IBM Cloud Pak for Data Support Manual

Tables of Contents

Disclaimer

This book is work in progress and will be updated in regular time intervals.

Please do not share this book outside of IBM or outside of your company.

{{site.data.keyword.attributedefs}}

Troubleshooting Cloud Pak for Data

Welcome to the support manual for Cloud Pak for Data. This book is designed to help you troubleshooting technical issues with Cloud Pak for Data and its components. The content is designed for version 3.5.x. Major differences in version 4.0.x are highlighted within the individual sections.

This page is about 70 percent complete. Any contributions are appreciated.

How to use the book

If you experience unexpected issues when using Cloud Pak for Data, go through the following sections step by step in order to investigate the problem.

Eliminate browser issues

In many cases, problems are caused by the browser and can be resolved by one of the following steps:

- Restart your browser.
- Log off and log on to CP4D again.
- Delete all cookies or use incognito / private browsing mode.
- Try with a different supported browser.
- Upgrade your browser to the latest version.

Check for cluster and pod health

If you are experiencing a problem with a task which has been successfully working before, there may be an issue with the stability of the cluster or with an individual pod.

- Health of the cluster and its nodes
- Recent pod restarts (See reason code in when describing the pod, see log of the previous container).
    - `Exit code: 137` points to an out-of-memory situation (increase memory limits!)
- High memory or CPU usage (consider to increase limits).

Refer to this page for additional troubleshooting recommendations on the platform level.

Identify error details

Try to find more information about the error on the UI, in the browser console, or in the logs of the relevant pods.

- **UI error messages:** You may have to hover over the error icon in order to get the details on the problem.
- **Browser logs:** Open the network tab of the browser console, re-run the scenario, and check whether you see any failing REST requests, e.g. http error 404 or error 500. See the response of the specific request for more details.
- **Pod logs:** Follow the architectural overview in the corresponding sections of this support manual to see, which pods are being involved in the scenario. Browse the logs of the corresponding logs and check whether there are any suspicious messages at the time the problem has occurred.

Perform basic research

Search for error messages or symptoms in the public and IBM knowledge sources that are available to you. Recommended knowledge sources include:

- CP4D user manual
- Public CP4D community
- IBM support search
- Stack overflow / Internet search
- This support manual (use search bar at the top of the screen)
- IBM internal references (if available to you)

Start from the failing component

If you know in which component or service the problem occurs, start from there and narrow down the issue.

- CP4D platform
- Watson knowledge Catalog
- Watson Studio

Start from the table of contents

If you don't know the failing component, look for failing capability in the table of contents on the left and follow the instructions on the corresponding page. Make sure that you

- Expand the table of contents by clicking on the less-than sign (<) in the lower left corner of the page.
- Check the **Show full table of contents** check box in order to see all topics.

Request support

If you cannot find a solution, use one of the below options to request additional support:

- IBM internal references (if available to you)
- Public CP4D community
- Open support case (Also available to IBM internal users)

For non-defect issues including ideas and enhancement requests you may open a request in the *IBM Data and AI Ideas Portal*:

- External links for customers and business partners
- Internal link for IBMers

{{site.data.keyword.attributedefs}}

Pods making up CP4D and its services

This page provides an overview on the pods making up CP4D and its services.

Restriction: Currently, only the CPD platform as well as WKC and parts of WSL are reflected. Any contributions to this page are appreciated.*

Pods for data persistence

The following table provides a rough overview on how the different types of metadata are persisted by CP4D.

Type of data	Database	Pod
Assets	CouchDB / Cloudant	wdp-couchdb
Governance artifact	DB2	wdp-db2
Global search index	ElasticSearch	elasticsearch-master
IAS, MI, AD, DQ data	DB2	iis-xmetarepo
IAS search index, QS results	SOLR	solr-0
Platform metadata	Cockroach DB	zen-metastoredb

Pod reference table

The following table provides reference information on the pods making up CP4D and its services.

Pod name	Service	Scope*	Description*	Impact of restart*
iis-services	WKC	AD, DQ, IAS, MI	Legacy IIS services tier. Runs IAS, AD, QS, DQ, MI backend and UI components.	All AD, QS, and DQ jobs are cancelled. The IAS, MI, and DQ UIs are restarted. Takes > 10 min to restart.
iis-xmetarepo	WKC	AD, DQ, IAS, MI	XMeta metadata repository storing IAS, MI, and AD metadata (all legacy IGC metadata).	Requires subsequent restart of iis-services pod.
finley-ml	WKC	AD, QS	Implements term assignment machine learning capabilities.	Running AD, QS, and DQ jobs may fail.
is-en-conductor	WKC	AD, DQ, DS	Legacy IIS engine tier. Runs DataStage jobs for AD, and DQ, as well as ODF.	Running AD and DQ jobs will fail.
is-engine-compute	WKC	AD, DS	Used for parallel execution of DataStage jobs for AD and DQ (if configured).	Running AD and DQ jobs may fail.
gov-catalog-search	WKC	UG	Indexer which updates SOLR index upon receiving Kafka events	tbd.
gov-enterprise-search	WKC	UG	UI serving the GraphExplorer visualization of enterprise Search graph data	
odf-fast-analyzer	WKC	QS	Backend for quick scan. Implements a local Hadoop cluster for running data discovery algorithms.	Running QS jobs will fail.
ia-analysis	WKC	AD, QS, DQ	Backend for data quality projects. Also used for publishing QS results.	Running AD and DQ jobs may fail. DQ UI does not work.
wkc-gov-ui	WKC	WKC	UI for WKC Glossary and workflow. Used for starting AD and QS jobs?	Affects UIs until pod is back.
gov-admin-ui	WKC	UG	Management UI of legacy "IGC New". Used for customizing asset display page in IAS. Uses gov-app-config-service as backend.	Affects UIs until pod is back.
gov-app-config-service	WKC	UG	Backend of gov-admin-ui	
gov-quality-ui	WKC	AD, DQ	Data quality UI. Reads and writes to legacy Information Analyzer APIs.	Affects UIs until pod is back.
audit-trail-service	WKC	AD, QS	Used for storing AD, and QS job history and details	tbd.
solr	WKC	AD, DQ, IAS	Search index and data cache for IAS and DQ UIs. Staging area for QS results.	Affects DQ and IAS UIs until pod is back.
kafka	WKC	UG	Runs Apache Kafka. Used for AD/QS and for OMRS metadata sync.	Running AD, SQ, and DQ jobs may fail.
zookeeper	WKC	UG	Maintaining configuration information for Kafka and Solr. Used by AD/QS.	Requires subsequent restart of kafka pod.
audit-trail-service	WKC	AD, QS	Stores AD and QS job history	Do not restart while discovery runs
cassandra-0	WKC	IAS	Repository for relationship graph data	Minor impact
portal-catalog	WKC	general	Running the WKC catalog UI	
portal-common-api	WKC	general		
portal-dashboards	WKC	general		
catalog-api	WKC	general	Back-end catalog and asset service, used to store metadata for catalog, project, etc.	Minor impact

Pod name	Service	Scope*	Description*	Impact of restart*
wdp-couchdb	WKC	general	Repository for storing asset metadata for projects, catalogs, etc.	Minor impact
wdp-db2	WKC	general	Repository for DPS, wkc-glossary, wkc-workflow, lineage	Restarting will take 5-15 minutes to start the pod, while starting some of the services will be down
dataconn-engine-opdiscovery	WKC	PRF, DPS		
dataconn-engine-service	WKC	PRF, DPS	Manages dataconn-engine-spark-cluster instances	
dataconn-engine-spark-cluster	WKC	PRF, DPS	Runs data flows	
dc-main	WKC	general	Runs catalog management UIs	Minor impact
elasticsearch-master	WKC	GS	Runs elastic search engine for global search	
gov-ui-commons	WKC	UG	Common bundle for UG UIs with resources such as fonts, icons etc., no functionality as such	
gov-user-prefs-service	WKC	UG	Microservice storing user specific configuration settings in key-value form, used by the UG UIs to save some personalized settings	
igc-ui-react	WKC	IAS	UI serving Information Assets area	
portal-job-manager	WKC			
portal-main	WKC			
portal-notifications	WKC			
rabbitmq-ha	WKC	general	Used for internal messaging, a bit like Kafka in the UG stack	No impact if restart one pod at a time since this service is HA
redis-ha-haproxy	WKC	general	Proxy service to redis-ha master	
redis-ha-server	WKC	general	Cache service for WKC UI and other backend microservices	No impact if restart one pod at a time since this service is HA
shop4info-event-consumer-0	WKC	UG	Enterprise search, receives events from Kafka	
shop4info-mappers-service	WKC	UG	Enterprise search related	
shop4info-rest-0	WKC	UG	Enterprise search, provides API interface	
shop4info-scheduler	WKC	UG	Enterprise search related	
shop4info-type-registry-service	WKC	UG	Enterprise search related	
spawner-api				
wdp-connect-connection	WKC	general	Provides access to the connection and datasource assets in the CAMS repository.	none
wdp-connect-connector	WKC	general	Helper for wdp-connect-connection	None
couchdb	CP4D	lite	Used to users and other metadata, not needed starting from V3.5	
wdp-lineage	WKC	general	Backend to serve WKC Activity lineage seen in the WKC Catalog Asset view on Lineage tab	
wdp-policy-service	WKC	DPS	Backend to do policy enforcement for data protection rules	
wdp-profiling	WKC	general	Profiling tab in WKC asset browser	
wdp-profiling-messaging	WKC	general	Profiling tab in WKC asset browser	
wdp-profiling-ui	WKC	general	Profiling tab in WKC asset browser	
wdp-shaper	WKC	general	Data refinery	
wkc-glossary-service	WKC	general	Backend for WKC glossary interacts with OMAG via Kafka and uses RabbitMQ for DPS. Connects to Db2 (BGDB), XMETA (ILGDB), uses Redis	
wkc-search	WKC	GS	Global Search	
wkc-workflow-service	WKC	WF	Backend for wkc governance artifacts workflow capabilities	
jobs-api				
jobs-ui				
metadata-discovery				
…	…	…	…	…

(*) AD: Automated discovery, QS: Quick scan, IAS: Information assets, MI: Metadata import, DS: DataStage, UG: Unified governance, WKC: Watson Knowledge Catalog, ODF: Open discovery framework, PRF: Profiling, DPS: Data protection service, GS: Global search, WF: workflow

{{site.data.keyword.attributedefs}}

Cloud Pak for Data platform

Refer to the data discovery section of the user manual on how to use the Cloud Pak for Data platform.

This chapter is intended to help you with debugging technical issues with the platform. We recommend to go through the following steps starting with the first one.

1. Follow the general CP4D troubleshooting recommendations to check whether there is any issue with your browser or with the health of the CP4D platform.

2. If you run into a specific error message, search for it using the search capability at the top of the Web page.

3. Identify the problem area and continue with the corresponding section of this support manual:

    - Connections
    - Global search
    - User management
    - Homepage
    - Gathering diagnostics
    - Platform internals

{{site.data.keyword.attributedefs}}

Connections

This page provides information troubleshooting connection issues in CP4D.

This page is about 20 percent complete. Any contributions are appreciated.

General recommendations

Perform the following tasks if you are experiencing issues when creating or testing global connections.

1. Check whether the connection type is supported.
2. Make sure that connection test is successful.
   - Check logs of `wdp-connect-connection` and `wdp-connect-connection` pods for more information if it is not.
3. Eliminate network issues.
   - Logon to CP4D worker node and check with command `curl -k -v <host>:<port>` whether database host and port are accessible.
4. Check whether the data source is up and running.
5. Validate connection parameters using a different tool.
   - Ask client to use standard tooling for validating the access to the data source.

Connections for testing (IBM internal)

For IBM internal environments you may use the DataQuack service to validate access to various types of data sources. See table in README.md file for connection parameters. Supported data sources include:

- MongoDB
- Oracle
- MySQL
- MariaDB
- PostgreSQL
- Minio
- Db2
- Microsoft SQL Server
- Informix

Tasks that may be requested by the support team

The following tasks may be requested by the support team during problem determination.

Update connection in WKC catalog to use APIKEY instead of user-id and password

Details are available here (IBM internal only).

