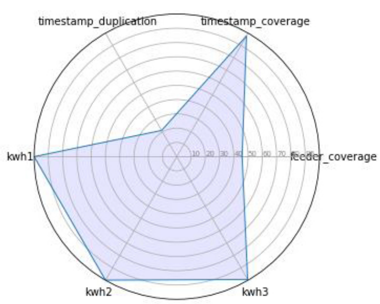


Figure 100. The feeder load data quality

Meter load

The average aggregated meter load gap against the feeder load is used to measure the meter load data quality. An example is as below:

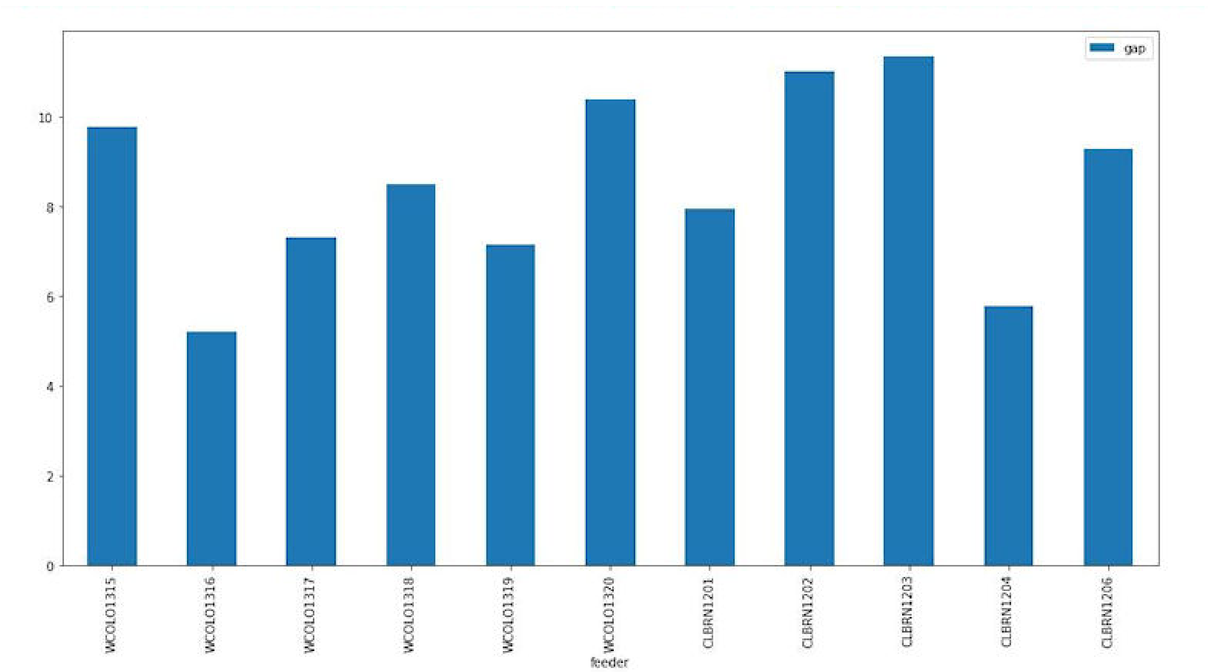


Figure 101. The meter load gap

Meter load has four data attributes defined:

- feeder_coverage: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- meter_coverage: The percentage of meters covered by the reading data flow. Some meters may not be recorded in the reading data for the specified time range.
- disintct_timestamp: The percentage of the timestamps with unique values.
- duplicate_timestamp: The percentage of timestamps with non-duplicated values.
- kwh: The percentage of valid kWh values.

The example shows a data quality chart and table for meter load:

	feeder_coverage	meter_coverage	distinct_timestamp	duplicate_timestamp	kwh
0	95.833333	98.434783	96.864307	99.547199	95.426399

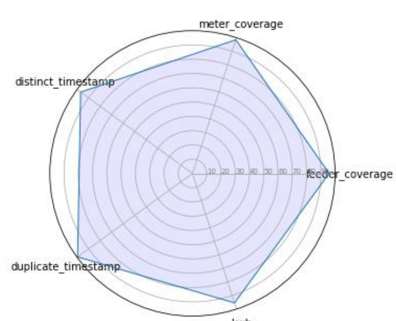


Figure 102. The meter load data quality

Feeder voltage

Feeder voltage has six data attributes defined:

- feeder_coverage: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- timestamp_coverage: The percentage of the timestamp coverage.

- timestamp_duplication: The percentage of timestamps with non-duplicated values.
- volt1: The percentage of valid volt values on phase 1.
- volt2: The percentage of valid volt values on phase 2.
- volt3: The percentage of valid volt values on phase 3.

The example shows a data quality chart and table for feeder voltage:

	feeder_coverage	timestamp_coverage	timestamp_duplication	volt1	volt2	volt3
0	95.833333	97.777778	48.676537	98.351038	80.959733	80.687085

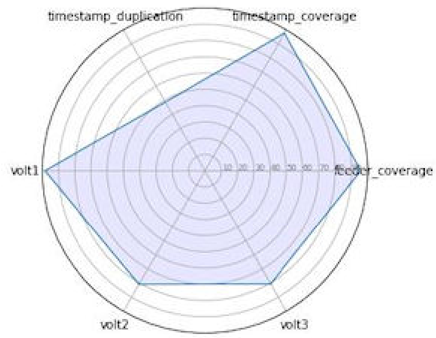


Figure 103. The feeder voltage data quality

Meter voltage

Meter voltage has seven data attributes defined:

- feeder_coverage: The percentage of feeders covered by the reading data flow. Some feeders may not be recorded in the reading data for the specified time range.
- meter_coverage: The percentage of meters covered by the reading data flow. Some meters may not be recorded in the reading data for the specified time range.
- disintct_timestamp: The percentage of the timestamps with unique values.
- duplicate_timestamp: The percentage of timestamps with non-duplicated values.
- volt1: The percentage of valid volt values on phase 1.
- volt2: The percentage of valid volt values on phase 2.
- volt3: The percentage of valid volt values on phase 3.

The example shows a data quality chart and table for meter voltage:

	feeder_coverage	meter_coverage	distinct_timestamp	duplicate_timestamp	volt1	volt2	volt3
0	95.833333	91.956522	92.624585	99.566525	95.788308	12.475503	12.474821

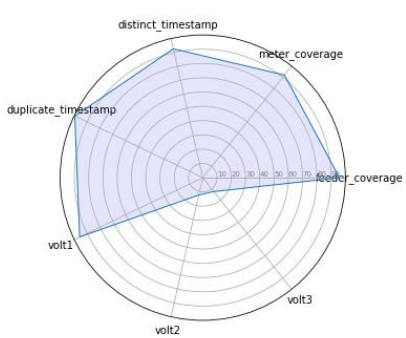


Figure 104. The meter voltage data quality report

Analysis result report

Analysis result report shows the quality of the different types of analysis reports in the form of radar charts and tables. The values in the tables are actual values of the measured data.

Meter coverage analysis for load

The meter coverage analysis for load has six data attributes defined:

- Total: The total number of meters.
- Analyzed: the total number of meters in the analysis.
- Not_analyzed_poly_phase: The number of meters not analyzed that have multiple phases.
- Not_analyzed_missing_load_data: The number of meters not analyzed due to missing load data.
- Not_analyzed_spatial_filtered: The number of meters not analyzed due to being filtered by spatial condition.
- Not_analyzed_other: The number of meters not analyzed due to other factors.

The example shows a chart and table for anaysis result report for load:

	total	analyzed	not_analyzed_poly_phase	not_analyzed_missing_load_data	not_analyzed_spatial_filtered	not_analy:
0	26021	24229	1711	81	0	0

<matplotlib.legend.Legend at 0x4926910>

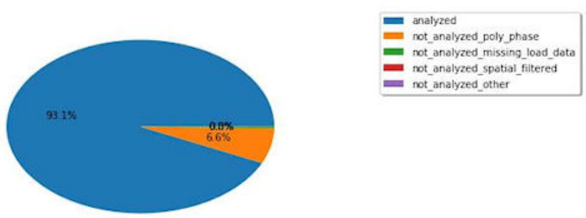


Figure 105. The meter coverage analysis report for load

Accuracy analysis for load

Load accuracy analysis is a statistics chart based on the feeder. An example is as below:

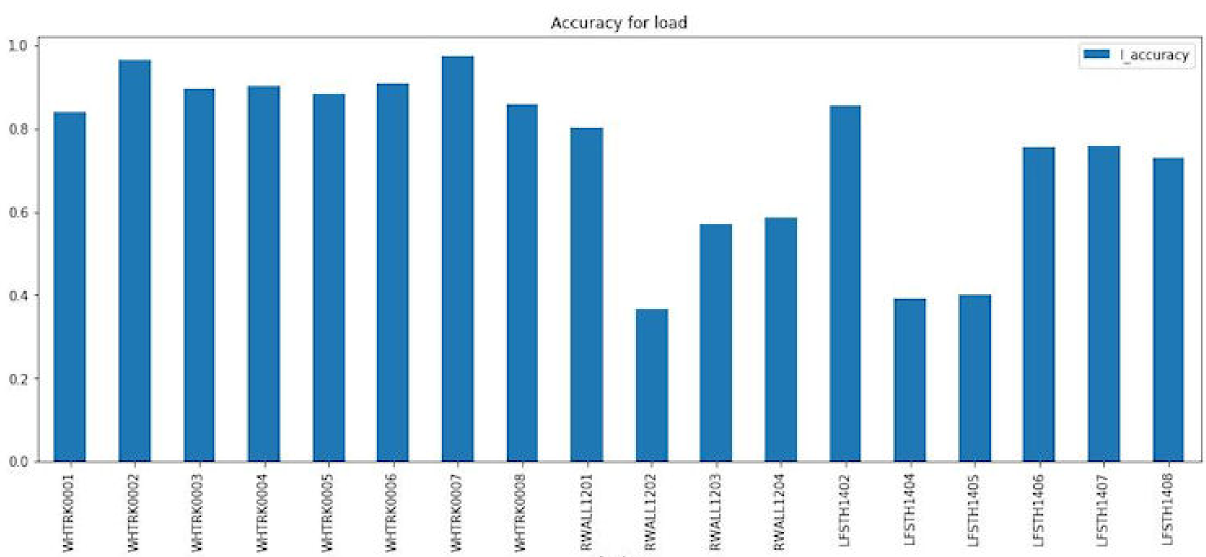


Figure 106. The accuracy analysis for load chart

Meter coverage analysis for voltage

The meter coverage analysis for voltage has six data attributes defined:

- Total: The total number of meters.
- Analyzed: the total number of meters in the analysis.
- Not_analyzed_poly_phase: The number of meters not analyzed that have multiple phases.
- Not_analyzed_missing_voltage_data: The number of meters not analyzed due to missing load data.
- Not_analyzed_spatial_filtered: The number of meters not analyzed due to being filtered by spatial condition.
- Not_analyzed_other: The number of meters not analyzed due to other factors.

The example shows a chart and table for anaysis result report for voltage:

	total	analyzed	not_analyzed_poly_phase	not_analyzed_missing_voltage_data	not_analyzed_spatial_filtered	not_a
0	23288	20854	2150	176	108	0

: <matplotlib.legend.Legend at 0x7fd15cd87b10>

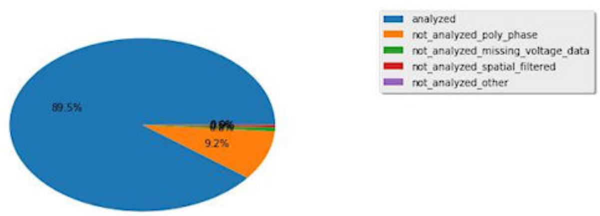


Figure 107. The meter coverage analysis report for voltage

Accuracy analysis for voltage

The voltage accuracy analysis chart shows the statistics based on the feeder. An example is as below:

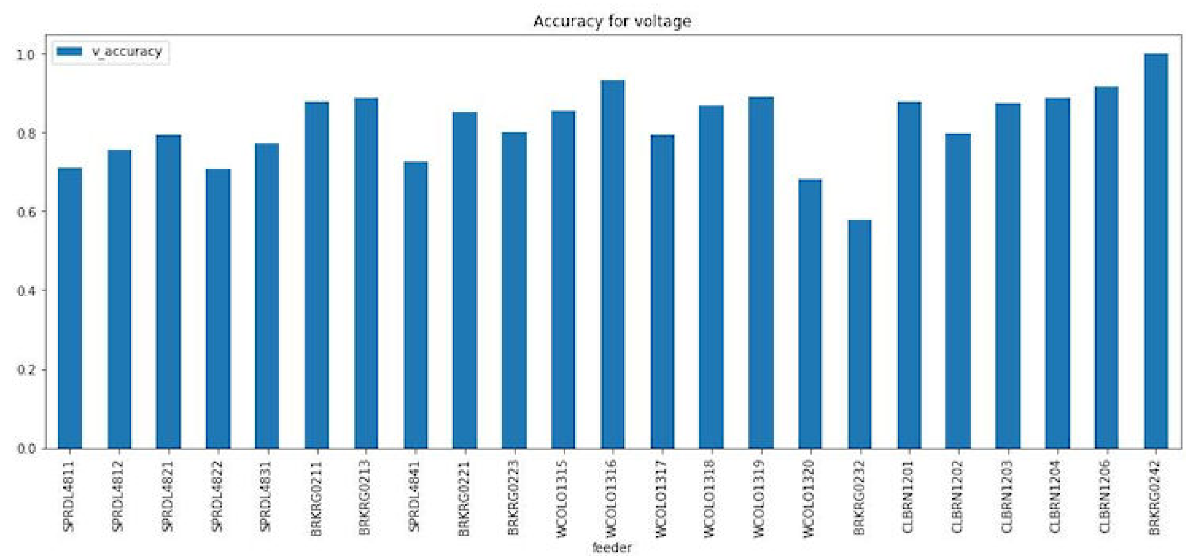


Figure 108. The accuracy analysis for voltage chart

Accuracy distribution for voltage

The accuracy distribution for voltage analysis is a range based chart, from which you can view the number of feeders in each accuracy range (0-100). An example is as below:

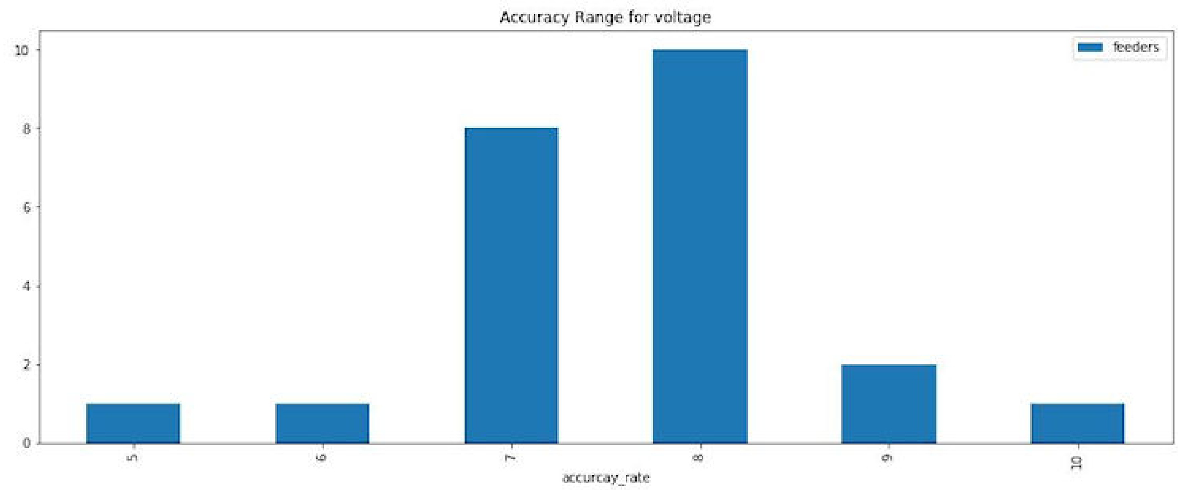


Figure 109. The accuracy distribution for voltage chart

Using the connectivity model application

To use the connectivity model application in IBM Maximo Asset Performance Management for Energy & Utilities SaaS, you must have completed the loading of your data to the application.

When you first open the Connectivity Model application you are presented with a map of the area of the utility with breadcrumbs for the utility name, substations and feeders.

Logging onto the Connectivity Model application

Log on to access the IBM Maximo Asset Performance Management for Energy & Utilities SaaS user interface for the Connectivity Model application.

Before you begin

Contact your local administrator to obtain your user ID and password. Your administrator is responsible for ensuring that you have the security access level that is appropriate to your role in your organization. Your administrator will also supply you with the web address URL for accessing the solution portal.

About this task

Use the following procedure to start a new browser session and access Maximo APM for E&U SaaS.