1. Obtain the bearer token.
2. Obtain connection guid and catalog id from the URL.
3. Update connection

```
curl --location --request PATCH 'https://<cluster-url>/v2/connections/<connections-guid>?catalog_id=<catalog-id>' \
--header 'Content-Type: application/json-patch+json' \
--header 'Authorization: Bearer <bearer token>' \
--data-raw '[
    {
        "op": "remove",
        "path": "/properties/username"
    },
    {
        "op": "remove",
        "path": "/properties/password"
    },
    {
        "op": "add",
        "path": "/properties/api_key",
        "value": "<api-key>"
    }
]'
```

Enable additional hive logging

Add the following lines to the Hive configuration:

```
<property>
    <name>hive.server2.logging.operation.enabled</name>
    <value>true</value>
</property
```

{{site.data.keyword.attributedefs}}

Troubleshooting global search problems

Refer to the following sections of the CP4D user manual on how to use the global search capabilities:

- Searching for assets across projects and catalogs (WKC/WSL)
- Finding and viewing governance artifacts(WKC/WSL)

This page is intended to help you with issues related to the search bar at the top of the CP4D UI and the search API. The global search capability is also used internally when browsing governance artifacts. We recommend to go through the following steps starting with the first one.

This page is about 50 percent complete. Any contributions are appreciated.

1. Eliminate browser and platform issues and perform basic research

If you experience unexpected issues when using global search, you should first check whether there is an issue with your browser or with the health of the CP4D platform or its pods. Follow the general CP4D troubleshooting recommendations in order to do so.

2. Look for specific know issues

Go through the following list of global search known issues and see whether it comprises the problem you are looking for or at least a similar one.

Problem	Error message or symptom
Global search API or UI does not return more that 10k results	`Result window is too large, from + size must be less than or equal to: [10000] but was [..]`
The information returned by the search API is not up-to-date	There is outdated information in the search result or information is missing.

Additional recommendations on how to use the search API (consider to add to user manual):

- Advanced search cookbook (IBM internal)
- How to retrieve Is business terms or data classes by category? Use simple search API, e.g. `GET /v3/search?query=categories.primary_category_name.keyword:{MyCategory} AND metadata.artifact_type:glossary_term`

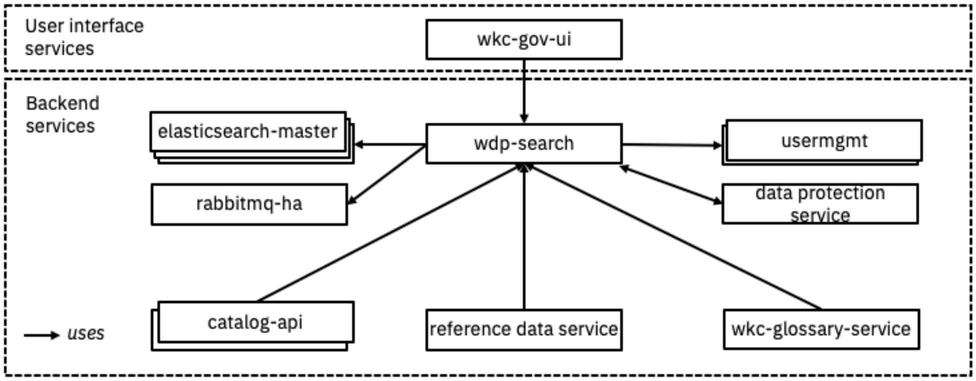
3. Understand the architecture and check for issues in related pods

This section will give you an understanding of the data flow, the data persistence, and the pods involved in the global search process. Check for relevant error messages in each of the pods at the time the problem occurred. Often, a problem can be resolved by restarting one or several of the involved pods. Make sure you understand the implications of the pod restart.

Each resulting hit will include coordinates specifying where to find the original document (this can be in the form of a URL, or an ID along with the microservice that indexed it.) It stores no data that isn't already stored in that data's home microservice.

The global search index is being updated with data from several other services, including the common asset management service (CAMS) and the glossary service. If for any reason, the data in the global search index should be out of sync, the data may be re-synced manually. This is being done by the corresponding services.

For updating the categories, business terms, data classes, reference data sets, etc. this API may be used.



See the global search Wiki (IBM internal) for additional details on the architecture and restrictions of the global search services.

4. Peform systematic debugging

If the problem persists, the following guidelines may help to isolate and resolve the issue. Start with the symptom you are seeing on you system.

{{site.data.keyword.attributedefs}}

User management

This page provides recommendations for troubleshooting CP4D user management issues.

This page is work in progress. Any contributions are appreciated.

User cannot be added because of conflicting user id

Cause by metadata being migrated from old couchdb database when migrating from 3.5.x to 3.5.y.

Reset user counter in metastoredb

```
oc exec -it zen-metastoredb-0 /bin/bash
cp -r /certs/ /tmp/
cd /tmp/ && chmod -R 0700 certs/
cd /cockroach
./cockroach sql --certs-dir=/tmp/certs/ --host=zen-metastoredb-0.zen-metastoredb
use zen;
Check for current value:
select * from user_id;
```

Navigate to the user list in the UI and sort by date. Copy the latest number and add + Then run the following command to set the counter:

```
SELECT setval('user_id', <max + 1>);
```

Remove old couchdb service before migrating from 3.5.x to 3.5.y

```
oc scale deployment couchdb --replicas=0
oc delete service couchdb-vc
```

Local user login page if SSO is enabled

https://cpd-cpd-cpd.apps.cpdarea631.ibmcloudpack.com/auth/login/zen-login.html

Debug LDAP Config

enable debug :

```
oc set env Deployment/usermgmt DEBUG="*"
oc scale deploy usermgmt --replicas=1
```

repeat the problem, collect logs of usermgmt pod. Then restore the default configuration:

```
oc set env Deployment/usermgmt DEBUG=""
oc scale deploy usermgmt --replicas=2
```

{{site.data.keyword.attributedefs}}

Troubleshooting CP4D homepage issues

This page is intended to help with debugging issues related to the CP4D home page or welcome page.

This page is work in progress. Any contributions are appreciated.

{{site.data.keyword.attributedefs}}

Gather diagnostics

This page provides information for debugging issues with gathering diagnostic information for CP4D.

This page is about 90 percent complete. Any contributions are appreciated.

General hints

Refer to the known issues below if you are running into unexpected issues when creating or downloading diagnostic files. If the problem persists,

- Do not select any services but only create the diagnostics for the platform (default). The health check section will still contain all the pod logs which will help do diagnose the problem, including the one with gathering the diagnostics itself.
- If you cannot download the diagnostic file, try to retrieve it directly from the pod as described below.

Known issues related to gathering diagnostics

Problem	Error message or symptom
Error 400 when gathering diagnostic job for Common Core Services (CCS) component	Http error 400
Error 500 when gathering diagnostic job for Common Core Services (CCS) component	Http error 500
Error 504 when grathering diagnostics	Http error 504
No feedback from UI when trying to download large diagnostic files	No response from UI.

Instructions for resolving specific issues

This section contains detailed instructions for some of the issues referenced above.

No feedback from UI when trying to download diagnostic file

After creating a diagnostic file and starting the download, there is no feedback in the UI, i.e. no Window is popping up for selecting a download location. However, in the browser console, you can see that the download is already going on. But then, when in reaches 1 GB file size, it fails.

Steps to reproduce

1. Open the browser console.
2. Create a diagnostic job that covers a large time period so the file size will be greater than 1 GB. In our scenario, components WKC and CCS were selected with 30 min time period.
3. Wait for the job to complete, then select to download the diagnostic file.
4. There won't be any feedback in the UI.
5. Follow the activity in the browser console.
6. You see that the file is actually being processed but it fails when 1 GB file size is reached.

Workaround

The generated diagnostic file is located under user-home/serviceability/collectedLogs. The filename matches the ID associated with the diagnostics collection. To download it to your local node from user-home you can use the following command. Replace `<namespace>` by the name of the OpenShift namespace in which CP4D was installed.

```
oc cp <namespace>/<nginx_pod>:/user-home/serviceability/<job_id>.zip <destination>
```

{{site.data.keyword.attributedefs}}

Troubleshooting Watson Knowledge Catalog issues

With the Watson Knowledge Catalog, you can have catalogs of curated assets that are supported by a governance framework. Refer to the data discovery section of the user manual on how to use Watson Knowledge Catalog.

This chapter is intended to help you with debugging technical issues with Watson Knowledge Catalog. We recommend to go through the following steps starting with the first one.

1. Follow the general CP4D troubleshooting recommendations to check whether there is any issue with your browser or with the health of the CP4D platform.
2. If you run into a specific error message, search for it using the search capability at the top of the Web page.
3. Identify the problem area and continue with the corresponding section of this support manual:
    - Catalogs
    - Asset profiling and asset preview
    - OMRS metadata synchronization
    - Governance artifacts
    - Data discovery
    - Data quality
    - Metadata import
    - Governance workflows
    - REST APIs

{{site.data.keyword.attributedefs}}

Troubleshooting data catalogs

This page provides recommendations for troubleshooting WKC catalogs.

This page is work in progress. Any contributions are appreciated.

{{site.data.keyword.attributedefs}}

Asset preview and asset profile

Refer to the CP4D user manual on how to use the asset preview and the asset profile capabilities.

This page describes how to debug the asset preview and the asset profile capabilities.

The page is about 20 percent complete. Any contributions are appreciated.

Relevant pods

The following pods are making up the asset preview and asset profile capabilities. See the pod reference for details on the individual pods.

- portal-catalog
- wdp-profiling
- wdp-profiling-messaging
- wdp-profiling-ui
- policy-service
- wdp-activities
- wdp-activities
- wdp-connect-service
- wdp-connect-library

Data flow

The basic data flow of the preview and profiling mechanism is:

1. Profiling services constructs a pipeline which describes source and target and process.
2. Reads source information from connection services, CAMS, etc.
3. Sends pipeline to activity service.
4. Reads progress from RabbitMQ, i.e. requests them from activity service (Do not mix up with the "Data refinery" data flow service).
5. Entire pipeline is resolved, e.g. reading details for specific connection id and send to data conn engine service.
6. Data conn engine service runs job on spark cluster, Spark uses R language, progress is logged on RabbitMQ

Known issues

- Personal connection assets results from profiling are viewable without credentials

{{site.data.keyword.attributedefs}}

Troubleshooting OMRS metadata synchronization issues

This chapter describes how to debug issues related to the internal OMRS metadata synchronization.

This page is about 80 percent complete. Any contributions are appreciated.

The OMRS sync service is responsible for internal metadata sync between Information assets catalog (based on legacy Information Governance Catalog) and WKC and vice versa. The synchronization is performed with one specific WKC catalog only, usually the *Default catalog*.

- Governance assets are synced from WKC to IGC only.
- Data assets created in WKC are synced to IGC (see restrictions below).
- Data assets created in IGC are synced to WKC (see restrictions below).

Specific restrictions apply for data assets:

- Only database tables, schemas, files, folders, and selected types of connections are synced.
- Supported connections: DB2, Oracle, Terradata.
- Synced data assets are read only on the target side, e.g. tables synced from IGC to WKC can only be modified or deleted in IGC.

If you believe that there is an issue with the OMRS sync, the first step is to validate in the following section whether the problem is indeed caused by the sync. Depending on the outcome, continue with the subsequent sections to further investigate the problem.

1. Diagnose OMRS sync issues

Problems with the OMRS sync can result in all kinds of symptoms, including the following ones:

Business Terms not assigned when running data discovery or data quality analysis

If a business term is not suggested or assigned as expected after running automated , or legacy quick scan, the business term may not have been synced from WKC to IGC internally.

Use the following steps to validate whether the business term was synced properly.

1. Logon to CP4D and navigate to the Information assets catalog.
2. Open the legacy *Information Governance Catalog (IGC)* UI which is still technically available but not officially supported in CP4D. You can reach this UI by replacing the URL path by `/ibm/iis/igc`, i.e. by navigating to URL `https://<hostname>/ibm/iis/igc`.
3. In the search bar in the middle of the screen, search for the missing business term.
4. If you cannot find a term which is available in the IGC UI even though you can see it in the business terms section of CP4D, this confirms that there is an issue with the OMRS sync.

Data classes not assigned when running data discovery or data quality analysis

If a data class is not suggested or assigned as expected after running automated discovery or legacy quick scan, the data class may not have been synced from WKC to IGC internally.

Use the following steps to validate whether the data class was synced properly:

1. Navigate to the data quality project that was selected when running automated discovery or legacy quick scan.
2. In the *Settings* tab, navigate to *Column analysis*.
3. In section *Data classification settings*, check whether the data class is available. Use the search capability if there are many data classes.
4. If you cannot find the data class, follow the steps in the previous section in order to check whether the data class is available in the You legacy *Information Governance Catalog (IGC)*. (Search for data classes rather than for business terms.)
5. If the data class is neither available in IGC nor in the data quality project, there is likely a problem with the OMRS sync.
6. If the data class is available in IGC but not in the data quality project, refer to the data discovery section for further troubleshooting.

Categories, policies, or governance rules are not synced

Follow the steps in the previous section in order to check whether the data class is available in the You legacy *Information Governance Catalog (IGC)*. Search for the corresponding artifact type rather than for business terms.

Classifications, reference data, data protection rules, data quality rules, data Rules, data definitions, or labels are not synced

As a limitation, these artifact types are not synced from WKC to IGC at all. This is not a defect but works as designed.

Data assets missing from Default catalog

Database tables and data files created in the WKC Default catalog are automatically synced to the Information assets catalog, e.g. if you add a DB2 table to the WKC Default catalog, it should also become visible in the Information assets catalog. If it does not show up, there is likely an issue with the OMRS sync. However, be aware that the sync is not supported for data assets other than database tables and data files, e.g. it is not supported for S3 files.

Data assets missing from Information assets catalog

Database tables and data files created in the Information assets catalog are automatically synced to the WKC Default catalog, e.g. if you add a DB2 table to the Information assets catalog by using Metadata import, automated discovery or legacy quick scan, it should also become visible in the WKC Default catalog. The sync is working for supported connection types only. Data assets are not being synced until they are associated to a connection.

If the data asset does not show up, there is likely an issue with the OMRS sync. However, be aware that the sync is not supported for data assets other than database tables and data files, e.g. it is not supported for S3 files.

Data assets not up-to-data in Default catalog

If you update an asset in the Information assets catalog the change will be synced to the Default catalog. This applies to the following changes only.

Attribute changes

- Name
- Description
- Column name
- Column description
- Column data type (database tables only)

Relationship changes

- Term assignment to columns
- Term assignment to assets

Data assets not up-to-data in Information assets catalog

If you update an asset in the Default catalog the change will be synced to the Information assets catalog. This applies to the following changes only.

Attribute changes

The following attributes are being synced for database tables, not for files.

- Name
- Description
- Column name
- Column description
- Column data type (database tables only)

Relationship changes

- Term assignment to columns
- Term assignment to assets
- Term to term relationships
- Term to category relationships
- Policy to policy relationships
- Policy to governance rule relationships
- Category to category relationships
- Data class to data class relationships

2. Eliminate platform issues

If you experience unexpected issues with OMRS sync, you should first check whether there is an issue health of the CP4D platform or its pods. Follow the corresponding general CP4D troubleshooting recommendations in order to do so.

3. Restart the omag pod

The `omag` pod is responsible for the actual sync process. In many cases, restarting this pod will resolve sync issues. It has typically no negative impact. You can restart the pod from within the OpenShift console or using the oc command:

```
oc scale --replicas=0 deploy omag
oc scale --replicas=1 deploy omag
```

If the problem persists you can also delete the pod. Use `oc get logs | grep omag` in order to identify the full name of the pod.

```
oc delete pod <name>
```

4. Run the WKC health check script

Run the wkc_healthchecks.sh script in order to perform a general health check of WKC and the internal OMRS sync process. The tool will return a status summary that may point to specific inconsistencies or sync issues.

Follow the recommendations in the README file of the tool. The tool may point to an issue with another component which is involved in the OMRS sync. If it does follow the instructions in the corresponding section.

Continue with the recommendations in the following table if the tool identifies an issue with the OMRS sync. Continue with the sections below if the tool does not identify any issue but you are still facing unexpected behaviour that points to an OMRS sync issue.

Problem in section	Actions to take
OMAG health	Check log of omag pod for more details. Consider to restart all OMAG-related pods.
Sync status for default catalog	Follow sections below.
Sync status of business terms	Resolve governance artifact sync issues
Sync status of data classes	Resolve governance artifact sync issues

If you still experience issues, consider to restart all related pods.

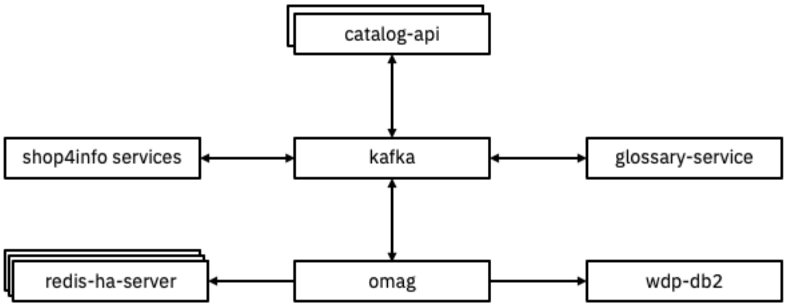
5. Look for specific know issues

Go through the following list of OMRS known issues and see whether it comprises the problem you are looking for.

- "Dangling" data assets in the default catalog
- Data asset in Default catalog does not have a connection attached to it
- Default catalog does not exist
- Health check looks OK but assets are missing from the default catalog

6. Understand basic architecture and data flow

The below figure shows the pods involved in the OMRS sync process and describes the basic data flow. The sync process is going through Apache Kafka queues. Several WKC services are connecting to the **kafka** pod in order to consume or produce relevant messages. The **shop4info** services processes all IGC GUI events and produces messages on kafka. The **omag** pods consumes from and produces to kafka to processes all data assets and glossary artifacts. It also writes to **redis-ha-server** and **wdp-db2**. The **catalog-api** pods consume and produce events related to data assets and relationships between data assets and governance artifacts (e.g. business terms). The **glossary-service** pod consumes and produces events related to governance artifacts (e.g. categories, business terms, and data classes).



Before moving to the next section, check the health and the logs of all involved pods.

7. Peform systematic debugging

If the problem persists, the following guidelines may help to isolate and resolve the issue. Start with the symptom you are seeing on your system.

Data assets not synced from WKC to IGC

Perform the following steps to find out why newly created data assets or updates to data assets are not synced from the Default catalog to IGC.

1. Identify the guid of the Default catalog. If it was manually created it may actually have a different name. After logging on to CP4D, navigate to the following URL in the browser. This command even works for such users who cannot access the catalog.

   ```
   https://<hostname>/v2/catalogs/default
   ```

   The result will look like the following. Retrieve the **guid** from the output. In the example, it is `e5333887-cae1-4ded-9a15-049140838494`.

   ```
   {
   "metadata": {
    "guid": "8752263f-f1da-4af1-944f-6758066c0112",
    "url": "/v2/catalogs/8752263f-f1da-4af1-944f-6758066c0112",
    "creator_id": "1000330999",
    "create_time": "2021-06-17T08:07:54Z"
   },
   "entity": {
    "name": "Default Catalog",
    "description": "The governed catalog where data assets are synchronized with the Information assets view.",
    "generator": "Catalog-OMRS-Synced",
    "bss_account_id": "999",
    "capacity_limit": 0,
    "is_governed": true,
    "auto_profiling": true
   },
   "href": "/v2/catalogs/8752263f-f1da-4af1-944f-6758066c0112"
   }
   ```

2. Run the following command as is to see the messages being added to the kafka omrs queue. You can either run the command on the linux command line

   ```
   oc exec -it kafka-0  -- /opt/kafka/bin/kafka-console-consumer.sh  --bootstrap-server kafka:9092  --topic open-
   metadata.repository-services.cohort.one_catalog.OMRSTopic
   ```

   or on the OpenShift consile, in the terminal window of the **kafka-0** pod:

   ```
   /opt/kafka/bin/kafka-console-consumer.sh  --bootstrap-server kafka:9092  --topic open-metadata.repository-
   services.cohort.one_catalog.OMRSTopic
   ```

   Keep the command running while proceeding with the next steps.
3. Now, create a new data asset in the default catalog or perform an update to an existing one. Navigate to the data asset and retrieve its id from the browser URL. In the example URL, the asset id is `8cbcb630-cb04-4df4-b5c7-646ed2bef1ac`.

   ```
   https://cp4d-cpd-cp4d.apps.cp4d-wkc-db2-sprk.cp.fyre.ibm.com/data/catalogs/8752263f-f1da-4af1-944f-
   6758066c0112/asset/8cbcb630-cb04-4df4-b5c7-646ed2bef1ac?context=icp4data
   ```

4. You should now see a message popping up on the kafka queue in step 3. If you don't see any change.
   - Double check the restrictions above whether the update is expected to be synced at all.
   - Restart the **kafka-0** pod and watch its logs until there are no updates any more. Then go back to step 2.
   - Try to restart the **catalog-api** pods as well.
5. Perform an OMRS health check by navigating to the following URL.

```
https://<hostname>/v2/catalogs/default/healthcheck
```

Data assets not synced from Information assets catalog to the Default catalog

Perform the following steps to find out why newly created data assets or updates to data assets are not synced from the Information assets catalog to the Default catalog.

1. Identify the guid of the Default catalog. If it was manually created it may actually have a different name. After logging on to CP4D, navigate to the following URL in the browser. This command even works for such users who cannot access the catalog.

   ```
   https://<hostname>/v2/catalogs/default
   ```

2. Check the sync status of the default catalog Replace the guid by the one retrieved in step 1.

   ```
   https://<hostname>/v2/catalogs/8752263f-f1da-4af1-944f-6758066c0112/open-metadata/sync_status
   ```

   The output provides details on the number of data assets, connection assets, and relationships being synced.
3. Run the following command as is to see the messages being added to the kafka omrs queue. You can either run the command on the linux command line

   ```
   oc exec -it kafka-0  -- /opt/kafka/bin/kafka-console-consumer.sh  --bootstrap-server kafka:9092  --topic open-metadata.repository-services.cohort.one_catalog.OMRSTopic
   ```

   or on the OpenShift consile, in the terminal window of the **kafka-0** pod:

   ```
   /opt/kafka/bin/kafka-console-consumer.sh  --bootstrap-server kafka:9092  --topic open-metadata.repository-services.cohort.one_catalog.OMRSTopic
   ```

   Keep the command running while proceeding with the next steps.
4. Optionally start tailing logs of the following pods in separate windows: **kafka**, **omag**, and **catalog-api**
5. Now, create or update an asset in the Information assets catalog. Navigate to the data asset and retrieve its id from the browser URL. In the example URL, the asset id is `b1c497ce.54bd3a08.2017ot79n.f6f8se9.sa2utv.v7volhur28kmc6i0j8qtd`.

   ```
   https://<hostname>/ibm/iis/igcui/assets/b1c497ce.54bd3a08.2017ot79n.f6f8se9.sa2utv.v7volhur28kmc6i0j8qtd
   ```

6. You should now see a message popping up on the kafka queue (step 3). When adding new assets, you should also see the numbers increasing when refreshing the sync status in step 2. If you don't see any change,

   - double check the restrictions above whether the update is expected to be synced at all.
   - check for any error messages appearing in the logs (step 3).
   - restart the **kafka-0** pod and watch its logs until there are no updates any more. Then go back to step 3.
7. Check whether the new or updated asset appears in the Default catalog. If the asset does not appear, restart the **catalog-api** pods as well.

8. If the asset itself appears in the default catalog but the actual change that you have performed on the asset is missing you may consider the following actions:

   - See restrictions above on whether the change is supposed to be synced at all.
   - Check the detailed entity information of the asset stored in the catalog by navigating to URL

     ```
     /v2/catalogs/8752263f-f1da-4af1-944f-6758066c0112/open-metadata/entity_info?guid=b1c497ce.54bd3a08.2017ot79n.f6f8se9.sa2utv.v7volhur28kmc6i0j8qtd
     ```

     where `8752263f-f1da-4af1-944f-6758066c0112` is the id of the default catalog from step 1 and `b1c497ce.54bd3a08.2017ot79n.f6f8se9.sa2utv.v7volhur28kmc6i0j8qtd` is the guid of the asset from step 5. A question mark in the result means that an attribute is missing.
   - Reshare the original IMAM import area that was created when importing the asset or when running automated discovery.
   - Rerun discovery for the asset in order to sync it from scratch.

8. Solutions for specific issues

This section covers detailed solutions for some of the known issues described above.