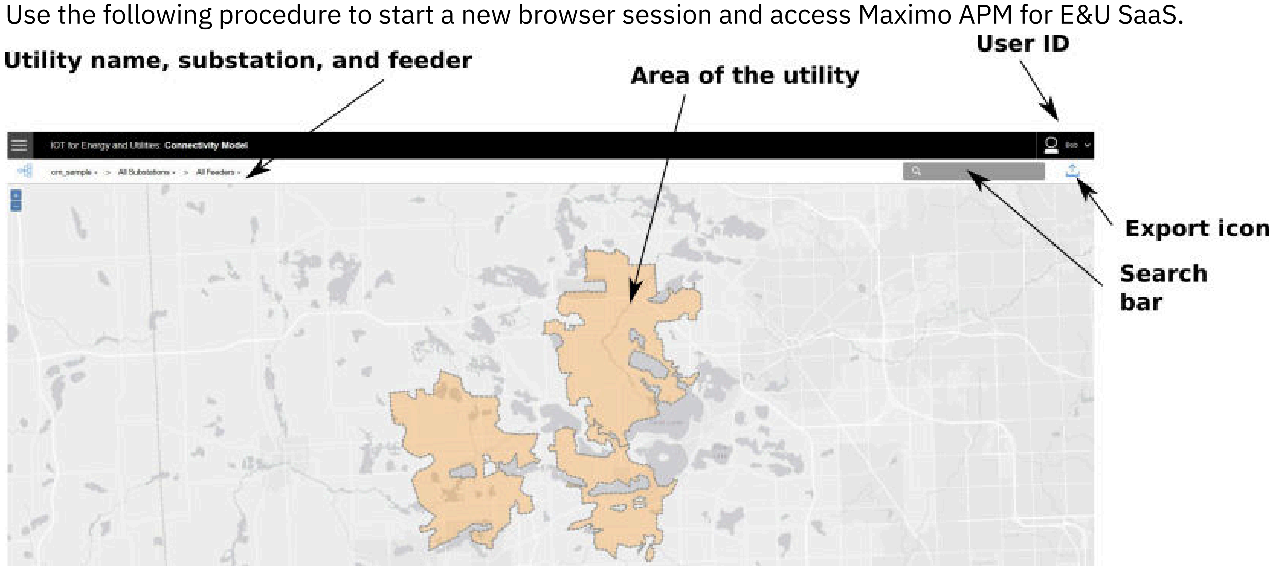


Figure 110. The logon screen of the Connectivity Model application

Procedure

1. Enter the URL into the address field of the browser.

   Note: The fully qualified domain name is required in the URL, for example, `https://web_hostname/ibm/#page_cmodel` where web_hostname is the host name of the web server. If you use the IP address instead of the registered fully qualified domain name, some windows do not open correctly. Also, if you do not use the https protocol, the link is redirected to use the https protocol.
2. On the login page, enter your user ID and password.
3. Click **Log In**.
4. Click the down arrow, and click **Energy** > **Connectivity Model**.

Results

Only the pages, features, and data that you have permission to access are displayed. Contact your administrator if you require more access.

Viewing the legend of the connectivity model application

The legend shows the icons used for the assets in the connectivity model and the colors indicating their status.

About this task

Via the legend you can filter the assets that show on the map for the connectivity model.

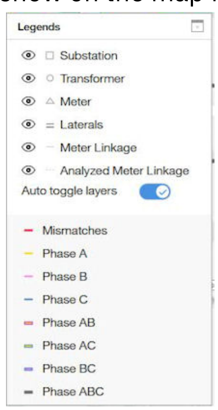


Figure 111. The legend for the Connectivity Model map

The assets that show in the legend are:

- Substation
- Transformer
- Meter
- Laterals
- Meter Linkage
- Analyzed Meter Linkage

The eye icon hides or displays the asset type on the map.

Different asset types are represented by different shapes on the map.

The **Auto toggle layers** switch enables the feature to hide or show assets when zoomed in.

The red color indicates the assets that have a mismatch status. The other colors represent the different phases of the assets.

Procedure

1. Select **Connectivity Model** from the menu bar.
2. Click the first drop-down button in the search bar and select the utility you need.
3. Click the second drop-down button to choose the substation.

   The Legends window opens.

Viewing the basic information of a substation and its feeder

You can view a substation and its feeders.

About this task

After you have selected utility, you can see all the substations in the selected utility. The substation information contains two parts:

- The name of the substation.
- The total number of asset phase and connection errors if errors are present.

The drop-down menu of a feeder is almost the same with the one of substation. You can open it by clicking the third drop-down button.

Procedure

1. In IBM Maximo Asset Performance Management for Energy & Utilities SaaS, click the first drop-down button. Click the utility you want to view.
2. Click the second drop-down button to select the substation you want to view.
3. To search for a certain sub-station, in the search field type a letter contained in the name of the sub-station.
4. Click on the name of the sub-station you want to view.
5. Click the third drop-down button to select the feeder you want to view.
6. Zoom in on the map to see the clusters of transformers connected to the sub station, and again to see the clusters of meters connected to the transformers.

Viewing the detailed information for transformers and the meters

In IBM Maximo Asset Performance Management for Energy & Utilities SaaS you can view more details of a transformer and meters in the hover card and preview panel.

About this task

When you click a cluster of transformers (circle that contains a number) a card shows all the transformers in the cluster and their names.

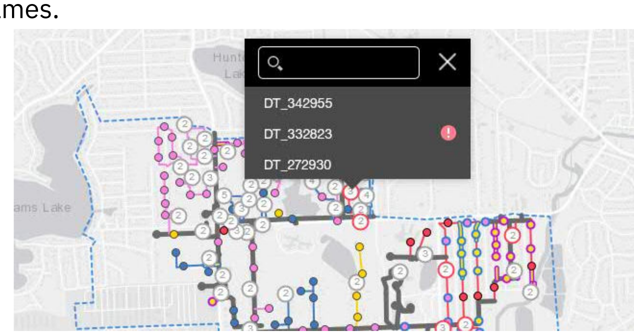


Figure 112. Cluster of transformers

When you click a cluster of meters (diamond that contains a number) a similar card shows all the meters in the cluster and their names.

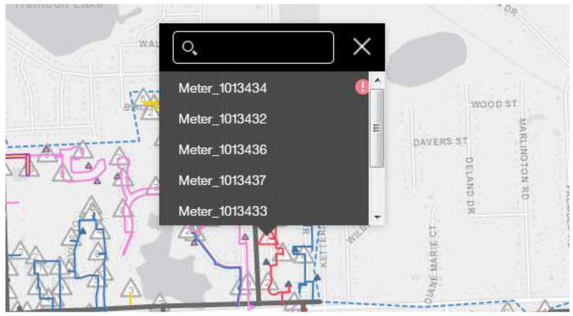


Figure 113. Cluster of meters

When you click a lateral (a thick line) a hover card shows the basic details of the lateral: the name of the lateral, the feeder details, the phase and the connectivity state, the number of transformers on the lateral and the number of meters.

When you click a transformer (small circle) a hover card shows the basic details of the transformer: the name of the transformer, the feeder of the transformer, the lateral, the number of meters, the phase and connectivity state.

When you click a meter (small diamond) a hover card shows the basic details of the meter: the name of the meter, the feeder of the meter, the lateral, the phase and connectivity state.

When you click on the asset in the hover card, the summary panel shows that contains more details the hierarchy tree of the assets and the detailed information about the selected asset. Click on the details tab to see the basic information and property details.

Procedure

1. In Maximo APM for E&U SaaS, click the first drop-down menu button and click the utility you want to view.
2. Click the second drop-down menu button and click the name of the sub-station you want to view.
3. Zoom in on the map to see the clusters of transformers connected to the sub station, and again to see the clusters of meters connected to the transformers.
4. Hover over a transformer on the map to see the general details of a transformer.
5. Click on the transformer to see the preview panel for the transformer.
6. Click on the meter to see the preview panel for the meter.
7. Click on the asset in the hierarchy tree to see the details of different assets.

Showing the confidence level of the connectivity results

A confidence score shows for meters and transformers for the level of confidence of the connectivity result.

About this task

The confidence score given for a meter is the level of confidence for the result as a percentage value. The confidence score given for transformer is the weighted average of the meters that are supplied by the transformer.

Procedure

1. Select **Connectivity Model** from the menu bar.
2. Click the first drop-down button in the search bar and select the utility you need.
3. Click the second drop-down button to choose the substation.
4. Click the third drop-down button to choose the feeder.
5. From the map, click a transformer, shown as a circle, that you want to view.

6. The **Connectivity Summary** screen shows:

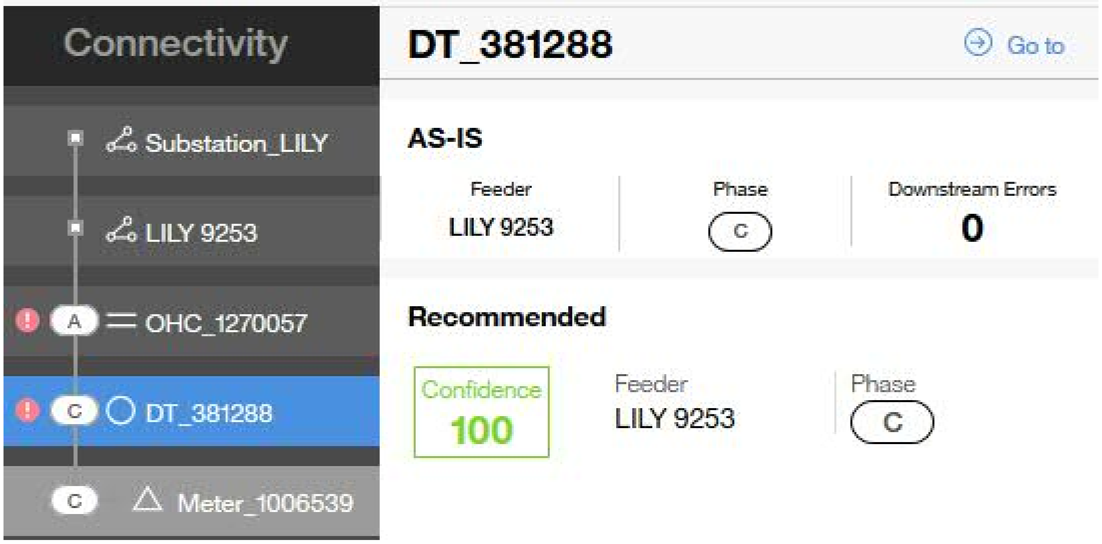


Figure 114. The connectivity results with the level of confidence for a transformer

7. From the **Connectivity Summary** screen click a meter, shown as a triangle, downstream from the transformer.

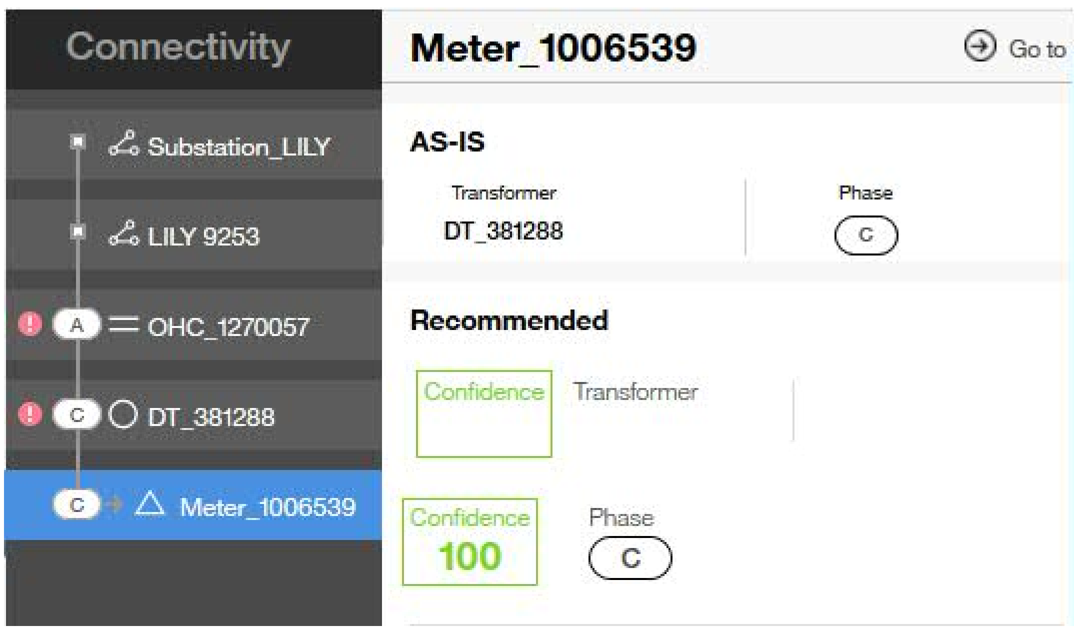


Figure 115. The connectivity results with the level of confidence for a meter

Exporting the asset information of a utility

You can export key performance indicators or asset information from the selected utility using the export facility.

About this task

In the **Export Asset** tab you can select to export the details on transformers or meters to your own system. In the **Export KPI** tab you can export the key performance indicators by utility or by sub-station. You can also select the date of the KPI.

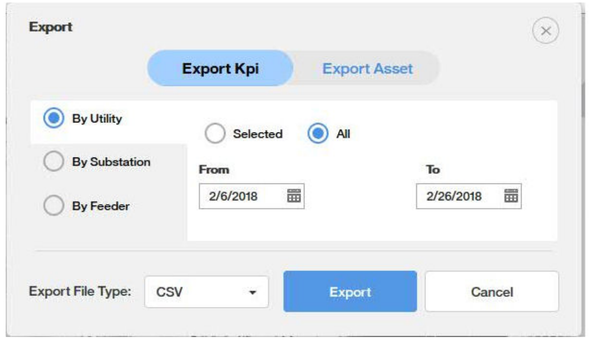


Figure 116. The export dialog box

Procedure

1. Click the Export icon in the search bar.
2. Select either **Export KPI** or the **Export Asset** tab.
3. Select the conditions.
   - **By Utility**
   - **By Substation**
   - **By Feeder**
   - **From** date **To** date
   - **Export File Type**
4. Click **Export**. You can either view or save the exported data.

Chapter 8. Optimizing wind farm operations

IBM Maximo Asset Performance Management for Energy & Utilities SaaS provides situational awareness for wind farm assets to help optimize wind farm performance and assess turbine health and risk.

Overview of the Asset 360 for Wind application

The IBM Maximo Asset Performance Management for Energy & Utilities SaaS application Asset 360 for Wind provides an advisor for wind farm operations and maintenance.

The Wind 360 application integrates the operational systems of the wind farm to deliver predictive, prescriptive, and cognitive solutions that improves the operational and maintenance efficiency of the wind farm.

The Asset 360 for Wind application dashboard

The Asset 360 for Wind application for IBM Maximo Asset Performance Management for Energy & Utilities SaaS has dashboards that have differing access rights dependent on the user profile.

Deputy manager - KPI Dashboard

The key performance indicator dashboard consists of reports that keep the user aware of the current performance of the wind farm without requesting further information. After logging in, the user with KPI Dashboard privileges can view the KPI dashboard and see an overall view of the performance of the wind farm.

Farm KPI report

The KPI report dashboard shows the overall operation index of the wind farm. The index is developed from four dimensions: power production, turbine availability, wind power conversion rate and turbine health. The deputy manager can also see the overall ranking among similar wind farms. The color of the scores represents different ranges: red: <60, yellow: 60-80, green: >=80.

Power Generation report

The Power Generation report shows the power production in mega watts, loss due to down time and the total lost time compared over a three week period which is the previous complete two weeks and current week.

MTBF and MTBR report

MTBF - Mean Time Between Failure

MTTR - Mean Time To Restore

The MTBF and MTBR report shows the comparison between MTBF and MTTR over the previous eight months. The time period can be altered.

Maintenance Cost report

The Maintenance Cost report compares the cost of maintenance of the turbines over the same time period of this year and last year on a monthly basis. You can use the calendar to choose the time period you want to view.

Utilization Hours report

The Utilization Hours report shows the power generated, loss time and utilization hours during the period of time selected from the calendar.

Utilization hour = power generation / turbine full capacity. The Utilization Hours report is the most important KPI for a wind farm.

Weather Forecast report

The Weather Forecast report, shows the weather information such as temperature, wind speed, wind direction, and humidity. You are also provided with maintenance suggestions according to the weather condition. The wind speed shows also for the next 12 hours.

All the weather data is from The Weather Company API.

Operational Engineer - Monitoring dashboard

The operational engineer is responsible for individual turbine and overall farm monitoring. The engineer needs to know the current state of each turbine within the wind farm.

The Map and list views

The Map view shows a visualization of data items associated with their relevant positions on the map. Using the information displayed on the map together with the list view, you can identify location patterns. A score for each turbine indicates the working effectiveness of each turbine.

You can view the basic information for each turbine by hovering over the icon representing the turbine. The hover card shows the turbine serial number, working status, index score, and wind speed. The icon color is consistent with the legend color.

Wind Speed and Direction Trend report

The Wind Speed and Direction Trend report shows the wind condition for the farm. You can hover on it to see the condition for a specific day. The deep blue part of the report represents the past, and the light blue part represents the forecast for the future days.

Farm report

The Farm report shows the three reports:

- Power generation report
- Utilization Hours report
- Turbine Availability report

Turbine report

The Turbine report shows three reports for an individual turbine:

- Turbine KPI report shows the overall operation index for a single turbine. The index is developed from four dimensions: power production, turbine availability, wind power convention rate and turbine health. The color of the scores represents different ranges: red: <60; yellow: 60-80, green: >=80.
- Power curve report shows the actual power curve of the wind turbine compared to the conceptual power curve. If the curves differ greatly, then the turbine is not operating in its optimum state.
- Utilization Hours report shows the utilization hours for a single wind turbine.

Turbine detail report

The Turbine detail report shows three reports:

- Turbine basic information shows the real-time data including active production, wind speed, communication state. The Turbine detail report also shows the turbine type, identification number, installation date, region of installation, wind farm information, and working wind speed.
- Health Degradation report shows the information for health / failure risk and asset details for the major components of the turbine, blade, generator, and transformer.
- Real-Time Monitoring shows 11 real time readings for: Wind Speed, Angle-Blade1, Angle-Blade2, Angle-Blade3, Power At, Turbine Speed, Vibration-X, Vibration-Y, Wind Direction, Yaw Speed, Yaw Wind Direction.

Deputy Manager - Maintenance Dashboard

The Maintenance Dashboard for the Wind360 application gives the user an automatic and optimized maintenance plan that takes into consideration differing constraints and goals. The user can review the plan and do user interface customization as necessary.

Subscribing to the IBM Insights for Weather service

IBM Maximo Asset Performance Management for Energy & Utilities SaaS uses widgets from The Weather Company. You need to subscribe to the IBM Insights for Weather to receive data for use with these widgets.

About this task

You need to be able to subscribe to Weather Company Data for IBM Bluemix and have the credentials for either a user for Maximo APM for E&U SaaS to be able to use the widgets for the Asset 360 for Wind application.