Restarting all OMRS related pods

If no other approach helps, you can try to restart the following pods in the given order. Wait for (1) and (2) to be fully up and running (no activity in the logs) before continuing.

1. zookeeper-0
2. kafka
3. omag
4. catalog-api (multiple pods)
5. glossary-service

Resolve governance artifact sync issues

Perform the following steps if you experience sync issues with business terms, data classes, or other governance artifacts.

1. Double check in the restrictions above whether the type of governance artifact is supposed to be synced at all.
2. Restart `glossary-service` pod and check again.
3. If you still see a gap and there are only individual artifacts missing from IGC which are available in WKC, perform a simple update on the artifacts in order to force a resync. For example, update the description field.
4. If you still see inconsistencies, resync the governance artifacts from scratch.

Creating the default catalog from scratch

If the healthcheck script does not identify any issues but there are still assets missing or additional assets in the default catalog, you may want to drop and recreate the default catalog.

Warning: Perform that step only if the default catalog just contains data assets that have been synced from the Information assets catalog originally. Do not perform the step if any data assets have been directly added to the default catalog as the metadata of those assets will be lost otherwise.

Perform the following steps to drop and recreated the default catalog.

1. Delete the default catalog by navigating to *Catalogs -> All catalogs*, select *Delete* from the three-dot-menu on the right.
2. Recreate the default catalog by following the steps described in section Default catalog does not exist below. The newly created catalog should be automatically synced with the Information assets catalog.

Removing dangling assets from Default catalog

Data assets created in the Information assets catalog, for example by running automated discovery, legacy quick scan, or by using metadata import, are automatically being synced to the Default catalog. When removing a data asset from the Information assets catalog later, it will also be removed from the Default catalog. If for some reason, some of those data assets are not being deleted, you may remove those "dangling" assets one by one using the following steps:

1. Navigate to the dangling assed in the Default catalog and retrieve the catalog id and the asset id from the URL In this example URL, the catalog id is bfdd8aeb-92d5-4818-806f-6b5202a5e8ce and the asset id is fafa4276-a58f-43b9-919b-dac4dcbfa1a8:

   ```
   https://<host>/data/catalogs/bfdd8aeb-92d5-4818-806f-6b5202a5e8ce/asset/fafa4276-a58f-43b9-919b-dac4dcbfa1a8?
   context=icp4data
   ```

2. Get service ID for CAMS

   ```
   $ oc get secrets wdp-service-id -o yaml | grep service-id-credentials | cut -d':' -f2- | sed -e 's/ //g'
   ```

   It should print out encoded string similar to `YVdOd05HUXRaR1YyT25KV1IwTk1OWFp6Ym5weWMxZz0=`

3. Decode it to get actual service id token

   ```
   $ echo YVdOd05HUXRaR1YyT25KV1IwTk1OWFp6Ym5weWMxZz0= | base64 -d
   aWNwNGQtZGV2OnJWR0NMNXZzbnpyc1g=
   ```

4. Instead of using steps 2 and 3 you can get the service id credentials from OpenShift console as follows :

   1. Logon to the OpenShift console on https:// :8443/console
   2. Navigate to Resources -> Secrets
   3. Open wdp-service-id.
   4. Click Reveal Secret.
   5. Copy value of service-id-credentials, e.g. aWNwNGQtZLMV2OstmUXA0b4R2d2M1kGS=.
5. Assign it as shell env variable

   ```
   token=aWNwNGQtZGV2OnJWR0NMNXZzbnpyc1g=
   ```

6. Use the token to call CAMS REST api to delete the assets.

   ```
   curl -k -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization:
   Basic $token" https://${hostname}/v2/assets/{asset id}?catalog_id={catalog id}
   ```

Creating the default catalog

First, check whether a synced catalog was created. It is usually being created automatically as *Default catalog* when installing CP4D. However, ot may have a different name if it was created manually. By default, only the admin user can see it. When browsing the catalogs under *Catalogs* menu option, the synced catalog has a speciall icon assigned to it. Navigate to the following URL in the browser in order to see details about the synced catalog. This command even works for such users who cannot access the catalog.

```
https://<hostname>/v2/catalogs/default
```

If the synced catalog does not exist, you can manually create it as described here.

Attaching a connection to data asset in Default catalog

If there is a data assets in the Default catalog that does not have a connection attached to it you may use the following steps in order to manually attach a connection.

1. Create a connection in WKC default catalog, e.g. an *Athena* connection.
2. Add a connection attachment to the synced connection asset which references this connection, via the CAMS `POST /v2/assets/{asset_id}/attachments` API

Example API call payload:

```
{
  "asset_type": "data_asset",
  "name": "Report2",
  "description": "Athena connection attachment for enabling data preview on Athena data asset synced from IGC",
  "connection_id": "{connection_asset_id}",
  "connection_path": "/{schema_name}/{table_name}"
}
```

Tip: Add a connected asset to the default catalog using that Athena connection, and then look at the connection attachment in that asset via GET /v2/assets/{asset_id} to double-check the values you're going to use in the API call for adding the attachment to the asset synced from IGC.

In order to call the `POST /v2/assets/{asset_id}/attachments` to modify an asset that was synced from IGC, the endpoint needs to be called using a service ID credential.

Run this command to get the credential:

```
oc get secrets wdp-service-id -o yaml | grep service-id-credentials | cut -d':' -f2- | sed -e 's/ //g' | base64 -D  && echo
```

This assumes you've done an oc login to the CPD environment.

Example:

```
$ oc get secrets wdp-service-id -o yaml | grep service-id-credentials | cut -d':' -f2- | sed -e 's/ //g' | base64 -D  && echo
aWNwNGQtZGV2OjZyMmpOc2VzZ0lqV3M=
```

Then execute a curl command to add the connection attachment, using that credential. This is an example:

```
curl -k -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Basic 
aWNwNGQtZGV2OkN3ZEZzV2R5bHN4QjM= ' -d '{    "asset_type": "data_asset",    "name": "Report2",    "description": "Athena 
connection attachment for enabling data preview on Athena data asset synced from IGC",   "connection_id": "f3559f96-980b-4014-
91fe-c23c8c3e89d8",    "connection_path": "/FDRDWH/EBAY_OC_CURATED"  }' 'https://pb-cpd.ibmcloudroks.net/v2/assets/b7a02067-
21ad-4dcd-b850-a7bddeab5a64/attachments?catalog_id=6c518b5b-a25c-45d4-8231-508c48f6c7f0'
```

{{site.data.keyword.attributedefs}}

Troubleshooting governance artifacts

This page describes how to debug issues related to governance artifacts.

This page is about 10 percent complete. Any contributions are appreciated.

Resyncing the governance artifacts

Use the following API for updating the categories, business terms, data classes, reference data sets, etc. in the **Information assets catalog** as well as in the **global search index**. Replace by the host name and port number used in the CP4D UI.

1. Get bearer token (see details here: https://cloud.ibm.com/apidocs/cloud-pak-data#get-authorization-token)

```
curl -k -X POST \
https://<hostname>/icp4d-api/v1/authorize \
-H 'cache-control: no-cache' \
-H 'content-type: application/json' \
-d '{"username":"admin","password":"<password>"}'
```

2. Start resync process for all governance artifacts Run the following API multiple times for the different artifact types by replacing with the id. Always start with `category`, then continue with `glossary_term`, `classification`, `data_class`, `reference_data`, policy, and `rule`. Do not use option `all` as it may mess up the sequence.

```
curl -k -X GET "https://<hostname>/v3/glossary_terms/admin/resync?artifact_type=<type>" -H "accept: application/json" -H
"Authorization: bearer <token>"
```

See the glossary service Wiki (IBM internal) for additional details.

Regular expression fails validation in the UI

{{site.data.keyword.attributedefs}}

Troubleshooting data discovery issues

Data discovery is about getting insight about the quality and business content of tables and files analyzed from various data connections. Refer to the data discovery section of the user manual on how to use data discovery as well as on the supported data sources.

This chapter is intended to help you with debugging data discovery issues, covering both automated discovery and quick scan. We recommend to go through the following steps starting with the first one.

This chapter is about 90 percent complete. Any contributions are appreciated.

1. Eliminate browser and platform issues and perform basic research

If you experience unexpected issues when using data discovery, you should first check whether there is an issue with your browser or with the health of the CP4D platform or its pods. Follow the general CP4D troubleshooting recommendations in order to do so.

2. Look for specific know issues

Go through the list of data discovery known issues and see whether it comprises the problem you are looking for or at least a similar one.

3. Check for issues in related pods

Refer to the data discovery architectural overview in order to get an understanding of the data flow, the data persistence, and the pods involved in the data discovery process. Check for relevant error messages in each of the pods at the time the problem occurred. Often, a problem can be resolved by restarting one or several of the involved pods. Make sure you understand the implications of the pod restart by checking out the last column of the pod reference table.

4. Peform systematic debugging

If the problem persists, the following guidelines may help to isolate and resolve the issue. Start with the symptom you are seeing on your system.

Data connection cannot be created or modified

Follow the steps in section Troubleshooting connection issues.

Connection cannot be added for data discovery

When using a global connection for data discovery for the very first time, a copy of the connection is being created in the XMeta database. Check the log of the `wkc-gov-ui` pod if there are any issues with adding the connection. Additional details may be found in the `/opt/IBM/InformationServer/ASBNode/logs/asb-agent-0.out` log file on the `is-en-conductor-0` pod.

As a workaround you may add and test the connection to XMeta manually in the Metadata import panel using the steps described in section Creating connections for use in quick scan or automated discovery jobs of the user manual.

Data discovery job cannot be submitted

Check the log of the `wkc-gov-ui` pod and the `iis-services` whether they point to any issue with starting the discovery.

Data discovery job fails

Perform the following steps in order to debug the problem:

1. Hover over the warning sign in the UI that indicates the failure. A message should pop up pointing to the detailed error message that may point to the root cause of the problem.
2. Use the ODFAdmin tool to debug the discovery job.
3. If DataStage is involved, follow the instruction on page Debugging DataStage jobs.

Data discovery job does not return the expected results

1. Validate the OMRS metadata sync status
   - For example, business terms or data classes may not be assigned properly in case of sync issues.
2. Make sure you understand the process flow.
3. Refer to best practices for automated discovery, quick scan, and business term assignment.
4. Consider to restart `solr-0` pod and run SOLR reindexing.

Data discovery results cannot be reviewed or published

1. Check pod logs, especially `iis-services`.
2. Follow progress in UI.
3. If using legacy quick scan, keep in mind that approval progress will be slow. It can take several minutes per dataset.

5. Ask for support

- Follow general guidelines for getting support.
- When opening a support case, add the following information:
  1. Error messages and logs captured above.
  2. For discovery failures: Output of ODFAdmin tool.

3. For DataStage related issues: osh script and DataStage job log.
4. CP4D diagnostics for CCS and WKC (Capture them after reproducing the issue).

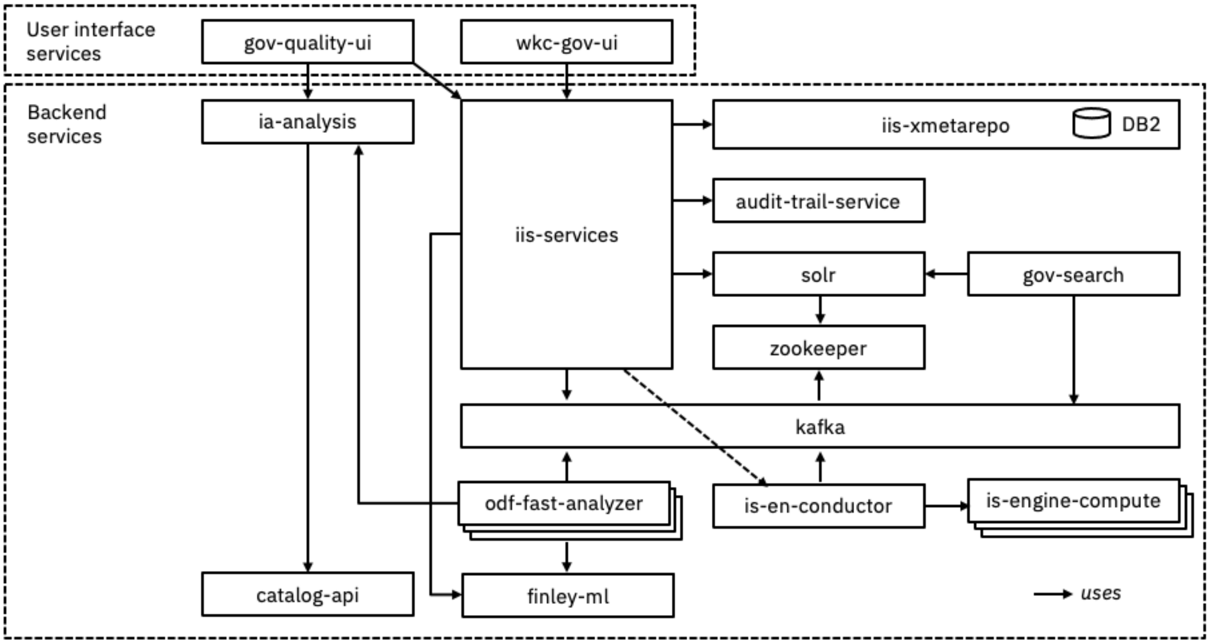
{{site.data.keyword.attributedefs}}

Data discovery architectural overview

This page provides an architectural overview of the WKC data discovery capabilities.

This page is about 90 percent complete. Any contributions are appreciated.

The following picture shows components being involved in the data discovery process. The arrows indicate which services are being used by other services.



When you navigate to *Governance -> Data discovery* in the main menu, the data discovery is opened which is implemented by the **wkc-gov-ui** pod internally. The data quality projects and the result of the quick scan analysis jobs are implemented by the **gov-quality-ui** pod. Most of the backend capabilities are provided by the **iis-services** pod. Only some capabilities related to data rule results is provided by the **ia-analysis** pod. Most of the metadata is stored in the XMeta and IADB DB2 databases running on the **iis-xmetarepo** pod. For improving the UI performance, some data from data quality projects is mirrored to Apache Solr running on the **solr** pod. The Solr index is kept up-to-date through the **gov-search** pod which is processing events from an Apache Kafka queue implemented by the **kafka** pod. The **kafka** and the **solr** pods are using configuration information from stored in Apache Zookeeper on the **zookeeper** pod.

Adding a connection for data discovery

When using a global connection for data discovery for the very first time, a copy of the connection is being created in the XMeta database. This step is being triggered by **wkc-gov-ui** pod. The copy of te connection can be browsed by navigating to *Governance -> Metadata Import* in the CP4D menu. In the *Repository Management* tab, select *Browse assets -> Data connections* on the left.

Running data discovery jobs

When you run a new or re-run and existing data discovery job, the internal Open Discovery Framework (ODF) running in the **iis-services** pod generates an "ODF request" that may consists of many sub-requests, each comprising a sequence of discovery services to be called, such as *column analysis*, *data quality analysis*, and *business term assignment*. The ODF request is put to an Apache Kafka queue from which it will be picked up by either the **odf-fast-analyzer**, the **is-en-conductor**, or again the **iis-services** pod, each being responsible for processing specific portions of the request. The **audit-trail-service** stores the history and the details of the discovery requests.

Processing automated discovery requests

Automated discovery requests are mainly implemented as DataStage jobs which are being processed by the **is-en-conductor** pod with help of the **is-engine-compute** pods. Per default, there is only one instance of the latter. When using a custom configuration for processing partitioned datasets, there may be multiple instances of the **is-engine-compute** pod, each processing one or multiple partitions of the dataset. The business term assignment is performed by the **iis-services** pod with help of the **finley-ml** pod which implements the machine learning based term assignment logic. The analysis results are written back to the Kafka queue from where they are being picked up by the **iis-services** pod and written to a data quality project within the IADB DB2 database. The technical metadata of the analyzed tables (dataset names, columns names, data types etc.) is immediately persisted to the *Information Assets* catalog implemented by the XMeta DB2 database.

Processing quick scan requests

Quick scan requests are being implemented as Hadoop jobs, which are running inside an embedded Hadoop engine within the **odf-fast-analyzer** pod. The machine learning based term assignment is again performed by the **finley-ml** pod. The results are being written to a staging area on Apache Solr (solr pod). No data is persisted to any meta data catalog until the discovery results are approved.

Reviewing and approving results

The review and approval process differs considerably between automated discovery and quick scan.

Automated Discovery

Automated discovery results are stored in data quality projects where they can be reviewed and adjusted in various ways. Finally the discovery results of individual datasets or all datasets can be published to the *Information Assets* catalog implemented by the XMeta DB2 database (`iis-xmetarepo` pod). This publication step only refers to the data discovery results while the technical metadata of all discovered tables is implicitly being published in the analysis phase (see related comment above).

Quick scan

Quick scan results are stored in a sand box on Solr. When reviewing the results, only the assigned business terms can be adjusted before publishing the results of individual datasets or of all datasets.

Next generation quick scan (NGQS): By default, the results are published to a selected WKC catalog using the REST APIs provided by the `catalog-api` pod. The technical metadata of the datasets is published along with the discovery results. The publishing process can be repeated multiple times in order to publish the results to multiple catalogs.

Legacy quick scan (hidden option): When enabled, the results are being published to a data quality project rather than to a WKC catalog. Similar as for automated discovery, the results can then be published to the Information Assets catalog. You cannot switch forth and back between NGQS and legacy quick scan but the feature needs to be enabled right away when setting up the cluster. Because of the limited scalability of the XMeta database, the publication process may be slow.

Characteristics of automated discovery vs. quick scan

While automated discovery and quick scan are pursuing similar objectives, they have different characteristics and follow different architectural approaches.

Characteristic	Quick scan	Automated discovery
Architecture	Hadoop, SOLR, lightweight connectors	DataStage, XMeta, DataStage connectors
Performance	High - even for large number of big datasets	Medium – use for small number of datasets or full analysis of individual tables
Target	Arbitrary WKC catalogs	Data quality project / Information assets
Asset import	Delayed – when publishing results	When starting discovery process
Sampling	Always operates on a sample of rows	Various sampling options or full analysis
Term assignment	Linguistic, optional lightweight ML (less data considered, English only)	Linguistic + data class based + machine learning
Connectors	Primary SQL dbs, Hive, native Hadoop	Broad set of connectors
Granularity	Schemas or folders	Schemas or individual data assets

Best Practices for using Automated Discovery and Quick Scan

There is a comprehensive document available on best practices for automated discovery and quick scan. The document may be shared with selected clients.

- Best Practices for using Automated Discovery and Quick Scan for 3.5 (IBM internal) Additional copies of the document are available here and here.
- Business term assignment best practices (publicly available)

{{site.data.keyword.attributedefs}}

Data discovery known issues

This page summarizes major known issues in the area of automated discovery and quick scan. The issues are sorted by the phase they happen in the data discovery process flow, starting with the creation of the connection to be used for data discovery.

This page is about 90 percent complete. Any contributions are appreciated.

Data connection cannot be created or modified

Problem	Error message or symptom
Error when creating PostgreSQL JDBC connection	`[CDIMR0070E] There was an error querying all tables in the public schema.`
Changes to platform-level connections aren’t propagated for discovery	The connection is not updated for data discovery
How to change the Context string in a WKC connection and use in Information Assets	The contest string is not updated.
Password change is not propagated to the copy handled by auto discovery and quick scan	The password is not updated. The discovery fails.
Cannot find suitable driver for the data source	`The connector failed to locate a suitable driver for the specified URL value.`
…	…

Connection cannot be added for data discovery

Problem	Error message or symptom
Adding a connection to data discovery fails if the connection name contains spaces	`Something unexpected happened when retrieving data.` Log message: `Keystore was tampered with, or password was incorrect`
Adding a connection to data discovery fails due to deletion of a global connection has failed	`keytool error: java.lang.Exception: Certificate not imported`
Automated data discovery fails browse connection hitting CDIMR7111E Could not clone data connection DataConnectionId	`Could not clone data connection DataConnectionId`
Unable to browse the connection in the discover page	Connections cannot be browsed after restarting is-en-conductor pod.
Connection that was made by using HDFS through the Execution Engine cannot be used with auto discovery	
Hive Kerberos connection cannot be created for quick scan	
Hive Kerberos connection cannot be created for automated discovery or data quality analysis	
…	…

Data discovery job cannot be started

Problem	Error message or symptom	Details (IBM internal)
Discovery jobs remain in status SUBMITTED forever	`The discovery service is currently not accepting new jobs. Try again later`	
Unable to resume quick scan jobs that are in a paused state	`The user does not have the permission to run Quickscan on the specified workspace.`	52276
Navigating to data discovery menu option takes user to CP4D login screen	Login page is displayed.	
ia-analysis pod does not recover from out-of-memory situation	`jajava.lang.OutOfMemoryError: GC overhead limit exceeded`	
…	…	…

Data discovery job does not complete

Problem	Error message or symptom	Details (IBM internal)
When the Solr pod stays offline for a long time, quick scan jobs are not restarted automatically	Quick scan job remains in status "analyzing" forever	
Quick scan hangs when it is analyzing a Hive table that was defined incorrectly	`ERROR hdfs_analyzer.DBMiniBatchRecordReader: Cannot read table`	32178
Discovery jobs are not resumed on restore	Discovery jobs are cancelled when the iis-services pod is restarted.	
…	…	…

Data discovery job fails

Problem	Error message or symptom	Details (IBM internal)

Problem	Error message or symptom	Details (IBM internal)
Analysis cannot be performed during automated discovery	`The assets will not be analyzed. There is an issue with the discovery service.`	
Automated discovery: Analysis step fails with log message "The JVM could not be created."	`The JVM could not be created.`	
Discovery service is reported to be unhealthy.	`Discovery service is unhealthy.`	
Automated discovery or data analysis fails on a dataset with a large VARCHAR column	`Fatal Error: Virtual data set.; output of \"pxbridge(3)\": the record is too big to fit in a block; the length requested is: 137208, the max block length is: 131072.`	
Data discovery fails when started by a Data Steward	`[CDIMR7118E] User 'XX' does not have permission to run service columnAnalysis`	19406
Discovery on a Teradata database fails	`The connector detected character data truncation for the link column C3. The length of the value is 12 and the length of the column is 6.`	
Quick scan, automated discovery, or a DataStage job on Parquet files through Hive fails on timestamp columns	`java.lang.ClassCastException: org.apache.hadoop.io.LongWritable cannot be cast to org.apache.hadoop.hive.serde2.io.TimestampWritableV2`	
Quick scan data discovery fails because of JDBC driver version mismatch	`ERROR hdfs_analyzer.DBMiniBatchRecordReader: Cannot read table`	
Automated discovery might fail when the data source contains a large amount of data	`DataStage job failed. Check the log for more details.\nDataStage job log:\nEntry 86: buffer(0),0: APT_BufferOperator: Add block to queue failed.  This means that your buffer file systems all ran out of file space, or that some other system error occurred.`	31864
Discovery jobs fail due to an issue with connecting to the Kafka service	`org.apache.kafka.common.KafkaException: Failed create new KafkaAdminClient`	
Quick scan fails on a Hive connection if the JDBC URL has an embedded space	`An error occurred when preparing request for external hadoop: 'null' (error code: 'CONNECTION_NOT_CONFIGURED')`	41204
Known issues with Hive and HDFS connections for data discovery	Various symptoms.	
Quick scan fails because of incorrect SELECT statement being sent to the data source	`SQL exception: SQL compilation error`	
Quick scan data discovery has failed for an unknown reason	`QUICKSCAN_ERROR_MESSAGE_START`	
Quick scan times out and fails	`java.io.IOException: com.ibm.iis.ia.services.exceptions.IAException: User isadmin with roles [..] and groups {} is not allowed to access workspace with ID .. to perform this action: WRITE_WORKSPACE`	43014
...	...	...