Procedure

1. Open the Blue Mix portal and subscribe to Weather Company Data for IBM Bluemix under Data and Analysis.

   You receive the credential information as example:

   ```
   { "credentials": { "username":
   "71b23b9c-65de-4c19-9e30-88a344a0bde8", "password": "MtTRzSLW6B", "host":
   "twcservice.mybluemix.net", "port": 443, "url":
   "https://71b23b9c-65de-4c19-9e30-88a344a0bde8:MtTRzSLW6B@twcservice.mybluemix.net"
   } }
   ```

2. Open the file for editing on the Liberty server: `/opt/IBM/WebSphere/Liberty/usr/servers/framework_server/lib/weather.properties`
3. Add the credentials to the file.

   For example:

   ```
   user=71b23b9c-65de-4c19-9e30-88a344a0bde8
   password=MtTRzSLW6B
   ```

4. Save and close the file.

Configuring for maintenance planning optimization analysis

You can prioritize the analysis of the maintenance to be carried out based on the lowest production loss, the resources you have at disposal, and for maintenance of the stable production of energy.

Configuring and creating a maintenance plan

Use the maintenance planner in IBM Maximo Asset Performance Management for Energy & Utilities SaaS to analyze the best schedule for maintenance based on the options: lowest production loss, resource work balance, and stable production.

Procedure

1. Go to **Wind 360** > **Maintenance planner** to open the **Maintenance planner** in Maximo APM for E&U SaaS.
2. Select one of the analysis options:`Lowest production lossResource work balanceStable production`
3. The Maintenance planner calculates the best maintenance plan according to your maintenance criteria.

Administering the Asset 360 for Wind application

The Asset 360 for Wind application needs to be set up before use.

The IBM Maximo APM for Energy & Utilities needs to be subscribed to the IBM Insights for Weather service, and to be able to use real time data, you must first regenerate the static data and then generate the real time data.

Performing simulator administration

Generating real time data for Wind 360

You must clean the application of existing data and generate static data from the IBM Maximo Asset Performance Management for Energy & Utilities SaaS Asset 360 for wind application before you can generate real time data.

Before you begin

To be able to regenerate static data you must have **User** credentials.

Procedure

1. In Maximo APM for E&U SaaS, click **Wind 360** > **Data simulator** > **Select All** > **Clear Static Data**

   When the static data is cleared a message is returned:

   ```
   clear:Turbine Power Statistic,Turbine Status Statistic,Turbine Score,Turbine
             Status,Generator Risk,Rotor Risk,Transformer Risk,
             Turbine Power,Turbine Wind Speed,
             Turbine Wind And Power Statistic,
             Windfarm Failure Recovery Statistic,Windfarm Score,
             Windfarm Wind History,Maintenance Basic Data,
             Turbine Maintenance
           cost-done
   ```

2. Click **Regenerate static Data**.

   When the static data is regenerated a message is returned:

   ```
   generate:Turbine Power Statistic,Turbine Status Statistic,Turbine Score,Turbine
             Status,Generator Risk,Rotor Risk,Transformer Risk,Turbine Power,
             Turbine Wind Speed,Turbine Wind And Power Statistic,
             Windfarm Failure Recovery Statistic,Windfarm Score,Windfarm
             Wind History,Maintenance Basic Data,Turbine Maintenance
           cost-done
   ```

3. Click the **Start** button to start the process of generating real time data.

   A message reads: **Now the realtime data generation is running**.

4. Click the **Stop** button to end the process.

   A message reads **Now the realtime data generation is stopped**.

Looking at a holistic view of the operations

IBM Maximo Asset Performance Management for Energy & Utilities SaaS provides an anaysis of the key performance indicators of the wind farm operations.

The analyzes give:

- A comparison between wind farms belonging to the same company.
- A view of the trend of the power generated.
- A view the mean time between failure and mean time to repair.
- A comparison of the maintenance costs compared to the previous year.
- A view of the utilization hours.
- The implications of weather on the operation.

Comparing one wind farm to the others in the same company

As a manager of a wind farm you can compare the key performance indicators of wind farms within the same company.

About this task

You must have management rights of access for IBM Maximo Asset Performance Management for Energy & Utilities SaaS

Procedure

1. Click **Wind 360** > **KPI**.
2. Select the **Country** > **Region** > **Wind farm** from the menu.
3. You can compare the key performance indicators by making a selection from the different wind farms.

View the trend of the power generated over time

In the Power Generation report you can see a comparison of the power production, lost power production and loss time in hours over a period of three weeks, this week and the two previous weeks.

About this task

Note: The trend of power generated over time is not linear. You need to consider for when the wind does not blow. You can do this by removing the out of service due to no wind hours and focus on the actual available hours of the wind turbine generator.

Procedure

1. Click **Wind 360** > **KPI**.
2. View the **Power Generation** report for the visualization.

Viewing repair and restoration statistics

In the MTBF and MTTR report you can see the comparison between the mean time between failure and mean time to repair over a defined period.

About this task

You define the time period that the report shows.

Procedure

1. Click **Wind 360** > **KPI**.
2. View the **MTBF & MTTR** report.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Viewing the maintenance costs

The Maintenance costs report shows the maintenance cost of the turbines in the wind farm during the same time periods for the current year and previous year.

About this task

You can use the calendar to select the period you require.

Procedure

1. Click **Wind 360** > **KPI**.
2. View the **Maintenance Cost** report for the visualization.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Viewing the utilization hours

The Utilization Hours report shows the power generation and power loss in MW and utilization hours during one time period.

About this task

One utilization hour = actual power generation in MW / turbine power generation at full capacity. It can be the most important KPI for a wind farm.

You can select the time period that the report shows.

Procedure

1. Click **Wind 360** > **KPI**.
2. View the **Utilization Hours** report.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Seeing implications of weather on the operation

In this report you get the current weather information: temperature and sun condition, wind speed and direction, and humidity and also wind speed and direction for the next 12 hours.

About this task

You are also provided with maintenance suggestions according to the weather condition.

Procedure

1. Click **Wind 360** > **KPI**.
2. View the **Weather Forcast** report.

Monitoring detailed operations

As an operations engineer you can monitor both the wind farm, and more detailed turbine operations.

The reports show:

- The status of a turbine
- The details condition of a turbine
- The wind speed comparison to power generation
- The wind speed and direction trend over time
- The utilization hours

Monitoring status of a turbine

As an operations engineer, you can monitor the status of each wind turbine in a wind farm.

About this task

You need operations engineer access rights for IBM Maximo Asset Performance Management for Energy & Utilities SaaS.

Procedure

1. Click **Wind 360** > **Monitoring**.
2. In the map view, hover over the wind farm icon and click **View Details**.
3. You can hover over each of the wind turbines to view the status of each one.

   The information that shows is:

   - Serial number of the turbine
   - Working status
   - Index score
   - Wind speed

Viewing the detailed condition of a wind turbine

As the operations engineer you can view the detail condition of each wind turbine.

About this task

You can view the health degradation of the blade, generator and transmission components of the turbine. Each gives information about the asset and the health and failure risk of the asset over time.

The asset information is:

- Asset name
- Asset serial number
- Manufacturer
- Working years
- Installation date
- Wind farm

Procedure

1. Click **Wind 360** > **Monitoring**.
2. In the map view, hover over the wind farm icon and click **View Details**.
3. Click the wind turbine that you need to see the details.
4. Click **Details** in the turbine overview report.
5. Click **`BladeGeneratorTransmission`** to view the **Asset information** and **Health/ failure risk** reports for that asset.

Comparing the wind speed and power generation

The details of wind speed and power generation are shown in the **Real-time** data report.

About this task

Procedure

1. Click **Wind 360** > **Monitoring**.

2. In the map view, hover over the wind farm icon and click **View Details**.
3. View the **Real-time data** report for **Active Production** and **Wind Speed**.

Viewing wind speed and direction trends

About this task

In the **Wind Speed/Direction Trend** report you can view the wind condition of the farm. You can also hover on it to see the specific number of these days. The deep blue part of the report represents the past days and the light blue part represents the forecast in the near future days.

Procedure

1. Click **Wind 360** > **Monitoring**.
2. View the **Wind Speed/Direction Trend** report for the current data, past data and future trend.
3. Hover over each of the days that show to view the wind speed and direction for that day.

Viewing utilization hours

The Utilization Hours report shows the power generation and power loss in MW and utilization hours during one time period.

About this task

One utilization hour = actual power generation in MW / turbine power generation at full capacity. It can be the most important KPI for a wind farm.

You can select the time period that the report shows.

Procedure

1. Click **Wind 360** > **Monitoring**.
2. View the **Utilization Hours** report.
3. Click the calendar icon and click the **from** and **to** calendars select the dates to show.
4. Click **OK**.
5. Click the **Download** icon to download the report.

Chapter 9. API Overview

IBM Maximo Asset Performance Management for Energy & Utilities SaaS is an analytics platform for the energy and utilities industries. The solution delivers functional support for asset health and risk analysis as well as providing a platform that can be used to extend the solution and support other analytics use cases. Use the services and application programming interfaces (APIs) that are provided to extend the solution, to add custom applications, and to integrate the solution with other systems.

The Maximo APM for E&U SaaS analytics platform provides a set of services, a service extension framework, service client APIs, and a user interface extension framework to support the development and integration of new applications. The link to Customizing Maximo APM for E&U SaaS

REST services

The Representational State Transfer (REST) services give you access to the data that is managed by the Maximo APM for E&U SaaS solution. You can use the set of uniform resource indicators (URIs) provided by the REST services to access data about resources such as physical assets, and to trigger analytics.

- Use the Common Information Model service to access data about current and historical logical resources and physical assets.
- Use the connectivity filter service to trigger network connectivity analytics.
- Use the asset health service to access data that is generated by asset health analytics. The output from the analytics includes asset class scores, asset scores, asset driver scores, asset factor scores, asset prescriptions, and container scores.

You can also call the REST services that are provided by the underlying IBM Intelligent Operations Center. For more information about the IBM Intelligent Operations Center REST services, see the related link.

Service extension framework

Use the service extension framework to quickly add and integrate custom REST services to load, query, update, and delete custom data. For more complex or domain-specific data access requirements, you can also integrate custom code for your new services.

Service client APIs

You can use the data access service Java™ API or the Dojo toolkit to make light-weight calls to the REST services from your client applications.

User interface extension framework

You can customize the layout and appearance of the Maximo APM for E&U SaaS user interface to meet your operational requirements. You can also add and customize user interface elements to display new data and content.

Sample scenario

For example, you have a requirement to add a custom application to load custom data into the Maximo APM for E&U SaaS solution. As well as accessing and displaying this custom data, your client application is also required to access data on assets that are managed by the solution.

- Use the service extension framework to create a REST service to load custom data into your custom database tables, and to access the custom data from your application. For more information about quickly creating a REST service to create, query, update, and delete custom data, see “Creating © Copyright IBM Corp. 2015 1 services” on page 47. If your client application has more complex operational requirements for data access and control, you can further customize the new service by integrating custom code. For more information, see “Custom service code”
- Use the REST services that are provided by IBM Insights Foundation for Energy on Cloud to access information about assets that are managed by the solution. For more information, see Chapter 2, “REST services,

- From your client application, you can directly call custom and existing REST services using the data access service Java API, or the Dojo store client interface. For more information, see Chapter 4, “Service client APIs,” on page
- To display the custom content that is managed by your new application, you can extend the user interface by adding new pages, portlets, and widgets. For more information, see Chapter 5, “Extending the user interface,” on page

Documentation for IBM Maximo Asset Performance Management for Energy & Utilities SaaS
Related information

REST services

IBM Maximo Asset Performance Management for Energy & Utilities SaaS provides a set of APIs that are implemented by REST services. The REST services provide a set of URIs that you can use to access the data that is managed by the solution.

You can call the Maximo APM for E&U SaaS REST services from any HTTP client application, and define the expected response in the form of a JSON object. All connections must use HTTPS.

You can use the data access service Java API or the Dojo toolkit to make light-weight calls from client applications to the REST services. For more information, see Chapter 4, “Service client APIs,” on page 55

Some of the services take a simple HTTP GET URI as input. More complicated services take JSON input through HTTP POST for create, HTTP put for update, and HTTP delete for deletion. Error messages and status indicators are returned in the HTTP response.

You can also access all the IBM Intelligent Operations Center REST services. For more information about IBM Intelligent Operations Center REST services, see the related links.

Authentication

Authentication in the Maximo APM for E&U SaaS portal environment is provided by the portal server login. When you log in to the solution with your user ID and password, the security context that is created is passed on to the REST service calls. Therefore, the authentication is handled automatically when the REST service is accessed from a portlet interface. Similarly, when the Java APIs are invoked from a portlet or servlet that is running in the portal, the security context is available and passed on to the service invocation.

You must use HTTP BASIC authentication when you are calling one of the solution's REST services or invoking the REST APIs from client application.

For more information about authentication, see the link to IBM Intelligent Operations Center 1.6 programming guide.

REST service access

Only authorized users can access the REST services. Authorized users are administrators and users who are members of the *EnergyUsers* user group. Administrators are members of the *EnergyAdmins* user group.

HTTP Methods

The Maximo APM for E&U SaaS REST interface provides the following HTTP methods:

Get
: Read a resource.

POST
Create a resource.

PUT
Update a resource.

DELETE
Delete a resource.

Error handling

For errors that are recognized during the processing of a REST request, an appropriate HTTP status code is returned to the calling client.

Supported standards

The IBM Insights Foundation for Energy on Cloud REST APIs support the following Internet standards:

Table 51. Internet standards

Standard	Reference
HTTP 1.1	RFC2616
MIME, Part 1: Format of Internet Message Bodies	RFC 2045
MIME, Part 2: Media Types	RFC 2046
MIME Media Types	The Internet Corporation for Assigned Names and Numbers (IANA)
JSON Schema	Internet Engineering Task Force

Common Information Model service

Use the Common Information Model (CIM) REST service to query metadata information, and to query and create logical resources and their associated physical assets.

The CIM service provides an interface to the CIM-based data model. Use the CIM service to access metadata information about resources, and information about instances of logical resources and physical assets. A logical resource represents a container, equipment, or a logical node in the network, for example, a substation, a transformer, or a junction. A physical asset represents a physical entity in the network that is bound to a logical resource. For example, a transformer logical resource might be bound to different transformer physical asset instances over time as the physical transformers are replaced in the network. Some logical resources by design do not have associated physical assets, for example, a substation or a geographical region. There are also some logical resources that do not have associated physical assets in the current solution, for example, switches.

Data loader REST Services API

resourceType Service

Get resourceType
Method: GET

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/resourceType`

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/resourceType?code=GeographicalRegion`

Required properties: None

Optional properties:

- code

Request

NA

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"id":1,"code":"GeographicalRegion",
"name":{"value":"Geographical Region",
"group":"IFEResourceType",
"key":"GeographicalRegion"},
"description":"A geographical region of a power system network model.
","isAsset":false,
"isResource":false,
"cimID":"GeographicalRegion",
"table":null,"serviceURL":"\/region"}
]
```


Create resourceType

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/resourceType`

For example:

`https://hostname/ibm/ife/api/cim-service/resourceType`

Required properties:

- code,
- name,
- isAsset,
- isResource,
- description,
- cimID,
- table,
- serviceURL

Optional properties: None

Request


```
[
{"code":"TEST","name":{"value":"TEST","group":"IFEResourceType","key":"TEST"},
"description":"A geographical region of a power system network model.",
"isAsset":false,"isResource":false,
"cimID":"TEST","table":null,"serviceURL":"\/test"}
]
```


Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

Update resourceType

URL

`https://hostname/ibm/ife/api/cim-service/resourceType?code=code`

For example:

`https://hostname/ibm/ife/api/cim-service/resourceType?code=TEST`

Required properties:

- code,
- name,
- isAsset,
- isResource,
- description,
- cimID,
- table,
- serviceURL

Optional properties: None

Request

```
[
{"code":"TEST","name":{"value":"Geographical
Region","group":"IFEResourceType","key":"TEST"},
"description":"A geographical region of a power system network model.",
"isAsset":false,"isResource":false,
"cimID":"TEST","table":null,
"serviceURL":"\/test"}
]
```

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Delete resourceType

Method: DELETE

URL

`https://hostname/ibm/ife/api/cim-service/resourceType?code=code`

For example:

`https://hostname/ibm/ife/api/cim-service/resourceType?code=TEST`

Required properties:

- code

Optional properties: None

Request: Not applicable

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**

- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

subType Service

Get subType

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/subType`

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/subType`

`https://`*`hostname`*`/ibm/ife/api/cim-service/subType?code=GeographicalRegion`

`https://`*`hostname`*`/ibm/ife/api/cim-service/subType?resourceType=1`

Required properties: None

Optional properties:

- code
- resourceType

Request: Not applicable

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:

```
[{"code":"GeographicalRegion","resourceType":1,"subType":0,
"name":{"value":"Default","group":"IFESubType","key":"GeographicalRegion"},
"description":"Default"}]
```

Create subType

Method: POST

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/subType`

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/subType`

Required properties:

- resourceType,
- subType,
- name,
- description

Optional properties: None

Request

```
[
{"resourceType":1,"subType":0,
"name":{"value":"Default","group":"IFESubType","key":"GeographicalRegion"},
```

```
"description":"Default"}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Update subType

Method: PUT

URL

https://*hostname*/ibm/ife/api/cim-service/subType?resourceType=*resourceType*&subType=*subType*

For example:

https://*hostname*/ibm/ife/api/cim-service/subType?resourceType=1&subType=1

Required properties:

- resourceType,
- subType,
- name,
- description

Optional properties: None

Request

```
[{"resourceType":1,"subType":1,
"name":{"value":"Default","group":"IFESubType","key":"GeographicalRegion"},
"description":"test"}]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Delete resourceType

Method: DELETE

URL

https://*hostname*/ibm/ife/api/cim-service/subType?resourceType=<resourceType>&subType=*subType*

For example:

https://*hostname*/ibm/ife/api/cim-service/subType?resourceType=1&subType=1

Required properties:

- resourceType,
- subType

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

objectId Service

Get objectId

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/objectid?namespace=`*`namespace`*`&localName=`*`cityName`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/objectid?namespace=http:%2F%2FcityName%23&localName=cityName`

Required properties:

- Namespace,
- localname

Optional properties: None.

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