Data discovery job does not return the expected results

Problem	Error message or symptom	Details (IBM internal)
All or specific datasets are missing from the data quality project	After running automated discovery, the expected assets do not appear in the data quality project.	
Unable to overwrite term assignments when you publish quick scan results	A new business term cannot be assigned if a previous term assignment exists.	61019
Data quality dashboard reports data quality threshold chart incorrectly	benchmark status is marked as `No Threshold.`	
Data quality columns tab view might not properly display term assignments	Long business term names may be truncated and other term assignments cannot be seen.	
Project sometimes does not appear in the UI	A newly created data quality project may not be displayed in the UI.	
Settings for discovery or analysis might be lost after a pod restart or upgrade	Various symptoms.	31688
Not able to scroll all data assets in a data quality project in tile view	Not all data sets are visible in the UI.	
Incorrect connections are associated with connected data assets after automated discovery	Cannot preview or re-discovery data asset.	
Column analysis fails if system resources or the Java heap size are not sufficient		
Unable to see data quality projects in the workspace	`java.sql.SQLException: com.ibm.db2.jcc.am.SqlSyntaxErrorException: \"IAVIEWS.IAREGISTEREDTABLE\" is an undefined name.. SQLCODE=-204, SQLSTATE=42704`	
The owner of the table-assets that are synced to the default catalog is shown as an administrator instead of an asset owner		
...	...	...

Data discovery results cannot be reviewed or published

Problem	Error message or symptom	Details (IBM internal)
Data sets remain in "Loading" state and do not publish	The review status of that asset remains in the “Loading” state forever.	
Publication of quick scan results from the schema filter view does not work		
Quick scan approval errors	`com.ibm.iis.imam.importapi.service.exception.BlockingIdentityIssuesException","errorMessage`	
Loading error (Connection XXX is not published) shown when publishing assets to catlog after Quick scan job done	`Connection ... is not published`	
Approving tables in quick scan results fails when a table name contains a special character	`java.lang.IllegalArgumentException: URLDecoder: Illegal hex characters in escape (%) pattern - For input string: "1-"`	20577
Quick scan Approve assets or Publish assets operation fails		
After you publish several assets in quick scan only one of the duplicate assets is published, while other duplicates fail		42548
...	...	...

Errors browsing or deleting data discovery jobs

Problem	Error message or symptom
Quick scan jobs ran with pre-3.5 quick scan are not shown	
How to clean-up discovery jobs?	The UI does not provide any way to delete old discovery jobs from the list.
...	...

Detailed instructions for specific issues

This section contains detailed instructions for some of the issues referenced above.

Recreating the SOLR index

Apache SOLR is being used as a cache for displaying asset information in data quality projects more quickly. It is also being used for searching and browsing in the Information assets catalog. The SOLR index will be kept up-to-date automatically. However, in some rare cases, reindexing the SOLR index may be required. In order to do so,

1. Login to the Web UI of CP4D
2. Update the browser URL to point to the following API and press enter.

```
https://<hostname>/ibm/iis/common-utils/rest/v1/app/reindex?batchSize=300&solrBatchSize=200&threadCount=6
```

3. The status will be reported by the browser. Depending on the amount of metadata on your cluster, the reindexing process may take up to 30 minutes.
4. If the indexing process fails, you may reduce the batch sizes, reduce the thread count, or run the indexing process by asset type.
   - Example:

```
https://<hostname>/ibm/iis/common-utils/rest/v1/app/reindex?
batchSize=500&solrBatchSize=300&threadCount=6&assetType=Information%20Analyzer%20Project
```

   - Valid asset types include Data Connection, Database Column.

Restarting iis Web application within the pod

Restarting the `iis-services` pod may take a long time. In order to speed-up the process it is often sufficient to only restart the iis Web application inside the pod. Perform the following steps to do so:

1. Login to the `iis-services` pod.
2. Stop the iis Web application:

```
/opt/IBM/InformationServer/wlp/bin/server stop iis
```

3. Wait until the command returns.
4. Start the iis Web application again:

```
/opt/IBM/InformationServer/wlp/bin/server start iis
```

5. Browse the pod logs and wait until there are no more updates. This indicates that the pod is fully operational again.
6. If the problem persists, you should still try to restart the entire pod.

Use XMeta database as DB2 database for discovery tests

If you don't have a database available for discovery testing you may use the internal XMeta database. Connection details:

- Connection type: DB2
- Database: XMETA
- Host: is-xmetadocker
- Port: 50000
- User name: xmeta
- Password: Retrieve value of XMETA_PASSWORD env variable:
  - Logon to `iis-services` pod.
  - Run `echo $XMETA_PASSWORD`

The discovery service is currently not accepting new jobs

If you see a message `The discovery service is currently not accepting new jobs. Try again later` on the automated discovery job page, there seems to be a problem with the internal DataStage engine processing the analysis requests. Restart the is-en-conductor pod in order to resolve the issue. If the discovery jobs still don't start, follow these instructions for getting rid of old messages on the Kafka queue. Refer to additional recommendations on the ODFAdmin page.

Analysis cannot be performed during automated discovery

See detailed instructions here Example commands:

1. Logon to `is-en-conductor-0` pod.
2. Run the following command as is:

   ```
   /opt/IBM/InformationServer/ASBNode/bin/imam.sh  --username isadmin --password \$ISADMIN_PASSWORD --server is-servicesdocker --port 9446 --name iadb --action delete --type area
   ```

How to update data discovery connections

As a current limitation, if you update a global connection, the change will not be synced to the copy of the connection that is used for data discovery. The only exception is the connection password which is being synced automatically.

WARNING: After performing the below steps you will no longer be able to re-discover original automated discovery or quick scan jobs that have been using the original connection. Instead, you will have to start them as a new discovery job. However, you will still be able to re-analyze data assets in data quality projects which have been analyzed using the original connection.

The basic steps for updating a discovery connection are

1. Update the platform connection with the desired parameters.
2. Delete the original connection copy from Metadata import section.
3. Add the updated connection to discovery.
4. If step 3 fails: Delete the certificate of the original connection deleted in step 2.
5. Update the attributes section of the connection.
6. Identify the host your assets belong to.
7. Assign the new connection to the host.

Details on the individuals steps:

STEP 1: Update the platform connection with the desired parameters

Under *Data -> Platform connections*, update the original platform connection as desired.

STEP 2: Delete the original connection from Metadata Import section

1. Navigate to *Governance -> Metadata Import*
2. In the *Repository Management* tab, select *Browse assets -> Data connections* on the left.
3. Open the connection to be deleted.
4. Click on *Delete* and confirm.

STEP 3: Add the updated connection to discovery

1. Navigate to *Governance -> Data Discovery*
2. Select *New discovery job -> Automated discovery* (or quick scan)
3. Under *Connection*, select *Find or add connection*.
4. Select the updated platform connection and click *Add*.
5. There is no need to start the actual discovery.

STEP 4: If step 3 fails: Delete the certificate of the original connection deleted in step 2

If step 3 fails, you will have to manually remove the certificate of the original connection that was deleted in step 2. Then run step 3 again.

Symptom: When running step 3, the folliwing error message is written into `wkc-gov-ui` pod log:

```
keytool error: java.lang.Exception: Certificate not imported, alias <CONNECTION_ALIAS> already exists
```

Solution: Perfom the folliwing steps to manually delete the certificate:

1. Get bearer token.

   ```
   curl -k -X POST \
   https://<host>/icp4d-api/v1/authorize \
   -H 'cache-control: no-cache' \
   -H 'content-type: application/json' \
   -d '{"username":"<user>","password":"<password>"}'
   ```

2. Delete the certificate using the CONNECTION_ALIAS from the above error message.

   ```
   curl -k -X DELETE \
   https://<host>/shared-connections/v1/certificates?alias=<CONNECTION_ALIAS> \
   -H "accept: application/json" \
   -H "Authorization: bearer <token>"
   ```

   See additional details in this technote.

STEP 5: Update the attributes section of the connection

If there is a trailing semicolon in the attributes section of the connection it has to be removed manually. Otherwise, quick scan will fail with an `Invalid URL` error shown in the log of the `odf-fast-analyzer` pod. Automated discovery will still complete successfully. Perform the following steps to see whether there is a trailing semicolon and to remove it if needed.

1. Check for a trailing semicolon.

   - Browse the connection on the *Metadata import* page under *Repository management -> Browse assets -> Data connections*.
   - If the current value is `sslConnection=true;` (with semicolon), the new value should be `sslConnection=true` (without semicolon)
   - If there is no trailing semicolon, no action is needed.
2. Identify connection rid

   - Logon to CP4D web UI
   - Navigate to IGC REST explorer on `https://<host>/ibm/iis/igc-rest-explorer`
   - Run `GET /search/{type}` API from within the web browser: Enter type `data_connection` then click on *Try it out!*
   - A json document like the following will be returned that contains the for each connection, e.g. `b1c497ce.8e4c0a48.2g57oas1k.gp7rqgv.fnim4m.sbs6sh5jf5oae3r7e0usf`:

     ```
     "items": [
     {
      "short_description": "",
      "_name": "Db2-b5",
      "native_id": "",
      "_context": [],
      "_type": "data_connection",
      "_id": "b1c497ce.8e4c0a48.2g57oas1k.gp7rqgv.fnim4m.sbs6sh5jf5oae3r7e0usf",
      "_url": "https://cp4d-cpd-cp4d.apps.cp4d-wkc-wml-wsl.cp.fyre.ibm.com/ibm/iis/igc-
     rest/v1/assets/b1c497ce.8e4c0a48.2g57oas1k.gp7rqgv.fnim4m.sbs6sh5jf5oae3r7e0usf"
     },
     ```

3. Update attribute section of the connection: Replace `<host>`, `<rid>`, `<attributes>`, and `<password>` before running the command. In the above example, `<attributes>` would be set to `sslConnection=true`.

   ```
   curl -k -X PUT https://<host>/ibm/iis/imam/dcm/updateConn/<rid> \
   -H "Accept: application/json" \
   -H "Accept-Language: en-US,en;q=0.5" \
   -H "Content-Type: application/json" \
   -H "X-Requested-With: XMLHttpRequest" \
   -H "Origin: https://<host>" \
   -H "Connection: keep-alive" \
   -H "Referer: <host>/ibm/iis/imam/console/main/main.jsp?tab=import" \
   -H "Pragma: no-cache" \
   -H "Cache-Control: no-cache" \
   --data "{\"parameters\":{\"Attributes\":\"sslConnection=true\"},\"channelId\":\"CAS/JDBCConnector__1.0\"}"  -u admin:
   <password>
   ```

STEP 6: Identify which host your assets belong to

1. Navigate to *Catalog -> Information assets*.
2. Open one of the data assets that was imported through the original connection.
3. In the Governance section, check the Host of the asset (it probably looks like the jdbc connection string of the original connection - note that the host name will remain the same)

STEP 7: Assign the new connection to the host

1. Navigate to *Governance -> Metadata Import*
2. In the Import tab, click on *New import area*.
3. Provide a name and select connection type *IBM -> JDBC Connector* and click *Next*.
4. Click on the search icon right to Data connection and select the newly created connection from the list.
5. Unselect all check boxes, i.e. do not select any asset types like unselect Include tables or Include views.
6. Click *Next*,
7. Under *Host system name*, select host that you have identified in step 6.
8. Click *Next* and then *Import*.
9. A small number of objects should be added to the import area, a database, a data connection, and hosts (the selected one and probably is-en-conductor-0.en-cond as well)
10. Preview the results and share them to the repository.

After performing the steps, the host should be connected to the new connection and you should be able to re-analyze all data assets in the existing DQ projects that were using the original connection.

Selection of data discovery related tasks

This page describes some relevant data discovery related tasks.

Adding a connection with custom IBM-branded JDBC driver

Perform the following steps in order to add a custom IBM-branded JDBC driver that can be used by automated discovery, data quality, and metadata import. Be aware that a JDBC driver from a partner like *Progress DataDirect* needs to be *branded* by IBM in order to be used. For IBM internal use, the lates drivers are available here.

1. Upload the driver as a custom JDBC driver.
   - Navigate to *Platform connections -> JDBC drivers* and upload the driver jar file.
2. Check for the driver to become available in the `is-en-conductor-0` pod.
   - Logon to the `is-en-conductor-0` pod.
   - Check for the driver jar file to become available in the `/user-home/_global_/dbdrivers` folder.
3. Register the driver in the isjdbc.config file.
   - Edit the `/opt/IBM/InformationServer/Server/DSEngine/isjdbc.config` file.
   - Add the location of the driver to the `CLASSPATH` variable, e.g. by appending `/user-home/_global_/dbdrivers/ISgreenplum.jar`.
   - Add the location of the driver to the `CLASS_NAMES` variable, e.g. by appending `com.ibm.isf.jdbc.greenplum.GreenplumDriver`.
   - Example for the final file:

     ```
     CLASSPATH=/opt/IBM/InformationServer/ASBNode/lib/java/db2jcc4.jar;/opt/IBM/InformationServer/ASBNode/lib/java/db2jcc_
     license_cisuz.jar;/opt/IBM/InformationServer/ASBNode/lib/java/ISoracle.jar;/opt/IBM/InformationServer/ASBNode/lib/jav
     a/IShive.jar;/opt/IBM/InformationServer/ASBNode/lib/java/ISsqlserver.jar;/opt/IBM/InformationServer/ASBNode/lib/java/
     ISmongodb.jar;/user-home/_global_/dbdrivers/ISgreenplum.jar
     CLASS_NAMES=com.ibm.db2.jcc.DB2Driver;com.ibm.isf.jdbc.oracle.OracleDriver;com.ibm.isf.jdbc.hive.HiveDriver;com.ibm.i
     sf.jdbc.sqlserver.SQLServerDriver;com.ibm.isf.jdbc.mongodb.MongoDBDriver;com.ibm.isf.jdbc.greenplum.GreenplumDriver
     ```

4. Restart the NodesAgent:
   - Run command `/opt/IBM/InformationServer/ASBNode/bin/NodeAgents.sh restart`.
5. Create a new connection with the driver in Metadata import section as described here.
   - The driver prefix (in this case `jdbc:ibm:greenplum:`) should be recognized an the connection test should be successful.
6. Check for the connection to become available.
   - Navigate to "Governance -> Metadata Import*.
   - In the *Repository Management* tab, select *Browse assets -> Data connections* on the left.
7. The connection is now ready for use by automated discovery, data quality, and metadata import.

{{site.data.keyword.attributedefs}}

Debugging data discovery and data quality runs using the ODFAdmin command line tool

This page describes how to use the ODFAdmin command line tool for debugging data discovery and data quality runs.

This page is about 90 percent complete. Any contributions are appreciated.

ODF stands for *Open Discovery Framework* which is the scheduling component used by the data discovery and data quality capabilities of WKC. The ODFAdmin tool can be used for troubleshooting

- Automated discovery
- Quick scan
- Data quality analyses performed in data quality Projects
- Data rules and quality rules

Starting the tool

The tool is available on the `is-en-conductor-0` pod. For starting the tool, logon to the pod on the oc command line or through the terminal tab of the OpenShift console (*Workloads -> Pods -> is-en-conductor-0 -> Terminal*). Then, run the `/opt/IBM/InformationServer/ASBNode/bin/ODFAdmin.sh` command. Please note that

- The tool may need **several minutes to come** up because it may have to process a large number of Kafka messages. The more requests have been started in the last couple of days the longer it will take for the tool to come up.
- After 3 minutes the tool will come up even if it has not processed all messages. Run the `a -l` command within the tool several times until the output does not change any more. This will indicate that all messages have been processed. If the output keeps changing from run to run it **will not represent the latest status**. Before all Kafka messages have been processed, discovery jobs may be shown as running even though they have already completed.
- Always use the `q` (quit) command for exiting from the tool. **Do not use Ctrl-Z** as this will leave a background service running which will consume memory and may cause the `is-en-conductor-0` pod to run out of memory at some point in time. Run the `ps -efa | grep ODFCommandLineInterface` command in order to check whether any old instances of the tool are running and consider to kill the corresponding processes or restart the pod in order to get rid of them.

Capability overview

After starting the tool it will display basic usage information. Additional information will be shown by entering `h` or one of the commands `a`, `s`, etc.

```
/opt/IBM/InformationServer/ASBNode/bin/ODFAdmin.sh
Initializing Open Discovery Framework...
Reading configuration file /opt/IBM/InformationServer/ASBNode/bin/../conf/odf.properties
Open Discovery Framework initialized.
======= Usage =======
ODFAdmin [--user <user> --password <password>][command]
======= Available commands =======
analyses (a): Manage ODF analysis requests.
services (s): Display registered discovery services.
engine (e): Display status of ODF Engine.
discovery (d): Discovery management tool
expert (x): Run performance expert tool.
help (h): Show detailed help and example commands.
quit (q): Quit the command line processor.
odf>
```

The most important commands are `a` (analyses) for debugging analysis requests as well as `e` (engine) for debugging the health of the discovery & analysis infrastructure. The `s` (services) can be used to show a list of internal discovery services and the internal runtimes they are be running on. The `d` (discovery) command can be used to cancel or reset discovery runs. The `x` commmand is no longer supported.

Showing details of analysis requests

This section explains how to list and debug analysis requests using the `a` (analyses) command. Note that an *analysis request* can represent one of the following activities:

- Automated discovery job
- Quick scan job
- Analysis performed from within a data qualit project
- Run of a data rule or quality rule

Perform the following steps for displaying details of an analysis requests:

1. Once the tool has initialized, run the `a - l` (analyses -all) command to show all running requests and those that have been completed within the last 24 hours (See here for increasing the time limit). For each request, the status, the internal queue (S - Standard or B - Background), submission time, execution time, curren progress, service sequence, and additional details on the type of request will be displayed. Example:

```
odf> a -l
================================================================================
Processing ODF command: a -l
================================================================================
======= Recent analysis requests =======
# STATUS   Q SUBMITTED      EXECTIME PROG SEQ ORIGIN              CURRENT-SERVICE
DESCRIPTION                                                   CORRELATION-ID
REQUEST-ID
1 FINISHED S 10/13-17:14:16 00:00:50 100% 1/1 ------              "Quick Scan Parallel Service"
-----------                                                   FastAnalyzer
qs_1634145255325
```

```
2 ACTIVE   B 10/14-10:03:04 --------   0% 1/9 Automatic Discovery 
"com.ibm.infosphere.ia.odf.services.ColumnAnalysisJobService" Auto-Discovery on 12 data sets of project Automated 
discovery 1a6c9.b98b35d5.2017ovi11.b71e9pq.qp0qoi.pgnl1gs5a6vsth9tk4f3s540-1634205758716 odf-request-521985eb-1f31-490d-
bfb9-7771dce19009_1634205785591
```

2. ODF uses a hierarchical model for managing discovery requests. Each request can have sub-requests which again can have subrequests. On each level you can see the subrequests by using the `a -d <row number>` (analyses -details) command. If you run discovery on a single dataset the command will show details for that table only. If you run discovery on multiple datasets the command will show a list of sub-requests, one for each dataset. Example:

```
odf> a -d 2
================================================================================
Processing ODF command: a -d 2
================================================================================
======= Discovery service details =======
Request id:     odf-request-521985eb-1f31-490d-bfb9-7771dce19009_1634205785591
Status details: -
The request does not contain any discovery services.
The request does not contain any data sets.
======= Subrequests =======
# STATUS Q SUBMITTED      EXECTIME PROG SEQ ORIGIN              CURRENT-SERVICE   DESCRIPTION                    REQUEST-
ID
1b ACTIVE B 10/14-10:03:06 --------   0% 1/9 Automatic Discovery "Column analysis" Data set EMPLOYMENT            odf-
request-8279d262-a5fb-4dee-a92f-ca9776d55989_1634205786568
2b ACTIVE B 10/14-10:03:07 --------   0% 1/9 Automatic Discovery "Column analysis" Data set BUILDING_TYPE         odf-
request-c8d76ac2-87af-4fd0-ac04-cd5a95ebc335_1634205787568
3b ACTIVE B 10/14-10:03:08 --------   0% 1/9 Automatic Discovery "Column analysis" Data set GENDER                odf-
request-60656588-5164-4667-ab23-0f1ca9b37e7e_1634205788573
4b ACTIVE B 10/14-10:03:09 --------   0% 1/9 Automatic Discovery "Column analysis" Data set MARITAL_STATUS        odf-
request-fcfff699-5f6a-4296-b5f5-81b15eb54e5c_1634205789605
5b ACTIVE B 10/14-10:03:10 --------   0% 1/9 Automatic Discovery "Column analysis" Data set INVOICE               odf-
request-62984431-17fa-4589-8889-5ab65b5e9ca2_1634205790640
6b ACTIVE B 10/14-10:03:11 --------   0% 1/9 Automatic Discovery "Column analysis" Data set EDUCATION             odf-
request-91a37030-5b9f-45da-b0e9-90e4c8c918cf_1634205791640
7b ACTIVE B 10/14-10:03:12 --------   0% 1/9 Automatic Discovery "Column analysis" Data set ACCOUNT               odf-
request-cd5798cb-6955-471c-8f86-779e26649890_1634205792669
8b ACTIVE B 10/14-10:03:13 --------   0% 1/9 Automatic Discovery "Column analysis" Data set CST_BUILDING_PROFILES odf-
request-8ea9c7b0-46c7-4012-8232-c8906f7ed2ec_1634205793704
9b ACTIVE B 10/14-10:03:14 --------   0% 1/9 Automatic Discovery "Column analysis" Data set CUSTOMER              odf-
request-ab0d1c89-b938-432b-85f0-36bcc450ea79_1634205794709
10b ACTIVE B 10/14-10:03:15 --------   0% 1/9 Automatic Discovery "Column analysis" Data set CST_PROFILES          odf-
request-a019d9a6-1a24-438b-b058-770f91a9f04d_1634205795704
```

3. The above request consists of 10 subrequests, `1b` to `10b`. You can go one level deeper by showing the subrequests for one of the tables, e.g. for `1b`. This will show details on the execution of the individual services making up data discovery, including column analysis and different forms of business term assignment. At the bottom of the output you will see a list of related subrequests that perform the actual column analysis and data quality analysis.

```
odf> a -d 1b
================================================================================
Processing ODF command: a -d 1b
================================================================================
======= Discovery service details =======
Request id:     odf-request-8279d262-a5fb-4dee-a92f-ca9776d55989_1634205786568
Parent request: odf-request-521985eb-1f31-490d-bfb9-7771dce19009_1634205785591
Status details: -
======= Discovery service sequence =======
# SUBMITTED      STARTED  FINISHED EXECTIME SERVICE-NAME                                       RESPONSE DETAILS
1 10/14-10:03:06 10:03:06 10:04:16 00:01:09 "Column analysis"                                  OK       All sub-
requests finished.
2 10/14-10:04:16 10:04:16 10:04:18 00:00:02 "Column analysis table-based classification"       OK       Column 
analysis context-based classification finished successfully
3 10/14-10:04:18 10:04:18 10:04:18 00:00:00 "Term assignment based on data set names"          OK       Service 
com.ibm.iis.odf.services.termclassification.matching.bg.MatcherDiscoveryService created 4 annotations
4 10/14-10:04:18 10:04:18 10:04:18 00:00:00 "Term assignment based on data classes"            OK       Service 
com.ibm.iis.odf.iisext.services.cbta.ClassBasedTermAssignmentService created 1 annotations
5 10/14-10:04:18 10:04:19 10:04:19 00:00:00 "null"                                             OK       Service 
com.ibm.iis.odf.services.termassignment.finley.FinleyPredictorService created 4 annotations
6 10/14-10:04:19 10:04:19 10:04:19 00:00:00 "Term Assignment Supervisor"                       OK       OK
7 10/14-10:04:19 10:04:19 10:04:19 00:00:00 "Term assignment result writer"                    OK       Annotation 
propagation finished successfully
8 10/14-10:04:19 10:04:19 10:04:19 00:00:00 "Actionable rules application after auto term assignment" OK       Actionable 
rules are applied.
9 10/14-10:04:19 -------- -------- -------- "Data quality analysis"                            -------- -------
======= Data sets =======
# NAME       TYPE         RID
1 EMPLOYMENT TableWithTAM ec1481df.fee6c3ac.2017ot79o.pq8tve9.ekq7of.6f2314qap96u15p7a6s7b
======= Subrequests =======
# STATUS   Q SUBMITTED      EXECTIME PROG SEQ ORIGIN              CURRENT-SERVICE                       DESCRIPTION 
REQUEST-ID
1s FINISHED S 10/14-10:03:07 00:01:07 100% 2/2 Automatic Discovery "Column analysis result writer"       ----------- odf-
pc-request-b1e47c50-0a23-4789-9f1e-133e4d6febf1_1634205787639
2s QUEUED   S 10/14-10:04:22 --------   0% 1/2 Automatic Discovery "Data quality analysis DataStage job" ----------- odf-
pc-request-778bea88-0277-4d60-a157-8d12dae7ccfc_1634205862756
```