```
[
{"oid":1000001,"namespace":"http:\/\/cityName#","localName":"WPP-01"},
{"oid":1000002,"namespace":"http:\/\/cityName#","localName":"WPP-02"}...
]
```

Create objectId

Method: POST

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/objectid` for example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/objectid`

Required properties:

- namespace,
- local name.

Optional properties: Not applicable

Request:

```
[
{"namespace":"http:\/\/cityName#","localName":"TEST"}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

location Service

Get location

Method: GET

URL

`https://hostname/ibm/ife/api/cim-service/location`

For example:

`https://hostname/ibm/ife/api/cim-service/location`

Required properties: Not applicable

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"oid":1047168,"mRID":"Loc_Pole_2043733",
"isActive":true,"direction":null,
"mainAddress":null,
"phone1":null,"phone2":null,"secondaryAddress":null,
"geometry":"POINT (-83.4710461 42.5901831)",
"locationDescription":null}...
]
```


Create location

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/location`

For example:

`https://hostname/ibm/ife/api/cim-service/location`

Required properties:

- namespace,
- mRID,
- isActive,
- direction,
- mainaddress,
- phone1,
- phone2,
- secondaryaddress,

- geometry,
- locationdescription

Optional properties: Not applicable

Request:

```
[
{"namespace":"http:\/\/cityName#",
"mRID":"TEST","isActive":true,
"direction":null,"mainAddress":null,
"phone1":null,"phone2":null,
"secondaryAddress":null,"geometry":null,
"locationDescription":null}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

resource Service

Get resource

Method: GET

URL

`https://`*hostname*`/ibm/ife/api/cim-service/resource`

`https://`*hostname*`/ibm/ife/api/cim-service/resource?container=`*container*

`https://`*hostname*`/ibm/ife/api/cim-service/resource?isContainer=`*true/false*

`https://`*hostname*`/ibm/ife/api/cim-service/resource?isAsset=`*true/false*

`https://`*hostname*`/ibm/ife/api/cim-service/resource?name=`*name*

`https://`*hostname*`/ibm/ife/api/cim-service/resource?mRID=`*mRID*

For example:

`https://`*hostname*`/ibm/ife/api/cim-service/resource`

`https://`*hostname*`/ibm/ife/api/cim-service/resource?resourceType=1021954`

`https://`*hostname*`/ibm/ife/api/cim-service/resource?isContainer=true`

`https://`*hostname*`/ibm/ife/api/cim-service/resource?isAsset=false`

`https://`*hostname*`/ibm/ife/api/cim-service/resource?name=pole_406172`

`https://`*hostname*`/ibm/ife/api/cim-service/resource?mRID=pole_406172`

Required properties: None

Optional properties:

- Conatainer,
- isContainer,
- isAsset,
- name,
- mRID

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"oid":1001130,"resourceType":8,
"mRID":"ST_418702","isActive":true,
"name":"ST_418702","description":"Substation Transformer 418702",
"isContainer":false,"location":1001137,
"container":1001125,"geometry":"POINT (-83.4093851 42.5653031)",
"resourceTypeName":{"value":"Substation Transformer",
"group":"IFEResourceType","key":"SubstationTransformer"},
"isAsset":true,"resourceTypeCode":"SubstationTransformer"}
]
```


Create resource

Method: POST

URL

`https://`*hostname*`/ibm/ife/api/cim-service/resource`

For example:

`https://`*hostname*`/ibm/ife/api/cim-service/resource`

Required properties:

- Namespace,
- mRID,
- isActive,
- Direction,
- mainaddress,
- phone1,
- phone2,
- secondaryaddress,
- geometry,
- locationdescription

Optional properties: Not applicable

Request:


```
[
{"namespace":"http:\/\/cityName#",
"resourceType":8,"mRID":"TEST","isActive":true,
"name":"TEST","description":"Substation Transformer 418702",
"isContainer":false,
"location":null,"container":null}
]
```


Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

connectivityNode Service

Get connectivityNode

Method: GET

URL

`https://hostname/ibm/ife/api/cim-service/connectivityNode`

`https://hostname/ibm/ife/api/cim-service/connectivityNode?container=container`

For example:

`https://hostname/ibm/ife/api/cim-service/connectivityNode`

`https://hostname/ibm/ife/api/cim-service/connectivityNode?container=1001123`

Required properties: None

Optional properties:

- container

Request: Not applicable.

```
```

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[{"oid":1067302,"mRID":"CN_ENJ_1000084","isActive":true,"name":"ENJ_1000084",
"description":"ENJ_1000084","container":1001123},
{"oid":1067303,"mRID":"CN_ENJ_1000813","isActive":true,"name":"ENJ_1000813",
"description":"ENJ_1000813","container":1001123},
{"oid":1067304,"mRID":"CN_ENJ_1001379","isActive":true,"name":"ENJ_1001379",
"description":"ENJ_1001379","container":1001123},
{"oid":1067305,"mRID":"CN_ENJ_1001521","isActive":true,"name":"ENJ_1001521",
"description":"ENJ_1001521","container":1001123}......
]
```


Create connectivityNode

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/connectivityNode`

For example:

`https://hostname/ibm/ife/api/cim-service/connectivityNode`

Required properties:

- Namespace,
- mRID,
- isActive,
- name,
- description,
- containerMRID

Optional properties: Not applicable

Request:

```
[
{"namespace":"http:\/\/cityName#",
"mRID":"CN_BJ_LILY_102591",
"isActive":true,"name":"CN_BJ_LILY_102591",
"description":"CN_BJ_LILY_102591",
"containerMRID":"Substation_LILY"}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

terminal Service

Get terminal

Method: GET

URL

`https://`*hostname*`/ibm/ife/api/cim-service/terminal`

`https://`*hostname*`/ibm/ife/api/cim-service/terminal?container=`*container*

`https://`*hostname*`/ibm/ife/api/cim-service/terminal?resource=`*resource*

`https://`*hostname*`/ibm/ife/api/cim-service/terminal?connectivityNode=`*connectivityNode*

For example:

`https://`*hostname*`/ibm/ife/api/cim-service/terminal`

`https://`*hostname*`/ibm/ife/api/cim-service/terminal?container=1001123`

`https://`*hostname*`/ibm/ife/api/cim-service/terminal?resource=1066925`

`https://`*hostname*`/ibm/ife/api/cim-service/terminal?connectivityNode=1076078`

Required properties: None

Optional properties:

- container,
- resource,
- connectivityNode

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:

```
[
{"oid":1081092,"mRID":"Terminal_DPD_1003964_1",
"isActive":true,"name":"Terminal 1",
"description":"Terminal 1","sequenceNumber":1,
"connected":true,"phases":null,
"resource":1066924,"connectivityNode":1076551,
"container":1001123}......
]
```

Create terminal

Method: POST

URL

https://*hostname*/ibm/ife/api/cim-service/terminal

For example:

https://*hostname*/ibm/ife/api/cim-service/terminal

Required properties:

- namespace,
- mRID,
- isActive,
- name,
- description,
- sequenceNumber,
- connected,
- phases,
- resourceMRID,
- connectivityNodeMRID

Optional properties: None

Request:


```
[
{"namespace":"http:\/\/cityName#",
"mRID":"Terminal_OHLS_1148744_0","isActive":true,
"name":"Terminal_OHLS_1148744_0",
"description":"Terminal_OHLS_1148744_0",
"sequenceNumber":0,"connected":true,
"phases":null,"resource":1022498,
"resourceMRID":"OHLS_1148744","connectivityNode":1076551,
"connectivityNodeMRID":"CN_BJ_NIXON_150759","container":1001123}
]
```


Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

measurement Service

get measurement

Method: GET

URL

https://*hostname*/ibm/ife/api/cim-service/measurement

https://*hostname*/ibm/ife/api/cim-service/measurement?resource=*resource*

https://*hostname*/ibm/ife/api/cim-service/measurement?terminal=*terminal*

https://*hostname*/ibm/ife/api/cim-service/measurement?measurementType=*measurementType*

For example:

`https://hostname/ibm/ife/api/cim-service/measurement`

`https://hostname/ibm/ife/api/cim-service/measurement?resource=1000005`

`https://hostname/ibm/ife/api/cim-service/measurement?`
`measurementType=LoadEnergy`

Required properties: None

Optional properties:

- measurementType,
- resource,
- terminal

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"oid":1000269,"mRID":"WTG001_WTURMainCost",
"isActive":true,"name":"WTURMainCost",
"measurementType":"WTURMainCost",
"phaseCode":null,"unitMultiplier":"none",
"unitSymbol":"none","resource":1000005,"terminal":null}......
]
```


Create measurement

Method: POST

Required properties:

-
-

URL

`https://hostname/ibm/ife/api/cim-service/measurement`

For example:

`https://hostname/ibm/ife/api/cim-service/measurement`

Required properties:

- Namespace,
- mRID,
- isActive,
- name,
- phaseCode,
- unitMultiplier,
- unitSymbol,
- resourceMRID

Optional properties:

- Resource,
- terminal,
- measurementType

Request:

```
[
{"namespace":"http:\/\/cityName#",
"mRID":"WTG001_WTURMainCost",
"isActive":true,"name":"WTURMainCost",
"measurementType":"WTURMainCost","phaseCode":null,
"unitMultiplier":"none","unitSymbol":"none",
"resourceMRID":"WTG001","terminal":null}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

supportAssociation Service

Get supportAssociation

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/supportAssociation`

`https://`*`hostname`*`/ibm/ife/api/cim-service/supportAssociation?supported=`*`supported`*

`https://`*`hostname`*`/ibm/ife/api/cim-service/supportAssociation?supporter=`*`supporter`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/supportAssociation`

`https://`*`hostname`*`/ibm/ife/api/cim-service/supportAssociation?supporter=1039659`

`https://`*`hostname`*`/ibm/ife/api/cim-service/supportAssociation?supported=1013659`

Required properties: None

Optional properties:

- supporter,
- supported

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:

```
[
{"oid":1116388,"mRID":"S_Pole_1561466_OHC_903404","isActive":true,"supporter":1039659,
"supported":1013659},
{"oid":1116758,"mRID":"S_Pole_1991129_OHC_903404","isActive":true,"supporter":1039878,
"supported":1013659}......
]
```

Create supportAssociation

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/supportAssociation`

For example:

`https://hostname/ibm/ife/api/cim-service/supportAssociation`

Required properties:

- Namespace,
- mRID,
- isActive,
- supporterMRID,
- supportedMRID

Optional properties:

- resource,
- terminal,
- measurementType

Request:

```
[
{"namespace":"http:\/\/cityName#",
"mRID":"S_Pole_405635_OHC_94778",
"isActive":true,"supporterMRID":"Pole_405635",
"supportedMRID":"OHC_94778"}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

flowRole Service

get flowRole

Method: get

URL

`https://hostname/ibm/ife/api/cim-service/flowRole`

`https://hostname/ibm/ife/api/cim-service/flowRole?role=role`

For example:

`https://hostname/ibm/ife/api/cim-service/flowRole`

`https://hostname/ibm/ife/api/cim-service/flowRole?role=0`

Required properties: None

Optional properties:

- role

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
{"oid":1066951,"isActive":true,"role":0},
{"oid":1067032,"isActive":true,"role":0}...
]
```


Create flowRole

Method: POST

URL

`https://`*hostname*`/ibm/ife/api/cim-service/flowRole`

For example:

`https://`*hostname*`/ibm/ife/api/cim-service/flowRole`

Required properties:

- namespace,
- mRID,
- isActive,
- role

Optional properties: None

Request:


```
[
{"namespace":"http:\/\/cityName#","mRID":"CN_ENJ_685484","isActive":true,"role":0}
]
```


Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

asset Service

Get asset

Method: GET

URL

`https://`*hostname*`/ibm/ife/api/cim-service/asset/`*type*

`https://`*hostname*`/ibm/ife/api/cim-service/asset/`*type*`?oid=`*oid*

`https://`*hostname*`/ibm/ife/api/cim-service/asset/`*type*`?container=`*container*

`https://`*hostname*`/ibm/ife/api/cim-service/asset/`*type*`?isActive=`*isActive*

For example:

`https://`*hostname*`/ibm/ife/api/cim-service/asset/Pole`

`https://`*hostname*`/ibm/ife/api/cim-service/asset/Pole?oid=1036986` `https://hostname/ibm/ife/api/cim-service/asset/Pole?container=1001124`

`https://hostname/ibm/ife/api/cim-service/asset/Pole?isActive=true`

Required properties:

- type

Optional properties:

- oid,
- container,
- isActive

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"oid":1037066,"mRID":"Pole_1510003",
"resourceType":7,"name":"Pole_1510003",
"description":"Pole 1510003","isContainer":false,
"location":1047287,"container":1001124,
"geometry":"POINT (-83.4860180 42.5771090)","serialNumber":"1510003",
"installationDate":"1945-01-01","removalDate":null,
"materialKind":null,"speciesType":"Western Red Cedar",
"reclaimedStatus":false,"preservativeKind":"Creosote",
"isActive":true,"treatmentKind":"B",
"length":35.0,"classification":null,"manufacturer":null}.....
]
```


Create asset

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/asset/type`

For example:

`https://hostname/ibm/ife/api/cim-service/asset/Pole`

Required properties:

- Namespace,
- mRID,
- resourceType,
- description,
- isContainer,
- location,
- container <serial Number>,
- installationDate,
- installationDate,
- removalDate,
- materialKind,
- speciesType,
- reclaimedStatus,

- preservativeKind,
- isActive,
- treatmentKind,
- length,
- classification,
- manufacturer

Note: The properties are based on the resourceType, Pole.

Optional properties: Not applicable

Request:


```
[
{"namespace":"http:\/\/cityName#",
"mRID":"Pole_402815","resourceType":7,
"name":"Pole_1510003","description":"Pole 1510003",
"isContainer":false,"location":1047287,
"container":1001124,"geometry":"POINT (-83.4860180 42.5771090)",
"serialNumber":"1510003","installationDate":"1945-01-01",
"removalDate":null,"materialKind":null,
"speciesType":"Western Red Cedar",
"reclaimedStatus":false,"preservativeKind":"Creosote",
"isActive":true,"treatmentKind":"B",
"length":35.0,"classification":null,"manufacturer":null}
]
```


Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

propertyType Service

Get propertyType

Method:

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType`

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?resourceType=`*`resourceType`*

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?code=`*`code`*

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?id=`*`id`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType`

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?resourceType=7`

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?code=installationDate`

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?id=3`

Required properties: None

Optional properties:

- resourceType,
- code,

- id

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
{"id":3,"resourceType":7,"code":"installationDate",
"name":{"value":"Installation Date","group":"IFEPropertyType",
"key":"Pole_installationDate"},
"description":"Date current installation was completed","cimID":null,
"column":"INSTALLATIONDATE","dataType":"DATE","propertyType":"Date",
"length":4,"scale":0,"isEnum":false,
"isReference":false,"isMinimal":true,"tag":"Age",
"group":{"group":"IFEPropertyGroup","key":"GeneralInformation"},
"unit":null,"allowNull":true}
]
```


Create propertyType

Method: POST

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType`

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType`

Required properties:

- Namespace,
- mRID,
- isActive,
- role

Optional properties: Not applicable

Request:


```
[{"namespace":"http:\/\/cityName#",
"resourceType":8,"code":"test",
"name":{"group":"IFEPropertyType","key":"5ce55b80-709b-4634-813e-9c6ddc4c62a3"},
"description":"","column":"test","dataType":"VARCHAR",
"length":0,"scale":0,"allowNull":true,"defaultValue":"",
"group":{"group":"IFEPropertyGroup","key":"GeneralInformation"}}]
```


Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

Update propertyType

Method: PUT

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?id=`*`id`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?id=82`

Required properties:

- namespace,
- resourceType,
- code,
- name,
- cimID,
- column,
- dataType,
- propertyType,
- length,
- scale,
- isEnum,
- isReference,
- isMinimal,
- tag,
- group,
- unit,
- allowNull

Optional properties: None

Request:


```
[
{"namespace":"http:\/\/cityName#","resourceType":7,"code":"installationDateTest",
"name":{"value":"Installation Date",
"group":"IFEPropertyType","key":"Pole_installationDate"},
"description":"Date current installation was completed","cimID":null,
"column":"INSTALLATIONDATETEST","dataType":"DATE","propertyType":"Date",
"length":4,"scale":0,"isEnum":false,
"isReference":false,"isMinimal":true,
"tag":"Age","group":{"group":"IFEPropertyGroup",
"key":"GeneralInformation"},"unit":null,"allowNull":true}

]
```


Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Delete propertyType

Method: DELETE

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType/`*`id`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType/82`

Required properties:

- id

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

measurementType Service

Get measurementType

Method:

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType`

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?resourceType=`*`resourceType`*

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?resourceTypeId=`*`resourceTypeId`*

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?measurementType=`*`measurementType`*

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?targetTable=`*`targetTable`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType`

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?resourceTypeId=7`

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?measurementType=DGA`

`https://`*`hostname`*`/ibm/ife/api/cim-service/propertyType?targetTable=CIM.DGAVALUE`

Required properties: None.

Optional properties:

- resourceType,
- resourceTypeId,
- measurementType,
- targetTable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"resourceTypeId":9,"measurementType":"CoolingStage",
"description":"CoolingStage","targetTable":"CIM.DIGITALVALUE",
```

```
"mapping":"value:String:VALUE,status:String:STATUS"}
]
```

Create measurementType

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/measurementType`

For example:

`https://hostname/ibm/ife/api/cim-service/measurementType`

Required properties:

- resourceTypeId,
- measurementType,
- description,
- targetTable,
- mapping

Optional properties: Not applicable

Request:

```
[
{"resourceTypeId":9,"measurementType":"CoolingStage",
"description":"CoolingStage","targetTable":"CIM.DIGITALVALUE",
"mapping":"value:String:VALUE,status:String:STATUS"}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Update measurementType

Method: PUT

URL

`https://hostname/ibm/ife/api/cim-service/measurementType?resourceTypeId=resourceTypeId&measurementType=measurementType`

For example:

`https://hostname/ibm/ife/api/cim-service/measurementType?resourceTypeId=6&measurementType=DGA`

Required properties:

- resourceTypeId,
- measurementType,
- description,
- targetTable,
- mapping

Optional properties: Not applicable

Request:

```
[
{"resourceTypeId":9,"measurementType":"tttt",
"description":"CoolingStage","targetTable":"CIM.DIGITALVALUE",
"mapping":"value:String:VALUE,status:String:STATUS"}
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

```

```

Delete measurementType

Method: DELETE

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?resourceTypeId=`*`resourceTypeId`*`&measurementType=`*`measurementType`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/measurementType?resourceTypeId=6&measurementType=DGA`

Required properties:

- id

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

reading Service

Get reading

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/reading/{resourceType:String}/{measurementType:String}`

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/reading/SubstationTransformer/DGA`

Required properties:

- resourceType,
- measurementType

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
{"measurement":1113115,"timestamp":"2008-06-16T23:00:00+08:00",
"CH4":16.265060424804688,"C2H4":81.32530212402344,"C2H2":2.4096388816833496}
]
```


Create reading

Method: POST

URL

`https://`*hostname*`/ibm/ife/api/cim-service/reading/{resourceType:String}/{measurementType:String}`

For example:

`https://`*hostname*`/ibm/ife/api/cim-service/reading/SubstationTransformer/DGA`

Required properties:

- Namespace,
- mRID,
- inspectionDate
- Value1,
- Value2,
- Value3

Note: The properties differ depending on the resourceType and measurementType.

Optional properties: Not applicable

Request:


```
[
{"namespace":"http:\/\/cityName#",
"mRID":"M_ST_38614_DGA","inspectionDate":"2013-01-01 01:10:00.000",
"Value1":"+6.20000E+001","Value2":"+6.00000E+000","Value3":"+3.20000E+001"}
]
```


Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

measurementProperties Service

Get meausrementProperties

Method: GET

URL

`https://`*hostname*`/ibm/ife/api/cim-service/measurementProperties`

`https://`*hostname*`/ibm/ife/api/cim-service/measurementProperties?resourceType=`*resourceType* https://*hostname*/ibm/ife/api/cim-service/measurementProperties?measurementType=*measurementType*

https://*hostname*/ibm/ife/api/cim-service/measurementProperties?resourceType=*resourceType*&measurementType=*measurementType*

For example:

https://*hostname*/ibm/ife/api/cim-service/measurementProperties

https://*hostname*/ibm/ife/api/cim-service/measurementProperties?resourceType=SubstationTransformer

https://*hostname*/ibm/ife/api/cim-service/measurementProperties?measurementType=DGA

https://*hostname*/ibm/ife/api/cim-service/measurementProperties?resourceType=SubstationTransformer&measurementType=DGA

Required properties: None

Optional properties:

- resourceType,
- measurementType

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"measurementType":"DGA","resourceType":"SubstationTransformer","propertyName":"CH4",
"propertyType":"Number","propertyNullable":false},
{"measurementType":"DGA","resourceType":"SubstationTransformer","propertyName":"C2H4",
"propertyType":"Number","propertyNullable":false},
{"measurementType":"DGA","resourceType":"SubstationTransformer","propertyName":"C2H2",
"propertyType":"Number","propertyNullable":false}
]
```


propertyRange Service

Get propertyRange

Method: GET

URL

https://*hostname*/ibm/ife/api/cim-service/measurementProperties?resourceType=*resourceType*

For example:

https://*hostname*/ibm/ife/api/cim-service/measurementProperties?resourceType=9

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**

- Forbidden: **403 Forbidden**

Response:

```
[{ "min": 40000,"dataType": "INTEGER",
"max": 120000,"column": "OPERATINGVOLTAGE","code": "operatingVoltage" },
{"min": 1368,"dataType": "INTEGER",
"max": 2781,"column": "SUMMEREMERGENCYRATING",
"code": "summerEmergencyRating"}]
```

readingTable Service

Get readingTable

Method: GET

URL

`https://`*hostname*`/ibm/ife/api/cim-service/readingTable`

For example:

`https://<hostname>/ibm/ife/api/cim-service/readingTable`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:

```
[{"description":"used to store digital value","tableName":"CIM.DIGITALVALUE"},
{"description":"used to store ana log","tableName":"CIM.ANALOGVALUE"},
{"description":"used to store DGA","tableName":"CIM.DGAVALUE"}]
```

Create readingTable

Method: POST

URL

`https://`*hostname*`/ibm/ife/api/cim-service/readingTable`

For example:

`https://`*hostname*`/ibm/ife/api/cim-service/readingTable`

Required properties:

- tableName,
- description

Optional properties:

Request: Not applicable.

```
[{"description":"used to store digital value","tableName":"CIM.DIGITALVALUE"}]
```

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Update readingTable

Method: PUT

URL

`https://hostname/ibm/ife/api/cim-service/readingTable?tableName=tableName`

For example:

`https://hostname/ibm/ife/api/cim-service/readingTable?tableName=CIM.DIGITALVALUE`

Required properties:

- tableName,
- description

Optional properties: Not applicable

Request:

```
[{"description":"used to store digital value","tableName":"CIM.DIGITALVALUE"}]
```

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Delete readingTable

Method: DELETE

URL

`https://hostname/ibm/ife/api/cim-service/readingTable?tableName=tableName`

For example:

`https://hostname/ibm/ife/api/cim-service/readingTable?tableName=CIM.DIGITALVALUE`

Required properties:

- tableName

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

measurementColumns Service

Get measurementColumns

Method: GET

URL

`https://hostname/ibm/ife/api/cim-service/measurementColumns`

For example:

https://*hostname*/ibm/ife/api/cim-service/measurementColumns

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[{"name": "NAME","tbname": "SYSTABLES","tbcreator": "SYSIBM  ","coltype": "VARCHAR
","nulls": "N","length": 128,"scale": 0,"default": null },
{"name": "CREATOR","tbname": "SYSTABLES","tbcreator": "SYSIBM  ","coltype": "VARCHAR ",
"nulls": "N","length": 128,"scale": 0, "default": null}]
```


resource Child Service

Get resourceChild

Method: GET

URL

https://*hostname*/ibm/ife/api/cim-service/resouceChild?code=*code*

For example:

https://*hostname*/ibm/ife/api/cim-service/resouceChild?code=*Substation*

Required properties:

- code

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[{"code":"Feeder"},
{"code":"Pole"},
{"code":"Switch"},
{"code":"DynamicProtectiveDevice"},
{"code":"Tower"}]
```


resource Count Service

Get resourceCount

Method:

URL

https://*hostname*/ibm/ife/api/cim-service/resouceCount?code=*code*

For example:

https://*hostname*/ibm/ife/api/cim-service/resouceCount?code=Substation

Required properties:

- code

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

```
[{"count":3}]
```

asset Count Service

Get assetCount

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/assetCount?code=`*`code`*

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/assetCount?code=Substation`

Required properties:

- code

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:

```
[{"count":14677}]
```

dataloaderHistory Service

Get dataloaderHistory

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/api/cim-service/dataloaderHistory`

For example:

`https://`*`hostname`*`/ibm/ife/api/cim-service/dataloaderHistory`

Required properties: None

Optional properties: Not applicable

Request: Not applicable

Status:

- Successful **200 OK**

- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[{"name":"20171123114336","status":3,"zipName":"utilityname.zip",
"description":"ExampleZip","uploadTime":"2017-11-23T11:44:43.573+08:00",
"log":"L29wdC9JQk0vZW5lcmd5L2RhdGEvbG9ncy9ydW5fZGF0YUxvZGVyXzIwMTcxMTIzVDExLTQ0LTQz\r
\nLmxvZw==",
"failureReason":null},
{"name":"20171123114737","status":4,"zipName":"utilityname.zip",
"description":"","uploadTime":"2017-11-23T11:47:46.945+08:00",
"log":null,
"failureReason":"config.properties is not existing, please check the uploading compressed
package."}]
```


Create dataloadHistory

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/dataloaderHistory`

For example:

`https://hostname/ibm/ife/api/cim-service/dataloaderHistory`

Required properties:

- Name,
- status,
- zipName,
- description

Optional properties: Not applicable

Request:


```
[{"name":"20171123114336","status":3,"zipName":"utilityname.zip","description":"ExampleZip"
},
{"name":"20171123114737","status":4,"zipName":"utilityname.zip","description":""}]
```


Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

concurrentGuard Service

Get concurrentGuard

Method: GET

URL

`https://hostname/ibm/ife/api/cim-service/concurrentGuard`

For example:

`https://hostname/ibm/ife/api/cim-service/concurrentGuard`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Delete concurrentGuard

Method: DELETE

URL

`https://hostname/ibm/ife/api/cim-service/concurrentGuard`

For example:

`https://hostname/ibm/ife/api/cim-service/concurrentGuard`

Required properties:

- id

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

multimedia Service

Get multimedia

Method:

URL

`https://hostname/ibm/ife/api/cim-service/multimedia`

For example:

`https://hostname/ibm/ife/api/cim-service/multimedia`

Required properties: None

Optional properties:

- id,
- mrid,
- url,
- type

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[{"id":1,"mRID":"Pole_271759","url":"\/ibm\/ah\/images\/powerAssetPic\/
polepic1.JPG","type":"1"},
{"id":2,"mRID":"Pole_271759","url":"\/ibm\/ah\/images\/powerAssetPic\/
polepic2.JPG","type":"1"},
```

```
{"id":3,"mRID":"Pole_271759","url":"\/ibm\/ah\/images\/powerAssetPic\/
polepic3.JPG","type":"1"}]
```

Create multimedia

Method: POST

URL

`https://hostname/ibm/ife/api/cim-service/multimedia`

For example:

`https://hostname/ibm/ife/api/cim-service/multimedia`

Required properties:

- namespace,
- mRID,
- url,
- type

Optional properties:

- id,
- mrid,
- url,
- type

Request:

```
[{"id":1,"mRID":"Pole_271759","url":"\/ibm\/ah\/images\/powerAssetPic\/
polepic1.JPG","type":"1"},
{"id":2,"mRID":"Pole_271759","url":"\/ibm\/ah\/images\/powerAssetPic\/
polepic2.JPG","type":"1"},
{"id":3,"mRID":"Pole_271759","url":"\/ibm\/ah\/images\/powerAssetPic\/
polepic3.JPG","type":"1"}]
```

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

DGANALYSIS Service

Get DGANALYSIS

Method: GET

URL

`https://hostname/ibm/ife/api/cim-service/readingUpdate/SubstationTransformer/DGANALYSIS`

For example:

`https://hostname/ibm/ife/api/cim-service/readingUpdate/SubstationTransformer/DGANALYSIS`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[{"measurement":1360441,"timestamp":"2011-08-11T00:00:00+08:00",
"analysis_method": "Triangle","fault_type": "T1","analysis_time": null,
"editor": null,"comments": null} ]
```


Update DGANALYSIS

Method: PUT

URL

`https://hostname/ibm/ife/api/cim-service/readingUpdate/SubstationTransformer/DGANALYSIS`

For example:

`https://hostname/ibm/ife/api/cim-service/readingUpdate/SubstationTransformer/DGANALYSIS`

Required properties:

- Analysis_method,
- fault_type,
- analysis_time,
- editor,
- comments,
- measurement,
- timestamp

Optional properties: Not applicable

Request:


```
[{"measurement":1360441,"timestamp":"2011-08-11",
"analysis_method":"Triangle","fault_type":"T1",
"analysis_time":2018-11-20,"editor":"Bob","comments":""} ]
```


Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Asset Health REST service

Use the asset health REST service to access data generated by asset health analytics.

The data that is generated by the analytics includes:

- Asset analysis history: The history of the analysis records.
- Summary: The summary of the analysis result with a specific ID, where ID is one batch of analysis. The user may run multiple analyzes
- Container summary: The container summary of all assets.
- Summary aggregate: The count of the number of records.
- Containment: The containment relationship of the asset.
- Degradation curve: The graphical curve showing the degradation of the asset.

- Network trace: The distribution of all assets in the network.
- Health details: The health details of the asset, including factors, rating, and max rating.
- Analysis summary: The information for one batch of the analysis.
- Preset filter: The filter conditions on the Distribution page.

ah history service

The ah history service is used to put the REST data in the AH.HISTORY table. This service uses to get the analysis history data in database.

Table 52. AH.HISTORY table

Column#	Name	Description
1	ID	The analysis result id.
2	TIME	The analysis time of the result.
3	ACTIVE	0 or 1, 1 means the analysis result is the active one. Only one result should be active.
4	STARTYEAR	The start year of the analysis result.
5	NUMBEROFYEARS	The duration of the analysis result, like 20 years.
6	SUMMARYTABLE	The name of the summary result table.
7	CONTAINMENTTABLE	The name of the containment result table.
8	AHI_DETAILTABLE	The name of the ahi_detail result table.
9	CRITICALITY_DETAILTABLE	The name of the criticality_detail result table.
10	NETWORKTABLE	The name of the network result table.
11	DEGCURVETABLE	The name of the degradation curve result table.
12	COMMENTS	

Resource URI `/ibm/ife/ah/api/ah-service/history`

Get / ibm/ife/ah/api/ah-service/history

Method: GET

URL

`https://<hostname>/ibm/ife/ah/api/ah-service/history`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**

- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
    {
        "id": 1516681787,
        "active": false,
        "startYear": 2018,
        "numberOfYears": 20,
        "time": "2018-01-23T04:29:47Z",
        "comments": "created at the timestamp 2018-01-23 04:29:47"
    },
    {
        "id": 1533916876,
        "active": false,
        "startYear": 2018,
        "numberOfYears": 20,
        "time": "2018-08-10T16:01:16Z",
        "comments": "created at the timestamp 2018-08-10 16:01:16"
}
]
```


ah summary service

The ah summary service is used to REST the data in AH.SUMMARY_ID table. The generated data includes the comprehensive analysis result for the assets.

Table 53. AH.SUMMARY_ID table

Column	Name	Description
1	OID	The asset oid
2	MRID	The asset mRID
3	RESOURCETYPE	The resource type of the asset
4	SUBTYPE	The sub type of the asset
5	NAME	Name of the asset
6	DESCRIPTION	Description of the asset
7	ISCONTAINER	If resource is container
8	CONTAINER	Oid of the container
9	CONTAINERNAME	The container name of the asset
10	FEEDER	Feeder oid
11	FEEDERNAME	The name of the feeder
12	SUBSTATION	Substation oid
13	SUBSTATIONNAME	The name of the substation
14	PHASE	Phase information
15	LOCATIONDESCRIPTION	Location description
16	GEOMETRY	Geometry of the resource
17	AHI	The AHI value
18	AHI_CONDITION	The AHI condition value
19	AHI_INTEGRITY	The AHI integrity value
20	AHI_RELIABLITY	The AHI reliability value

Table 53. AH.SUMMARY_ID table (continued)		
Column	**Name**	**Description**
21	AGE	The real age of the asset
22	EFFECTIVEAGE	The effective age of the asset
23	CRITICALITY	The criticality value
34	CRITICALITY_INDEX	The criticality index value
25	CRITICALITY_CONDITION	The criticality condition value
26	NOC	The NOC value of the asset
27	FAILURE1	Failure value in the start year
28	FAILURE1_INDEX	Failure index value in the start year
29	FAILURE1_CONDITION	Failure condition value in the start year
30	...	...
31	FAILUREk	Failure value in the k year, k is number of years
32	FAILUREk_INDEX	Failure index value in the k year, k is number of years
33	FAILUREk_CONDITION	Failure condition value in the k year, k is number of years
34	RISK1	risk value in the start year
35	RISK1_CONDITION	risk condition value in the start year
36	...	...
37	RISKk	risk value in the k year, k is number of years
38	RISKk_CONDITION	risk condition value in the k year, k is number of years

Resource URI: `/ibm/ife/ah/api/ah-service/ history/{id:Number}/summary`

GET / ibm/ife/ah/ah/api/ah-service/history/{id:Number}/summary

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/ah/api/ah-service/history/{id:Number}/summary`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Too large to display.

ah container summary service

The ah summary service is used to put the REST data in the AH.SUMMARY_ID table, and do further calculations at the container level.

Resource URI: `/ibm/ife/ah/api/ah-service/ history/{id:Number}/container-summary`

GET / ibm/ife/ah/ah/api/ah-service/history/{id:Number}/container-summary

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/ah/api/ah-service/history/{id:Number}/container-summary`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
    {
        "count": 1,
        "ahi": null,
        "ahi_integrity": null,
        "ahi_reliability": null,
        "age": null,
        "effectiveAge": null,
        "criticality": null,
        "oid": 1000015,
        "risk": [
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null
        ],
        "risk_condition": [],
        "failure": [
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
            null,
```

```
            null,
            null,
            null,
            null,
            null,
            null
        ],
        "failure_condition": [],
        "mRID": "OHC_2030011",
        "resourceType": 12,
        "subType": 6,
        "name": "OHC_2030011",
        "description": "Overhead Cable 2030011",
        "container": 1027712,
        "geometry": "MULTILINESTRING ((-83.4174168 42.5891231, -83.4175098 42.5891283))"
    }]
```


ah summary aggregate service

The ah summary aggregate service is used to REST the data in the AH.SUMMARY_ID table and do further calculations.

Resource URI: `/ibm/ife/ah/api/ah-service/ history/{id:Number}/summary-aggregate`

GET / ibm/ife/ah/ah/api/ah-service/history/{id:Number}/ summary-aggregate

Method: GET

URL

`https://`*hostname*`/ibm/ife/ah/api/ah-service/history/{id:Number}/ summary-aggregate`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "count": 43122
    }
]
```


ah containment service

Table 54. AH.CONTAINMENT table

Column	Name	Description
1	OID	The container id.
2	NAME	The container name.
3	TIER	The container tier.
4	PARENT	The parent of the container.
5	PARENTTYPE	The type of parent.
6	PARENTNAME	The name of parent.
7	CONTAINER	The container of the container.

Table 54. AH.CONTAINMENT table (continued)		
Column	**Name**	**Description**
8	RESOURCETYPE	The resourceType.

Resource URI

GET / ibm/ife/ah/ah/api/ah-service/history/{id:Number}/containment

Method: GET

URL

`https://hostname/ibm/ife/ah/api/ah-service/history/{id:Number}/containment`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

```

```

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
{
        "oid": 1000001,
        "name": "DT_155362",
        "tier": 1,
        "parent": 1027713,
        "parentType": 4,
        "parentName": "LILY 9120",
        "container": 1027713,
        "resourceType": 10
    },
    {
        "oid": 1000001,
        "name": "DT_155362",
        "tier": 2,
        "parent": 1027697,
        "parentType": 3,
        "parentName": "LILY",
        "container": 1027713,
        "resourceType": 10
    }
```


ah degcurve service

The ah degcurve service is used put the REST data in the AH.DEGCURVE table.

Table 55. AH.DEGCURVE table		
Column	**Name**	**Description**
1	RESROUCETYPE	The resource type of oil circuit breaker.
2	SUBTYPE	The sub type of oil circuit breaker
3	AGE	The age of the oil circuit breaker.
4	FAILURE	The failure probability of the age.

Resource URI: `/ibm/ife/ah/api/ah-service/ history/{id:Number}/degcurve`

GET/ibm/ife/ah/ah/api/ah-service/history/{id:Number}/ degcurve?resourceType=< resourceType >

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/ah/api/ah-service/history/{id:Number}/degcurve?resourceType=`*`resourceType`*

Required properties:

- property name="resourceType"
- type="Number"

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[{"subTypeName":{"value":"Wood Power Pole","key":"Pole_3","group":"IFESubType"},
"curve":[{"failure":0.00,"age":1},{"failure":0.00,"age":2},{"failure":0.01,"age":3},
{"failure":0.03,"age":4},{"failure":0.07,"age":5},{"failure":0.16,"age":6}
,{"failure":0.30,"age":7},{"failure":0.51,"age":8},{"failure":0.81,"age":9},
{"failure":1.24,"age":10},{"failure":1.81,"age":11},{"failure":2.55,"age":12},
{"failure":3.50,"age":13},{"failure":4.67,"age":14},{"failure":6.11,"age":15},
{"failure":7.83,"age":16},{"failure":9.86,"age":17},
{"failure":12.23,"age":18},{"failure":14.94,"age":19},{"failure":18.01,"age":20},
{"failure":21.43,"age":21},{"failure":25.20,"age":22},
{"failure":29.29,"age":23},{"failure":33.68,"age":24},{"failure":38.33,"age":25},
{"failure":43.17,"age":26},{"failure":48.15,"age":27},
{"failure":53.21,"age":28},{"failure":58.25,"age":29},{"failure":63.21,"age":30},
{"failure":68.00,"age":31},{"failure":72.56,"age":32},
{"failure":76.82,"age":33},{"failure":80.73,"age":34},{"failure":84.25,"age":35},
{"failure":87.36,"age":36},{"failure":90.04,"age":37},
{"failure":92.31,"age":38},{"failure":94.19,"age":39},{"failure":95.70,"age":40},
{"failure":96.90,"age":41},{"failure":97.81,"age":42},
{"failure":98.50,"age":43},{"failure":98.99,"age":44},{"failure":99.34,"age":45},
{"failure":99.59,"age":46},{"failure":99.74,"age":47},
{"failure":99.85,"age":48},{"failure":99.91,"age":49},{"failure":99.95,"age":50},
{"failure":99.97,"age":51},{"failure":99.98,"age":52},
{"failure":99.99,"age":53},{"failure":99.99,"age":54},{"failure":99.99,"age":55},
{"failure":99.99,"age":56},{"failure":99.99,"age":57},
{"failure":99.99,"age":58},{"failure":99.99,"age":59},{"failure":99.99,"age":60},
{"failure":99.99,"age":61},{"failure":99.99,"age":62},
{"failure":100.00,"age":63},{"failure":100.00,"age":64},{"failure":100.00,"age":65},
{"failure":100.00,"age":66},{"failure":100.00,"age":67},
{"failure":100.00,"age":68},{"failure":100.00,"age":69},{"failure":100.00,"age":70},
{"failure":100.00,"age":71},{"failure":100.00,"age":72},
{"failure":100.00,"age":73},{"failure":100.00,"age":74},{"failure":100.00,"age":75},
{"failure":100.00,"age":76},{"failure":100.00,"age":77},
{"failure":100.00,"age":78},{"failure":100.00,"age":79},{"failure":100.00,"age":80},
{"failure":100.00,"age":81},{"failure":100.00,"age":82},
{"failure":100.00,"age":83},{"failure":100.00,"age":84},{"failure":100.00,"age":85},
{"failure":100.00,"age":86},{"failure":100.00,"age":87},
{"failure":100.00,"age":88},{"failure":100.00,"age":89},{"failure":100.00,"age":90},
{"failure":100.00,"age":91},{"failure":100.00,"age":92},
{"failure":100.00,"age":93},{"failure":100.00,"age":94},{"failure":100.00,"age":95},
{"failure":100.00,"age":96},{"failure":100.00,"age":97},
{"failure":100.00,"age":98},{"failure":100.00,"age":99},
{"failure":100.00,"age":100}],"subType":3,"resourceTypeName":
{"value":"Pole","key":"Pole","group":"IFEResourceType"},"resourceType":8}]
```


ah network source service

The ah network service is used to put the REST data in the AH.NETWORK table. The generated data includes the asset position for the entire network.

Table 56. AH.NETWORK table		
Column	**Name**	**Description**
1	FEEDER	The feeder oid of the source and the load.
2	SOURCE	The asset oid of the source.
3	LOAD	The asset oid of the load.
4	OID	The asset oid of the assets in the network.
5	RESOURCETYPE	The resource type of the asset.
6	SUBTYPE	The sub type of the asset.
7	PHASE	The phase of the asset.
8	CONTAINER	The container of the asset.
9	ORDER	The order of the asset.
10	TIER	The tier of the asset.
11	PATHMIN	The minimum path index of the asset.
12	PATHMAX	The maximum path index of the asset.

Resource URI: `/ibm/ife/ah/api/ah-service/history/{id:Number}/network/source`

GET/ibm/ife/ah/ah/api/ah-service/history/{id:Number}/network?type=<type>&source=<source>&selected=<oid>

Method: GET

URL

```
https://hostname/ibm/ife/ah/api/ah-service/history/{id:Number}/network?
type=<type> &source=<source>&selected=oid
```

For example:

```
/ibm/ife/ah/ah/api/ah-service/history/12345/network?
type=downstream&source=1004053&selected=1004053
```

Required properties:

- <property name="type" type="String" filter="false"/>
- <property name="selected" type="Number" filter="false"/>
- <property name="source" type="Number" filter="false"/>

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:

```
[{"oid":1067376,"mRID":"brk_503578","resourceType":18,"subType":1,"name":"brk_503578",
"container":1028780,"phase":7,"order":0,"tier":30,"pathMin":0,"pathMax":0},
{"oid":1023210,"mRID":"UGLS_2040903","resourceType":13,"subType":0,"name":"UGLS_2040903",
"container":1013836,"phase":null,"order":2,"tier":28,"pathMin":0,"pathMax":0},
{"oid":1067450,"mRID":"brk_503577","resourceType":18,"subType":1,"name":"brk_503577",
"container":1028781,"phase":7,"order":3,"tier":60,"pathMin":1,"pathMax":1},
{"oid":1023207,"mRID":"UGLS_2040902","resourceType":13,"subType":0,"name":"UGLS_2040902",
"container":1013834,"phase":null,"order":5,"tier":58,"pathMin":1,"pathMax":1},
{"oid":1016242,"mRID":"BBLS_1974058","resourceType":13,"subType":0,"name":"BBLS_1974058",
"container":1013091,"phase":null,"order":7,"tier":56,"pathMin":1,"pathMax":1},
{"oid":1067490,"mRID":"brk_272530","resourceType":18,"subType":17,"name":"brk_272530",
"container":1028781,"phase":7,"order":9,"tier":54,"pathMin":1,"pathMax":1}
]
```


ah detail service

The ah detail service is used to put the REST data in the AH.DEGCURVE table. The generated data includes the degradation curves of all asset types.

Table 57. AH.AHI_DETAIL_ table

Column	Name	Description
1	OID	The asset oid.
2	FACTOR	The control criteria.
3	RATING	The condition rating value.
4	MAXRATING	The max condition rating value.
5	WEIGHT	The weight of the control criteria.
6	INTEGRITY	The AHI integrity value.
7	RELIABILITY	The AHI reliability value.
8	VALUE	The condition rating value.
9	VALUETIME	The record time of the condition rating value.

Resource URI: `/ibm/ife/ah/api/ah-service/ history/{id:Number}/detail`

GET/ibm/ife/ah/ah/api/ah-service/history/{id:Number}/ detail?assetId==<assetId>&time=<time>

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/ah/api/ah-service/history/{id:Number}/ detail?assetId=`*`assetId`*`&time=`*`time`*

Required properties:

- <property name="assetId" type="Number" />
- <property name="time" type="Number" /> (for example, 2019)

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[{"noc":2476,"failure_condition":"A","numberOfYears":20,"feeder":"LILY",
```

```
"criticality":32.29,"stage3Rating":40,
"isActive":true,"operatingVoltage":120000,"ahi_integrity":83.96,
"manufacturer":"813290","nominalVoltage":null,"mRID":"ST_1393137",
"criticalityDetail":[{"valueTime":"2018-08-14
07:49:16.0","rating":2476,"weight":1.00,"maxRating":7668,"factor":"Number of Customers",
"value":2476.00}],"riskOverTime":[{"risk":6.0,"time":2018},{"risk":6.6,"time":2019},
{"risk":7.24,"time":2020},{"risk":7.9,"time":2021},{"risk":8.6,"time":2022},
{"risk":9.33,"time":2023},{"risk":10.09,"time":2024},{"risk":10.87,"time":2025},
{"risk":11.68,"time":2026},{"risk":12.51,"time":2027},{"risk":13.36,"time":2028},
{"risk":14.23,"time":2029},
{"risk":15.1,"time":2030},{"risk":15.99,"time":2031},{"risk":16.88,"time":2032},
{"risk":17.77,"time":2033},{"risk":18.66,"time":2034},{"risk":19.54,"time":2035},
{"risk":20.41,"time":2036},{"risk":21.26,"time":2037}],"assetId":1006628,
"phaseDesignation":7,"stage1Rating":24,"ratedkva":24000.0,"effectiveAge":37.0,"stage2Rating"
:32,"ahi":55.71,
"serialNumber":"1393137","winterEmergencyRating":3374,"risk_condition":"A",
"subTypeCode":"54","startYear":2018,"installationDate":"2003-03-28","summerNormalRating":266
1,
"failureOverTime":[{"failure":18.6,"time":2018},{"failure":20.46,"time":2019},
{"failure":22.42,"time":2020},{"failure":24.49,"time":2021},{"failure":26.65,"time":2022},
{"failure":28.91,"time":2023},{"failure":31.25,"time":2024},{"failure":33.68,"time":2025},
{"failure":36.19,"time":2026},{"failure":38.76,"time":2027},{"failure":41.39,"time":2028},
{"failure":44.07,"time":2029},{"failure":46.79,"time":2030},{"failure":49.53,"time":2031},
{"failure":52.29,"time":2032},{"failure":55.04,"time":2033},{"failure":57.79,"time":2034},
{"failure":60.52,"time":2035},{"failure":63.21,"time":2036},{"failure":65.85,"time":2037}],
"summerEmergencyRating":2771,"removalDate":null,"ahi_condition":"C","criticality_condition":
"B",
"failure":18.6,"ahiDetail":[{"integrity":90.07,"valueTime":"2016-11-29
00:00:00.0","reliability":6.06,"rating":3,"weight":1.00,"maxRating":4,"factor":"Bushing
Condition","value":3.00},
{"integrity":90.07,"valueTime":"2017-04-24
00:00:00.0","reliability":6.06,"rating":4,"weight":4.00,"maxRating":4,"factor":"Bushing DGA
Oil Analysis","value":4.00},
{"integrity":90.07,"valueTime":"2017-06-07
00:00:00.0","reliability":6.06,"rating":1,"weight":1.00,"maxRating":4,"factor":"Conservator
\/Oil Preservation System (Airbag Integrity)",
"value":1.00},{"integrity":90.07,"valueTime":"2016-12-29
00:00:00.0","reliability":6.06,"rating":0,"weight":4.00,"maxRating":4,"factor":"DGA
Moisture Metal Content",
"value":0.00},{"integrity":90.07,"valueTime":"2016-05-12
00:00:00.0","reliability":6.06,"rating":0,"weight":4.00,"maxRating":4,"factor":"DGA Oil
Analysis",
"value":0.00},{"integrity":90.07,"valueTime":"2017-06-11
00:00:00.0","reliability":6.06,"rating":3,"weight":4.00,"maxRating":4,"factor":"Furan Oil
Analysis",
"value":3.00},{"integrity":90.07,"valueTime":"2016-09-14
00:00:00.0","reliability":6.06,"rating":2,"weight":1.00,"maxRating":4,"factor":"Gaskets
Seals and Pressure Relief",
"value":2.00},{"integrity":90.07,"valueTime":"2017-07-07
00:00:00.0","reliability":6.06,"rating":0,"weight":1.00,"maxRating":4,"factor":"LTC Control
and Mechanism Cabinet",
"value":0.00},{"integrity":90.07,"valueTime":"2017-04-02
00:00:00.0","reliability":6.06,"rating":3,"weight":1.00,"maxRating":4,"factor":"Oil
Leaks","value":3.00},
{"integrity":90.07,"valueTime":"2017-01-16
00:00:00.0","reliability":6.06,"rating":3,"weight":2.00,"maxRating":4,"factor":"Overall
Power Transformer","value":3.00},
{"integrity":90.07,"valueTime":"2017-08-09
00:00:00.0","reliability":6.06,"rating":1,"weight":3.00,"maxRating":4,"factor":"Overall Tap
Changer Condition",
"value":1.00},{"integrity":90.07,"valueTime":"2017-03-02
00:00:00.0","reliability":6.06,"rating":4,"weight":1.00,"maxRating":4,"factor":"Radiators\/
Cooling System",
"value":4.00},{"integrity":90.07,"valueTime":"2016-10-22
00:00:00.0","reliability":6.06,"rating":2,"weight":2.00,"maxRating":4,"factor":"ST
Thermograph (IR)","value":2.00},
{"integrity":90.07,"valueTime":"2018-08-11
06:40:27.0","reliability":6.06,"rating":0,"weight":1.00,
"maxRating":4,"factor":"ST_VI_COMPONENT","value":0.00},{"integrity":90.07,
"valueTime":"2018-08-11
06:40:49.0","reliability":6.06,"rating":1,"weight":1.00,"maxRating":4,"factor":"ST_VI_TAP",
"value":1.00},{"integrity":90.07,"valueTime":"2016-06-23 00:00:00.0",
"reliability":6.06,"rating":0,"weight":1.00,"maxRating":4,"factor":"Tank
Condition","value":0.00},{"integrity":90.07,"valueTime":"2017-03-17
00:00:00.0","reliability":6.06,
"rating":0,"weight":1.00,"maxRating":4,"factor":"Tank Leaks","value":0.00},
{"integrity":90.07,"valueTime":"2016-10-11
00:00:00.0","reliability":6.06,"rating":4,"weight":3.00,
"maxRating":4,"factor":"TC Oil Quality Tests","value":4.00},
{"integrity":90.07,"valueTime":"2017-05-26
00:00:00.0","reliability":6.06,"rating":2,"weight":4.00,
```

```
"maxRating":4,"factor":"Winding Doble
Test","value":2.00}],"winterNormalRating":3374,"name":"ST_1393137",
"ahi_reliability":0.0,"risk":6.0,"time":2018,"age":15.38,
"resourceTypeName":{"value":"Substation
Transformer","key":"SubstationTransformer","group":"IFEResourceType"},"resourceType":9}]
```


ah analysis summary service

The ah analysis summary service is used to put the REST data in the AH.HISTORY table.

Table 58. AH.HISTORY table

Column#	Name	Description
1	ID	The analysis result id.
2	TIME	The analysis time of the result.
3	ACTIVE	0 or 1, 1 means the analysis result is the active one. Only one result should be active.
4	STARTYEAR	The start year of the analysis result.
5	NUMBEROFYEARS	The duration of the analysis result, like 20 years.
6	SUMMARYTABLE	The name of the summary result table.
7	CONTAINMENTTABLE	The name of the containment result table.
8	AHI_DETAILTABLE	The name of the ahi_detail result table.
9	CRITICALITY_DETAILTABLE	The name of the criticality_detail result table.
10	NETWORKTABLE	The name of the network result table.
11	DEGCURVETABLE	The name of the degradation curve result table.
12	COMMENTS	

Resource URI: `/ibm/ife/ah/api/ah-service/ history/{id:Number}/detail`

GET / ibm/ife/ah/ah/api/ah-service/analysis-summary

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/ah/api/ah-service/analysis-summary`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
    {
        "id": 1534239107,
        "startYear": 2018,
        "numberOfYears": 20
    }
]
```


ah assetDataColumn service

Resource URI: `/ibm/ife/ah/api/ah-service/assetDataColumn`

GET / ibm/ife/ah/ah/api/ah-service/assetDataColumn?dataType=<dataType> &resourceType=<resourceType>

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/ah/api/ah-service/assetDataColumn?dataType=`*`dataType`* `&resourceType=`*`resourceType`*

For example:

`/ibm/ife/ah/api/ah-service/assetDataColumn?dataType=master&resourceType=SubstationTransformer`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

```
```

Status:

- Successful **200 OK**
- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
{"index":0,"column":"OID"},
{"index":1,"column":"SERIALNUMBER"},
{"index":2,"column":"ISACTIVE"},
{"index":3,"column":"INSTALLATIONDATE"},
{"index":4,"column":"REMOVALDATE"},
{"index":5,"column":"OPERATINGVOLTAGE"},
{"index":6,"column":"SUMMEREMERGENCYRATING"},
{"index":7,"column":"SUMMERNORMALRATING"},
{"index":8,"column":"WINTEREMERGENCYRATING"},
{"index":9,"column":"WINTERNORMALRATING"}
…
]
```


ah presetFilter service

The ah presetFilter service is used to put the REST data in the AH.DEGCURVE table. The generated data includes: id, name, and filter_condition. The id is the identity of the filter_condition.

Resource URI: `/ibm/ife/ah/api/ah-service/presetFilter`

GET / ibm/ife/ah/ah/api/ah-service/ presetFilter

Method: GET

URL

`https://<hostname>/ibm/ife/ah/api/ah-service/ presetFilter`

Required properties: None

Optional properties:

- code

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "id": 1,
        "name": "3333",
        "filter_condition": {
            "container": [
                "1027696",
                "1027716",
                "1027717",
                "1027718",
                "1027719",
                "1027720",
                "1027721",
                "1027722",
                "1027723",
                "1027724",
                "1027725",
                "1027712",
                "1027698",
                "1027702",
                "1027703",
                "1027704",
                "1027705",
                "1027706",
                "1027707",
                "1027708"
            ],
            "advancedFilter": "[{\"id\":8,\"cql\":\"(SERIALNUMBER like '%2%') AND
(INSTALLATIONDATE > '2013-08-01' and INSTALLATIONDATE < '2018-08-01')\"}]",
            "parent": 1027694,
            "analysisObject": {
                "analysisTime": {
                    "comments": "created at the timestamp 2018-08-11 06:47:17",
                    "numberOfYears": 20,
                    "startYear": 2018,
                    "active": true,
                    "id": 1533970037,
                    "time": "2018-08-11T06:47:17Z"
                },
                "analysisHistory": "UsePreviousAnalysis"
            },
            "advancedFilterObject": {
                "editFilterObjects": [
                    {
                        "assetClassType": "Pole",
                        "propertyDatas": {
                            "Serial Number": {
                                "propertDescription": "Serial Number, contains 2",
                                "assetClassId": 8,
                                "propertyFilter": {
                                    "dataType": "VARCHAR",
                                    "values": [
                                        {
                                            "disabled": false,
                                            "value": "2"
                                        }
                                    ],
                                    "assetClassId": 8,
```

```
                                    "name": "Serial Number",
                                    "id": "SERIALNUMBER",
                                    "operation": "contains"
                                },
                                "cql": "SERIALNUMBER like '%2%'"
                            }
                        },
                        "openCondition": true
                    },
                    {
                        "assetClassType": "Pole",
                        "propertyDatas": {
                            "Installation Date": {
                                "propertDescription": "Installation Date, is between
2013-08-01 and 2018-08-01",
                                "assetClassId": 8,
                                "propertyFilter": {
                                    "date_range": "between",
                                    "endDate": "2018-08-01",
                                    "dataType": "DATE",
                                    "assetClassId": 8,
                                    "name": "Installation Date",
                                    "startDate": "2013-08-01"
                                },
                                "cql": "INSTALLATIONDATE > '2013-08-01' and
INSTALLATIONDATE < '2018-08-01'"
                            }
                        },
                        "openCondition": true
                    }
                ],
                "criteria": "allCriteria"
            },
            "resourceType": [
                8,
                12
            ]
        }
    }
]
```

POST / ibm/ife/ah/ah/api/ah-service/presetFilter

Method: POST

URL

`https://`*`hostname`*`/ibm/ife/ah/api/ah-service/presetFilter`

Required properties:

- <property name="name" type="String" />
- <property name="filter_condition" type="Object"/>

Optional properties: Not applicable

Request:

```
"0":{
    "name": "apitest",
    "filter_condition":{
        "resourceType":[
            {"0":"9"}
        ],
        "parent":"1028753",
        "analysisObject":{
            "analysisTime":{
                "id":1542033944,
                "active":"true",
                "startYear":2018,
                "numberOfYears":20,
                "time":"2018-11-12T14:45:44+08:00",
                "comments":"created at the timestamp 2018-11-12 14:45:44"
            },
            "analysisHistory":"UseSystemDefault"
        }
        "advancedFilterObject":{
            "criteria":"allCriteria",
            "editFilterObject":[]
        }
```

```
    }
}
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

PUT /ibm/ife/vi/api/ah-service/ presetFilter?id=<id>&name=<name>

Method: PUT

URL

`https://`*`hostname`*`/ibm/ife/vi/api/ah-service/ presetFilter?id=`*`id`*`&name=`*`name`*

Required properties:

- <property name="id" type="Number" />
- <property name="name" type="String" />

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

Delete

Method: DELETE

URL

`https://<hostname> /ibm/ife/vi/api/ah-service/ presetFilter?id=<id>`

Required properties:

- <property name="id" type="Number" />

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

Asset Investment Plan service

Use the asset investment plan REST service to access data generated by asset health analytics, and to produce the investment plan.

The data that is generated by the asset investment plan:

- Default-cost: The default cost for one asset type.

- Project: A single asset investment plan.
- Scope: The scope of one project.
- Scenario: The investment scenario.
- Summary: The summary of investment plans for different assets.
- Replace-summary: The summary of replace plans for different assets.
- Container-summary: The summary of the health scores of the involved assets.
- Summary-aggregate: The number of assets in one project.

The data in asset investment plan follows the Asset Investment Plan schema in DB2. The investment plan is based on the asset health scores from asset health analysis.

Base URI: `/ibm/ife/aip/api/aip-service`

aip default-cost service

The aip default-cost service is used to store the default cost of the investment plan.

Table 59. AIP.DEFAULT_COST

Column	Name	Description
1	RESOURCETYPE	The asset resource type.
2	SUBTYPE	The asset subtype.
3	REPLACEMENT_COST	The replacement cost.

Resource URI: `/ibm/ife/aip/api/aip-service/default-cost`

GET / ibm/ife/aip/api/aip-service/default-cost

Method: GET

URL

`https://<hostname>/ibm/ife/aip/api/aip-service/default-cost`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

```
[
    {
        "resourceType": 9,
        "subType": 54,
        "replacement_cost": 10
    }
]
```

POST / ibm/ife/aip/api/aip-service/default-cost

Method: POST

URL

`https://<hostname>/ibm/ife/aip/api/aip-service/default-cost`

Required properties:

- <property name="resourceType" type="Number" />
- <property name="subType" type="Number" />
- <property name="replacement_cost" type="Number" />

Optional properties: Not applicable

Request:

```
body:
[
    {
        "resourceType": 9,
        "subType": 54,
        "replacement_cost": 10
    }
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

aip project service

The aip project service is used to store the project information of the investment plan.

Table 60. AIP.PROJECT table

Column	Name	Description
1	ID	The project ID.
2	NAME	The project name.
3	DESCRIPTION	The project description.
4	HEALTH_RESULT	The asset health result that is used in this project.
5	FILTER_CONDITION	The predefined filter condition of the project.

Resource URI: `/ibm/ife/aip/api/aip-service/project`

GET / ibm/ife/aip/api/aip-service/project, GET / ibm/ife/aip/api/aip-service/project?id=<id>, GET / ibm/ife/aip/api/aip-service/project?name=<name>

Method: GET

URL

`https://hostname/ibm/ife/aip/api/aip-service/project`

Required properties: None

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:

```
Response    {
        "id": 49,
        "name": "test sniff pole",
        "description": "test sniff pole",
        "health_result": 1500567605,
        "filter_condition": {
            "container": [
                1028754,
                1028766,
                1028767,
                1028768,
                1028769,
                1028770,
                1028771,
                1028772,
                1028755,
                1028759,
                1028760,
                1028761,
                1028762,
                1028763,
                1028764,
                1028765
            ],
            "parent": 1028751,
            "resourceType": [
                8
            ]
        }
    }
```

POST / ibm/ife/aip/api/aip-service/project

Method: POST

URL

`https://hostname/ibm/ife/aip/api/aip-service/project`

Required properties:

- <property name="name" type="String" />
- <property name="description" type="String" />
- <property name="health_result" type="Number" />
- <property name="filter_condition" type="Object"/>

Optional properties: Not applicable

Request:

```
body:
[{
        "name": "apitest2",
        "description": "apitest2",
        "health_result": "1500488037",
        "filter_condition": {
            "parent":"",
            "resourceType": [9]
        }
    }]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

DELETE / ibm/ife/aip/api/aip-service/project?id=<id>

Method: DELETE

URL

`https://*hostname*/ibm/ife/aip/api/aip-service/project?id=*id*`

Required properties:

- <property name="id" type="Number" />

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

aip scope service

The aip scope service is used to return the scope of project.

Table 61.

Column	Name	Description
1	ID	The ID.
2	RESOURCETYPE	The asset resource type
3	SUBTYPE	The asset sub type.
4	REPLACEMENT_COST	The cost of replacement.

Resource URI: `/ibm/ife/aip/api/aip-service/project/{id:Number}/scope`

GET / ibm/ife/aip/api/aip-service/ project/{id:Number}/scope

Method: GET

URL

`https://*hostname*/ibm/ife/aip/api/aip-service/ project/{id:Number}/scope`

Required properties: None.

Optional properties: Not applicable.

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "id": 49,
        "resourceType": 8,
        "subType": 0,
        "replacement_cost": 100
    },
    {
```

```
        "id": 49,
        "resourceType": 8,
        "subType": 3,
        "replacement_cost": 1
    }
]
```

POST / ibm/ife/aip/api/aip-service/project/{id:Number}/scope

Method: POST

URL

`https://hostname/ibm/ife/aip/api/aip-service/project/{id:Number}/scope`

Required properties:

- <property name="resourceType" type="Number" />
- <property name="subType" type="Number" />
- <property name="replacement_cost" type="Number" />

Optional properties: Not applicable

Request:

```
body:
[{
        "resourceType": 8,
        "subType": 3,
        "replacement_cost": 100
    }]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

PUT / ibm/ife/aip/api/aip-service/project/{id:Number}/scope

Method: PUT

URL

`https://hostname/ibm/ife/aip/api/aip-service/project/{id:Number}/scope`

Required properties:

- <property name="resourceType" type="Number" />
- <property name="subType" type="Number" />
- <property name="replacement_cost" type="Number" />

Optional properties: Not applicable

Request:

```
body:
[{
        "resourceType": 8,
        "subType": 3,
        "replacement_cost": 100
    }]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**

- Forbidden: **403 Forbidden**

Response: Not applicable.

aip scenario service

The aip scenario service is used to return the investment scenario of the project.

Table 62. AIP.PROJECT_SCOPE table

Column	Name	Description
1	ID	The ID.
2	PROJECT	The project scenario.
3	NAME	The name of the scenario.
4	DESCRIPTION	The scenario description.
5	STARTYEAR	The start year.
6	NUMBEROFYEARS	The number of years.
7	FAILURE_THRESHOLD	The threshold of failure.
8	TYPE	The scenario type.
9	TOTAL_BUDGET	The total budget of the scenario.
10	TARGET_BUDGET	The target budget of the scenario.
11	TARGET_RISK	The target risk of the scenario.
12	RESULT_TABLE	The table of results.
13	STATUS	The status of the scenario.
14	ANALYSISSTART	The analysis start.
15	ANYLYSISEND	The analysis end.

Resource URI: `/ibm/ife/aip/api/aip-service/scenario`

GET / ibm/ife/aip/api/aip-service/scenario, GET / ibm/ife/aip/api/aip-service/scenario?id=<id>, GET / ibm/ife/aip/api/aip-service/scenario?name=<name>

Method: GET

URL

`https://`*`hostname`*`/ibm/ife/aip/api/aip-service/scenario`

Required properties: None.

Optional properties: Not applicable.

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
    {
        "id": 58,
        "project": 49,
```

```
        "name": "Sustain",
        "description": null,
        "startYear": 2017,
        "numberOfYears": 20,
        "failure_threshold": 99,
        "type": 1,
        "total_budget": null,
        "target_budget": null,
        "target_risk": "[0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,
0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91]",
        "result_table": "AIP.SCENARIO_RESULT_1500606306636",
        "status": 2,
        "analysisStart": "2017-07-21T11:05:02.837+08:00",
        "analysisEnd": "2017-07-21T11:05:06.910+08:00"
    }
]
```

POST / ibm/ife/aip/api/aip-service/scenario

Method: POST

URL

`https://hostname/ibm/ife/aip/api/aip-service/scenario`

Required properties:

- <property name="id" type="Number" />
- <property name="project" type="Number" />
- <property name="name" type="String" />
- <property name="description" type="String" />
- <property name="startYear" type="Number" />
- <property name="numberOfYears" type="Number" />
- <property name="failure_threshold" type="Number" />
- <property name="type" type="Number" />
- <property name="total_budget" type="Number" />
- <property name="target_budget" type="String" />
- <property name="target_risk" type="String" />
- <property name="result_table" type="String" />
- <property name="status" type="Number" />
- <property name="analysisStart" type="Timestamp" />
- <property name="analysisEnd" type="Timestamp" />

Optional properties: Not applicable

Request:

```
body:
[
    {"project": 49,
        "name": "Sustain",
        "description": null,
        "startYear": 2017,
        "numberOfYears": 20,
        "failure_threshold": 99,
        "type": 1,
        "total_budget": null,
        "target_budget": null,
        "target_risk": "[0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,
0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91]",
        "result_table": "AIP.SCENARIO_RESULT_1500606306636",
        "status": 2,
        "analysisStart": "2017-07-21T11:05:02.837+08:00",
        "analysisEnd": "2017-07-21T11:05:06.910+08:00"
    }
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable

PUT / ibm/ife/aip/api/aip-service/ scenario

Method: PUT

URL

`https://*hostname*/ibm/ife/aip/api/aip-service/scenario`

Required properties:

- <property name="id" type="Number" />
- <property name="project" type="Number" />
- <property name="name" type="String" />
- <property name="description" type="String" />
- <property name="startYear" type="Number" />
- <property name="numberOfYears" type="Number" />
- <property name="failure_threshold" type="Number" />
- <property name="type" type="Number" />
- <property name="total_budget" type="Number" />
- <property name="target_budget" type="String" />
- <property name="target_risk" type="String" />
- <property name="result_table" type="String" />
- <property name="status" type="Number" />
- <property name="analysisStart" type="Timestamp" />
- <property name="analysisEnd" type="Timestamp" />

Optional properties: Not applicable

Request:

```
body:
[
    {
        "id":58,
        "project": 49,
        "name": "apitest",
        "description": "new description",
        "startYear": 2017,
        "numberOfYears": 20,
        "failure_threshold": 99,
        "type": 1,
        "total_budget": 1000,
        "target_budget": "",
        "target_risk": "[0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,
0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91,0.91]",
        "result_table": "AIP.SCENARIO_RESULT_1500606306123",
        "status": 2,
        "analysisStart": "2017-07-21T11:05:02.837+08:00",
        "analysisEnd": "2017-07-21T11:05:06.910+08:00"
    }
]
```

Status:

- Successful **201 Created**

- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

DELETE / ibm/ife/aip/api/aip-service/ scenario?id=<id>

Method: DELETE

URL

`https://*hostname*/ibm/ife/aip/api/aip-service/ scenario?id=*id*`

Required properties:

- <property name="id" type="Number" />

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

aip summary service

The aip summary service is used to return the investment scenario summary of the project.

The aip summary service uses asset health and asset investment plan as the data source.

Resource URI: `/ibm/ife/aip/api/aip-service/scenario/{id:Number}/summary`

GET / ibm/ife/aip/api/aip-service/scenario/{id:Number}/summary

Method: GET

URL

`https://<hostname>/ibm/ife/aip/api/aip-service/scenario/{id:Number}/summary`

Required properties: None.

Optional properties: Not applicable.

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
    {
        "oid": 1001322,
        "mRID": "DT_107492",
        "resourceType": 10,
        "subType": 38,
        "name": "DT_107492",
        "description": "Distribution Transformer 107492",
        "isContainer": false,
        "container": 1028774,
        "containerName": "NIXON8197",
        "feeder": 1028774,
        "feederName": "NIXON8197",
```

```
        "substation": 1028753,
        "substationName": "NIXON",
        "phase": 1,
        "locationDescription": "",
        "geometry": "POINT (-83.3965650 42.6616911)",
        "ahi": 76.31,
        "ahi_condition": "B",
        "ahi_integrity": 76,
        "ahi_reliability": 54.84,
        "age": 3.17,
        "age_index": 3,
        "effectiveAge": 18,
        "effectiveAge_index": 18,
        "criticality": 5.05,
        "criticality_index": 1,
        "criticality_condition": "A",
        "replace": [
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            1
        ],
        "cost": [
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0
        ],
        "comment": [
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "",
            "risk"
        ],
```

```
            "failure": [
                4.05,
                5.01,
                6.11,
                7.37,
                8.8,
                10.42,
                12.23,
                14.23,
                16.43,
                18.83,
                21.43,
                24.22,
                27.21,
                30.36,
                33.68,
                37.14,
                40.73,
                44.41,
                48.15,
                51.94
            ],
            "risk": [
                0.2,
                0.25,
                0.3,
                0.37,
                0.44,
                0.52,
                0.61,
                0.71,
                0.82,
                0.95,
                1.08,
                1.22,
                1.37,
                1.53,
                1.7,
                1.87,
                2.05,
                2.24,
                2.43,
                2.62
            ],
            "failure_condition": [
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "B",
                "B",
                "B",
                "B",
                "B",
                "B",
                "C"
            ],
            "risk_condition": [
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
                "A",
```

```
            "A",
            "A",
            "A",
            "A",
            "A"
        ]
    }]
```

aip replace-summary service

The aip replace-summary service is used to return the investment scenario replacement summary of the project.

The aip replace-summary service uses the asset health and asset investment plan data as the data source.

Resource URI: `/ibm/ife/aip/api/aip-service/scenario/{id:Number}/replace-summary`

GET / ibm/ife/aip/api/aip-service/scenario/{id:Number}/replace-summary

Method: GET

URL

`https://<hostname>/ibm/ife/aip/api/aip-service/scenario/{id:Number}/replace-summary`

Required properties: None.

Optional properties: Not applicable.

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "special_id": "2036DT_107492",
        "oid": 1001322,
        "time": 2036,
        "mRID": "DT_107492",
        "resourceType": 10,
        "subType": 38,
        "name": "DT_107492",
        "criticality": 5.05,
        "criticality_index": 1,
        "criticality_condition": "A",
        "replace": 1,
        "cost": 0,
        "comment": "risk",
        "failure": 51.94,
        "failure_condition": "C",
        "risk": 2.62,
        "risk_condition": "A",
        "geometry": "POINT (-83.3965650 42.6616911)"
    }]
```


aip container-summary service

The aip container-summary service is used to return the investment scenario container summary of the project.

The aip container-summary service uses the asset health and asset investment plan data as the data source.

Resource URI: `/ibm/ife/aip/api/aip-service/scenario/{id:Number}/container-summary`

GET /ibm/ife/aip/api/aip-service/scenario/{id:Number}/container-summary

Method: GET

URL

`https://hostname/ibm/ife/aip/api/aip-service/scenario/{id:Number}/container-summary`

Required properties: None.

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable


```
[
    {
        "count": 329,
        "ahi": 49.18,
        "ahi_integrity": 79.89,
        "ahi_reliability": 28.18,
        "age": 3.44,
        "effectiveAge": 29.72,
        "criticality": 0.77,
        "oid": 1028761,
        "ahi_condition": "D",
        "criticality_condition": "A",
        "replace": [
            3,
            3,
            3,
            2,
            2,
            2,
            1,
            2,
            2,
            1,
            2,
            2,
            1,
            2,
            1,
            0,
            0,
            2,
            0,
            1
        ],
        "cost": [
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0,
            0
        ],
```

```
        "risk": [
            0.23,
            0.22,
            0.21,
            0.21,
            0.22,
            0.22,
            0.22,
            0.23,
            0.23,
            0.23,
            0.24,
            0.24,
            0.24,
            0.24,
            0.23,
            0.24,
            0.25,
            0.26,
            0.25,
            0.26
        ],
        "risk_condition": [
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A",
            "A"
        ],
        "failure": [
            29.73,
            31.8,
            34.18,
            36.79,
            39.58,
            42.37,
            45.17,
            48.13,
            50.99,
            53.79,
            56.65,
            59.25,
            61.73,
            64.26,
            66.43,
            68.57,
            70.86,
            73.02,
            74.72,
            76.55
        ],
        "failure_condition": [
            "B",
            "B",
            "B",
            "B",
            "B",
            "B",
            "B",
            "B",
            "C",
            "C",
            "C",
            "C",
            "C",
            "C",
            "D",
```

```
            "D",
            "D",
            "D",
            "D",
            "D"
        ],
        "mRID": "COMLK9050",
        "resourceType": 4,
        "subType": 0,
        "name": "COMLK9050",
        "description": "COMLK9050",
        "container": 1028755,
        "geometry": "POLYGON ((-83.5158514 42.6144914, -83.5158433 42.6143915, -83.5158307
42.6143225,
-83.5158292 42.6143147, -83.5158106 42.6142467, -83.5158037 42.6142282,
-83.5158007 42.6142209, -83.5157694 42.6141578, -83.5157588 42.6141411,
-83.5157546 42.6141344, -83.5157114 42.6140787, -83.5156980 42.6140643,
-83.5115315 42.6076629, -83.5114921 42.6075858, -83.5114400 42.6075167,
-83.5113768 42.6074577, -83.5112879 42.6073879, -83.5112242 42.6073455,
-83.5100559 42.6064047, -83.5098885 42.6062415, -83.5098533 42.6062103,
-83.5096501 42.6060463, -83.5095743 42.6059959, -83.5091274 42.6057322,
-83.5090696 42.6056990, -83.5090409 42.6056861, -83.5095996 42.5957430,
-83.5097822 42.5956932, -83.5098858 42.5956521, -83.5100249 42.5955778,
-83.5101108 42.5955198, -83.5102284 42.5954213, -83.5102846 42.5953661,
-83.5103314 42.5953029, -83.5103983 42.5951960, -83.5104373 42.5951195,
-83.5106861 42.5945338, -83.5107154 42.5944402, -83.5107258 42.5943427,
-83.5110018 42.5905854, -83.5110036 42.5905259, -83.5110011 42.5904875,
-83.5109939 42.5904206, -83.5109872 42.5903759, -83.5109793 42.5903425,
-83.5109741 42.5903247, -83.5109587 42.5902820, -83.5109446 42.5902510,
-83.5109359 42.5902344, -83.5109124 42.5901955, -83.5108925 42.5901678,
-83.5108809 42.5901532, -83.5108502 42.5901197, -83.5108252 42.5900964,
-83.5108111 42.5900845, -83.5107744 42.5900577, -83.5035320 42.5849618,
-83.5035115 42.5849476, -83.5034419 42.5849087, -83.5005100 42.5833538,
-83.5004183 42.5833193, -83.5003215 42.5833034, -83.5002236 42.5833066,
-83.5001281 42.5833289, -83.4938706 42.5846331, -83.4938130 42.5846364,
-83.4937733 42.5846426, -83.4937171 42.5846572, -83.4936794 42.5846711,
-83.4936272 42.5846963, -83.4882652 42.5859274, -83.4812167 42.5809290,
-83.4811741 42.5808406, -83.4811151 42.5807625, -83.4809686 42.5806299,
-83.4808891 42.5805660, -83.4807984 42.5805198, -83.4807002 42.5804928
, -83.4805986 42.5804864, -83.4802735 42.5804988, -83.4802559 42.5804998,
-83.4801765 42.5805120, -83.4801594 42.5805166, -83.4800839 42.5805441,
-83.4800678 42.5805518, -83.4799993 42.5805935, -83.4799850 42.5806042,
-83.4799258 42.5806586, -83.4799141 42.5806718, -83.4798667 42.5807367,
-83.4798576 42.5807519, -83.4798238 42.5808249, -83.4798179 42.5808416,
-83.4797989 42.5809196, -83.4797964 42.5809372, -83.4797931 42.5810175,
-83.4797941 42.5810352, -83.4802066 42.5873334, -83.4802470 42.5879931,
-83.4803356 42.5894673, -83.4803692 42.5900243, -83.4803838 42.5901182,
-83.4804160 42.5902076, -83.4806752 42.5908919, -83.4807248 42.5909763,
-83.4808700 42.5912110, -83.4809362 42.5912835, -83.4810152 42.5913418,
-83.4811038 42.5913834, -83.4811991 42.5914068, -83.4829574 42.5917220,
-83.4880125 42.5993210, -83.4880117 42.5994059, -83.4880197 42.5995165,
-83.4880397 42.5996243, -83.4878949 42.6048761, -83.4878788 42.6049634,
-83.4878744 42.6050108, -83.4878742 42.6051032, -83.4878910 42.6051939,
-83.4892581 42.6136392, -83.4892636 42.6137129, -83.4897769 42.6180190,
-83.4897949 42.6181154, -83.4898312 42.6182065, -83.4898846 42.6182886,
-83.4940480 42.6234259, -83.4941105 42.6235015, -83.4941865 42.6235635,
-83.4942730 42.6236094, -83.4943669 42.6236376, -83.4944644 42.6236470,
-83.4944897 42.6236468, -83.4953150 42.6236446, -83.5049705 42.6235270,
-83.5050669 42.6235095, -83.5051581 42.6234735, -83.5052404 42.6234205,
-83.5053109 42.6233523, -83.5053668 42.6232718, -83.5054057 42.6231818,
-83.5054264 42.6230861, -83.5117536 42.6175279, -83.5143198 42.6179113,
-83.5144139 42.6179451, -83.5145128 42.6179597, -83.5145205 42.6179602,
-83.5145206 42.6179602, -83.5148360 42.6179750, -83.5148514 42.6179756,
-83.5149448 42.6179714, -83.5150356 42.6179500, -83.5151211 42.6179119,
-83.5151979 42.6178586, -83.5152634 42.6177921, -83.5153154 42.6177143,
-83.5153520 42.6176285, -83.5153721 42.6175371, -83.5158393 42.6145908,
-83.5158514 42.6144914))"
     }]
```

aip summary-aggregate service

The aip summary-aggregate service is used to return the investment scenario summary aggregate for the project.

The aip summary-aggregate service uses the asset health and asset investment plan data as the data source.

Resource URI: `/ibm/ife/aip/api/aip-service/scenario/{id:Number}/summary`

GET /ibm/ife/aip/api/aip-service/scenario/{id:Number}/summary-aggregate

Method: GET

URL

`https://hostname/ibm/ife/aip/api/aip-service/scenario/{id:Number}/summary-aggregate`

Required properties: None.

Optional properties: Not applicable.

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "count": 6017
    }
]
```


Custom analysis model services

The custom analysis model enables you to be able to run SPSS on the user interface.

The data that is generated by the Custom analysis model is:

- ModelFolder: The SPSS streams and relevant data.
- ConfigFolder: The configurations to run different SPSS analyzes.
- History: The history for the SPSS analyzes.
- Upload: The upload zip files that go to the model folder.
- Download: The download models or configurations from the user interface.

Base URI: `/ibm/ife/customAna-service/proxyService`

model folder service

The data source used by the model folder service are local files on the host.

Resource URI: `/ibm/ife/vi/api/customAna-service/proxyService/modelFolderService`

Get modelFolderService

Method: GET

URL

`https://hostname/ibm/ife/api/customAna-service/proxyService/modelFolderService`

For example:

`GET /ibm/ife/api/customAna-service/proxyService/modelFolderService`

Required properties: None.

Optional properties:

- code

Request: Not applicable.

Status:

- Successful **200 OK**

- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "datas": [
            {
                "updateDate": 1535957156000,
                "name": "model",
                "count": 6,
                "type": "folder",
                "child": [
                    {
                        "updateDate": 1533911115000,
                        "name": "APM",
                        "count": 2,
                        "type": "file"
}]
}]
]
```


Create modelFolderService

Method: POST

URL

https://*hostname*/ibm/ife/api/customAna-service/proxyService/
modelFolderService?name=*name* &path=*path*

Required properties:

- name
- path

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

Delete modelFolderService

Method: DELETE

URL

https://<hostname>/ibm/ife/api/customAna-service/proxyService/
modelFolderService? path=<path>

Required properties:

- path

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**

- Forbidden: **403 Forbidden**

Response: Not applicable.

spss configFolder Service

The data source used by the spss config folder service are local files on the host.

Resource URI: `/ibm/ife/vi/api/customAna-service/proxyService/configFolderService`

Get configFolderService

Method: GET

URL

`https://hostname/ibm/ife/vi/api/customAna-service/proxyService/configFolderService`

For example:

`GET /ibm/ife/api/customAna-service/proxyService/configFolderService`

Required properties: None.

Optional properties:

- code

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "updateDate": 1542162995000,
        "streams": 0,
        "name": "api",
        "description": "",
        "global": [],
        "type": "configFolder",
        "child": []
},
{
        "updateDate": 1535462874000,
        "streams": 1,
        "name": "TEST_SA_T3_conf",
        "description": "",
        "global": [
            {
                "globalParameters": [
                    {
                        "parameterName": "dsname",
                        "command": "IFEDBSSL"
                    },
                    {
                        "parameterName": "dsuser",
                        "command": "db2inst1"
                    },
                    {
                        "parameterName": "dspwd",
                        "command": "pw4ibmioteu"
                    }
                ],
                "parameters": 3
            }
        ],
        "type": "configFolder",
        "child": [
            {
                "path": "TEST_SA_T3/SAsample.str",
```

```
                "updateDate": 1535462501000,
                "name": "SAsample.str",
                "streamParameters": [],
                "type": "stream",
                "parameters": 0
            }
        ]
    }
]
```


Create configFolderService

Method: POST

URL

`https://`*`hostname`*`/ibm/ife/vi/api/customAna-service/proxyService/configFolderService?name=`*`name`*

Required properties:

- name

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

Update configFolderService

Method: PUT

URL

`https://<hostname>/ibm/ife/vi/api/vi-service/viConfig`

Required properties:

- <property name="child" type="Array" />
- <property name="global" type="Array" />
- <property name="streamParameters" type="Array" />
- <property name="updateDate" type="Number" />
- <property name="name" type="String" />
- <property name="oldName" type="String" />
- <property name="parentName" type="String" />
- <property name="path" type="String" />
- <property name="streams" type="Number" />
- <property name="type" type="String" />
- <property name="description" type="String" />

Optional properties: Not applicable

Request:


```
[
    {
        "updateDate": 1542162995000,
        "streams": 0,
        "name": "api",
        "description": "",
        "global": [],
```

```
        "type": "configFolder",
        "child": []
},
{
        "updateDate": 1535462874000,
        "streams": 1,
        "name": "TEST_SA_T3_conf",
        "description": "",
        "global": [
            {
                "globalParameters": [
                    {
                        "parameterName": "dsname",
                        "command": "IFEDBSSL"
                    },
                    {
                        "parameterName": "dsuser",
                        "command": "db2inst1"
                    },
                    {
                        "parameterName": "dspwd",
                        "command": "pw4ibmioteu"
                    }
                ],
                "parameters": 3
            }
        ],
        "type": "configFolder",
        "child": [
            {
                "path": "TEST_SA_T3/SAsample.str",
                "updateDate": 1535462501000,
                "name": "SAsample.str",
                "streamParameters": [],
                "type": "stream",
                "parameters": 0
            }
        ]
    }
]
```

Status:

- Successful **201 Created**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

Delete configFolderService

Method: DELETE

URL

`https://`*`hostname`*`/ibm/ife/api/customAna-service/proxyService/configFolderService? path=`*`path`*

Required properties:

- path

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

history Service

Table 63.		
Column	**Name**	**Description**
1	BATCHNUMBER	The id of a batch.
2	STATUS	The status of the analysis.
3	CONFIGURATION	The configuration used by the analysis.
4	ANALYSISTIME	The time of the analysis.
5	DURATION	The duration of the analysis.
6	ISSCHEDULER	The analysis is the scheduler.
7	ANALYSISNAME	The analysis name.
8	ISACTIVE	The analysis is active.
9	CONTAINER	The analysis container.
10	LOGS	The log path of the analysis.

Resource URI: `/ibm/ife/vi/api/customAna-service/proxyService/historyService`

Get historyService

Method: GET

URL

`https://`*hostname*`/ibm/ife/api/customAna-service/proxyService/historyService`

For example:

`GET /ibm/ife/api/customAna-service/proxyService/historyService`

Required properties: None.

Optional properties:

- code

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response:


```
[
    {
        "duration": "0 hours 2 mins 36 seconds",
        "container": "7667",
        "analysisTime": "2018-11-12T06:16:58.780Z",
        "configuration": "ST_DGA",
        "isScheduler": 0,
        "isActive": 0,
        "logs": "logs/1542003418",
        "batchNumber": 1542003418,
        "status": 1,
        "analysisName": "analysis_1542003418"
    }
]
```

Delete historyService

Method: DELETE

URL

`https://hostname/ibm/ife/api/customAna-service/proxyService/ historyService?batchNumber=batchNumber`

Required properties:

- batchNumber

Optional properties: Not applicable.

Request: Not applicable.

Status:

- Successful **200 OK**
- Bad request: **400 Bad request**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

download Service

The data source used by the download service are local files on the host.

Resource URI: `/ibm/ife/vi/api/customAna-service/proxyService/customana-export`

Get download Service

Method: GET

URL

`https://hostname/ibm/ife/api/customAna-service/proxyService/customana-export?filePath=filePath>`

For example:

`GET/ibm/ife/api/customAna-service/proxyService/customana-export?filePath=<filePath>`

Required properties: None.

Optional properties: Not applicable

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

upload Service

The data source used by the upload service are local files on the host.

Resource URI: `/ibm/ife/vi/api/customAna-service/proxyService/customana-upload`

Get upload Service

Method: GET

URL

`https://hostname/ibm/ife/api/customAna-service/proxyService/customana-upload`

For example:

```
GET /ibm/ife/api/customAna-service/proxyService/customana-upload
```

Required properties: None

Optional properties:

- Use multiform-data to upload the file.

Request: Not applicable.

Status:

- Successful **200 OK**
- DB Error: **500 Internal Server Error**
- Forbidden: **403 Forbidden**

Response: Not applicable.

Reference services

The reference services for IBM Maximo APM for Energy & Utilities are session, syspro, i18n, and UI Framework services.

Chapter 10. Glossary

This glossary provides terms and definitions for the IBM Maximo Asset Performance Management for Energy & Utilities software and products.

The following cross-references are used in this glossary:

- *See* refers you from a nonpreferred term to the preferred term or from an abbreviation to the spelled-out form.
- *See also* refers you to a related or contrasting term.

"A" on page 293 "C" on page 293 "F" on page 293 "H" on page 293 "R" on page 293 "T" on page 293 "V" on page 293 "Z" on page 294

A

asset
Lines and equipment that deliver power in a transmission network.

C

circuit
A set of conductors that transmit high-voltage electrical energy over long distances.

corridor
The area of transmission or distribution lines that run from the circuit start to the circuit end.

corridor segment
Section of a corridor, defined according to business practice. Corridors are typically segmented into spans of land between two poles, or uniform lengths, or according characteristics of the power line or land use.

F

feeder
See circuit.

H

hazard tree
An isolated tree that has specific characteristics requiring special attention or special handling from the utility provider.

R

region
See territory.

T

territory
A logical subset of a geographical area's power grid.

V

vegetation area
Group of trees or bushes that are considered as a whole and qualified by a set of attributes.

Z

zone
Subdivision of an area of a corridor, such as the area directly below transmission lines and the areas on the borders of a corridor.

Chapter 11. Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work
must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies” and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Chapter 12. Reviewing vegetation data

You use the Vegetation Management application to review the status of vegetation and the actions that you can take to control the level of vegetation in the corridors of your region. You can use filters to refine the views to show the information that is displayed on the **Dashboard**, **Map, and List** views.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work
must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies” and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.