4. The above request has again two subrequests, `1s` and `2s`. You go again one level deeper by showing details on one of them, e.g. `2s`. This will show the details of the subrequest which consists of two services, the *Data quality analysis DataStage job* which is running as a DataStage job on the internal DataStage runtime and the *Data quality analysis result writer* which is persisting the results in the XMeta database. This request does not have further subrequests.

```
odf> a -d 2s
================================================================================
Processing ODF command: a -d 2s
================================================================================
======= Discovery service details =======
Request id:     odf-pc-request-778bea88-0277-4d60-a157-8d12dae7ccfc_1634205862756
Parent request: odf-request-8279d262-a5fb-4dee-a92f-ca9776d55989_1634205786568
```

```
Status details: -
======= Discovery service sequence =======
# SUBMITTED      STARTED  FINISHED EXECTIME SERVICE-NAME                          RESPONSE DETAILS
1 10/14-10:04:23 -------- -------- -------- "Data quality analysis DataStage job" -------- -------
2 --------       -------- -------- -------- "Data quality analysis result writer" -------- -------
======= Data sets =======
# NAME       TYPE         RID
1 EMPLOYMENT TableWithTAM ec1481df.fee6c3ac.2017ot79o.pq8tve9.ekq7of.6f2314qap96u15p7a6s7b
```

5. You can rerun previous command by simply typing a number key and pressing enter. `1` will repeat te last command, `2` the last but one, etc. Just pressing enter will show a command history. After a while, the subrequest has completed successfully.

~~~
odf> 1
================================================================================
Processing ODF command: a -d 2s
================================================================================
======= Discovery service details =======
Request id:     odf-pc-request-778bea88-0277-4d60-a157-8d12dae7ccfc_1634205862756
Parent request: odf-request-8279d262-a5fb-4dee-a92f-ca9776d55989_1634205786568
Status details: All discovery services ran successfully
======= Discovery service sequence =======
# SUBMITTED      STARTED  FINISHED EXECTIME SERVICE-NAME                          RESPONSE DETAILS
1 10/14-10:04:23 10:06:48 10:07:40 00:00:51 "Data quality analysis DataStage job" OK       DataStage job ran successfully.
2 10/14-10:07:40 10:07:40 10:07:54 00:00:14 "Data quality analysis result writer" OK       Post processing finished
successfully
======= Data sets =======
# NAME       TYPE         RID
1 EMPLOYMENT TableWithTAM ec1481df.fee6c3ac.2017ot79o.pq8tve9.ekq7of.6f2314qap96u15p7a6s7

```
~~~

6. If the request fails on any of the above levels, a detailed error message will be displayed. On the level of the DataStage jobs, an excerpt of the DataStage job log is displayed. Additional details on the DataStage job failure will be available on the DataStage console.

7. For debugging performance issues you can add the `-p` option to any of the commands above. This will show you exactly, how much time was spent on each of the services and even on each individual step within a service. Refer to the best practices documents for data discovery and term assignment on this page for improving the performance in each of the areas. Example output:

```
odf> a -d 1b -p
================================================================================
Processing ODF command: a -d 1b -p
================================================================================
======= Discovery service details =======
Request id:     odf-request-6a0468e3-3d32-428f-b136-48e5dd04e653_1634233692388
Parent request: odf-request-6a09394c-85ed-4350-a24d-572c8295ada4_1634233691376
Status details: All discovery services ran successfully
======= Discovery service sequence =======
# SUBMITTED      STARTED  FINISHED EXECTIME SERVICE-NAME                                              RESPONSE DETAILS
1 10/14-17:48:12 17:48:12 17:49:22 00:01:10 "Column analysis"                                         OK       All sub-
requests finished.
2 10/14-17:49:22 17:49:22 17:49:25 00:00:02 "Column analysis table-based classification"              OK       Column
analysis context-based classification finished successfully
3 10/14-17:49:25 17:49:25 17:49:25 00:00:00 "Term assignment based on data set names"                 OK       Service
com.ibm.iis.odf.services.termclassification.matching.bg.MatcherDiscoveryService created 4 annotations
4 10/14-17:49:25 17:49:25 17:49:25 00:00:00 "Term assignment based on data classes"                   OK       Service
com.ibm.iis.odf.iisext.services.cbta.ClassBasedTermAssignmentService created 1 annotations
5 10/14-17:49:26 17:49:26 17:49:26 00:00:00 "null"                                                    OK       Service
com.ibm.iis.odf.services.termassignment.finley.FinleyPredictorService created 4 annotations
6 10/14-17:49:26 17:49:26 17:49:26 00:00:00 "Term Assignment Supervisor"                              OK       OK
7 10/14-17:49:26 17:49:26 17:49:26 00:00:00 "Term assignment result writer"                           OK       Annotation
propagation finished successfully
8 10/14-17:49:26 17:49:26 17:49:26 00:00:00 "Actionable rules application after auto term assignment" OK       Actionable
rules are applied.
9 10/14-17:49:26 17:49:27 17:52:45 00:03:18 "Data quality analysis"                                   OK       All sub-
requests finished.
======= Data sets =======
# NAME       TYPE         RID
1 EMPLOYMENT TableWithTAM ec1481df.fee6c3ac.2017ot79o.pq8tve9.ekq7of.6f2314qap96u15p7a6s7b
======= Subrequests =======
# STATUS   Q SUBMITTED      EXECTIME PROG SEQ ORIGIN             CURRENT-SERVICE                       DESCRIPTION
REQUEST-ID
1s FINISHED S 10/14-17:48:13 00:01:08 100% 2/2 Automatic Discovery "Column analysis result writer"       ----------- odf-
pc-request-b4fb8420-6a7f-4d4b-8e37-7582449b81ce_1634233693287
2s FINISHED S 10/14-17:49:28 00:00:56 100% 2/2 Automatic Discovery "Data quality analysis result writer" ----------- odf-
pc-request-1310cf6c-c33a-4108-8bc0-a5dbd8793b43_1634233768825
======= Service execution times =======
SERVICE-NAME                                                          AvgExecTimeMs MinExecTimeMs MaxExecTimeMs
"Column analysis"                                                             70159         70159         70159
"Column analysis table-based classification"                                   2585          2585          2585
"Term assignment based on data set names"                                       121           121           121
"Term assignment based on data classes"                                          62            62            62
com.ibm.iis.odf.services.termassignment.finley.FinleyPredictorService           254           254           254
"Term Assignment Supervisor"                                                      7             7             7
"Term assignment result writer"                                                  11            11            11
"Actionable rules application after auto term assignment"                         2             2             2
"Data quality analysis"                                                      198233        198233        198233
"Column analysis DataStage job"                                               51868         51868         51868
"Column analysis result writer"                                               16327         16327         16327
"Data quality analysis DataStage job"                                         42185         42185         42185
"Data quality analysis result writer"                                         14029         14029         14029
======= Discovery service diagnostics table 1 =======
SERVICE-NAME                                                          SubReqPrepMs MetaDataMs JobBuilderMs PreProcMs
SplitMs RunSubReqMs
"Column analysis"                                                              433         90          330       196
701       69358
```

```
"Column analysis table-based classification"                                     -          5            -         -
-           -
"Term assignment based on data set names"                                        -        108            -         -
-           -
"Term assignment based on data classes"                                          -         52            -         -
-           -
com.ibm.iis.odf.services.termassignment.finley.FinleyPredictorService            -         56            -         -
-           -
"Term assignment result writer"                                                  -          6            -         -
-           -
"Actionable rules application after auto term assignment"                        -          -            -         -
-           -
"Data quality analysis"                                                       1651       1449          200       128
1858      196375
"Column analysis DataStage job"                                                  -          -            -         -
-           -
"Column analysis result writer"                                                  -         14            -         -
-           -
"Data quality analysis DataStage job"                                            -          -            -         -
-           -
"Data quality analysis result writer"                                            -         24            -         -
-           -
Sum over all services                                                         2084       1804          530       324
2559      265733
======= Discovery service diagnostics table 2 =======
SERVICE-NAME                                                          PrepEngineMs GetResultsMs CrColMDataMs ContClassMs
VerifyMs TermCount
"Column analysis"                                                                -            -            -           -
-         -
"Column analysis table-based classification"                                  1667            5          908           4
-         -
"Term assignment based on data set names"                                        -            -            -           -
1       150
"Term assignment based on data classes"                                          -            -            -           -
0         -
com.ibm.iis.odf.services.termassignment.finley.FinleyPredictorService            -            -            -           -
43         -
"Term assignment result writer"                                                  -            -            -           -
-         -
"Actionable rules application after auto term assignment"                        -            -            -           -
-         -
"Data quality analysis"                                                          -            -            -           -
-         -
"Column analysis DataStage job"                                                  -            -            -           -
-         -
"Column analysis result writer"                                                  -            -            -           -
-         -
"Data quality analysis DataStage job"                                            -            -            -           -
-         -
"Data quality analysis result writer"                                            -            -            -           -
-         -
Sum over all services                                                         1667            5          908           4
44       150
======= Discovery service diagnostics table 3 =======
SERVICE-NAME                                                          AssociateTermsMs   AutoRulesMs DataStageMs
PrepPosProMs KafkaReadMs RunPosProcMs
"Column analysis"                                                                    -             -           -
-           -            -
"Column analysis table-based classification"                                         -             -           -
-           -            -
"Term assignment based on data set names"                                          109             -           -
-           -            -
"Term assignment based on data classes"                                             53             -           -
-           -            -
com.ibm.iis.odf.services.termassignment.finley.FinleyPredictorService              200             -           -
-           -            -
"Term assignment result writer"                                                      -             -           -
-           -            -
"Actionable rules application after auto term assignment"                            - 1634233766958           -
-           -            -
"Data quality analysis"                                                              -             -           -
-           -            -
"Column analysis DataStage job"                                                      -             -       51861
-           -            -
"Column analysis result writer"                                                      -             -           -
13326           0            -
"Data quality analysis DataStage job"                                                -             -       42178
-           -            -
"Data quality analysis result writer"                                                -             -           -
28           0        14001
Sum over all services                                                              362 1634233766958       94039
13354           0        14001
```

Checking ODF health

The **e** (engine) command can be used for debugging the health of the ODF component which is driving the data discovery and data analysis capabilities.

Use the following steps for a systematic health check:

1. Run the **e -h** (engine -health) command to run a quick ODF health check. This command is being peformed automatically on the **iis-services** every 30 seconds or so. It will check whether a basic analysis request can be performed end-to-end including the required messaging infrastructure (Apache Kafka). Expected output is:

```
odf> e -h
================================================================================
Processing ODF command: e -h
================================================================================
ODF health status: OK
Additional messages:
Details message: All discovery services ran successfully
```

If the commands reports an error, restart the `kafka-0` pod. If that does not help, restart the `zookeeper-0` pod and then the `kafka-0` pod again. It that still doesn't help, restart the `is-en-conductor-0` pod and wait until it is fully up and running. If the problem still persists, restart the `iis-services` pod as well but be aware of the implications (see pod reference table).

2. Run the `e -l` command to check whether the internal runtimes needed for processing the actual analysis requests can be reached. The expected output is:

```
odf> e -l
================================================================================
Processing ODF command: e -l
================================================================================
Java runtime available
DataStage runtime available
FAHadoop runtime available
ParallelCompound runtime available
FAExternalHadoop runtime available
FAExternalHadoop2 runtime not available
SparkLivy runtime not available
```

The FAExternalHadoop2 and the SparkLivy are not expected to run. Perform the following steps, if any of the other runtimes is indicated as not available:

3. **Java runtime not available**: There must be a severe issue with the `iis-services` pod. If you haven't done that already, restart the pod and wait until it is fully up and running (no activity in the log). Be aware of the implications (see pod reference table).
4. **ParallelCompound runtime not available**: Same action as for Java runtime.
5. **DataStage runtime not available** There must be a severe issue with the `is-en-conductor-0` pod which is performing the automated discovery column and data quality analysis. If you haven't done that already, restart that pod and wait until it is fully up and running (no activity in the log).
6. **FAHadoop runtime not available**: There must be a severe issue with the `odf-fast-analyzer` pod which is performing the quick scan analysis. If you haven't done that already, restart that pod.
7. **FAExternalHadoop runtime available**: Same action as for FAHadoop runtime.

If the problem persists, check the logs of the pods that have been restarted.

Monitor ODF usage

Run the `e -m` (engine -memory) command to see information about memory usage and queue allocation. Example:

```
odf> e -m
================================================================================
Processing ODF command: e -m
================================================================================
======= Memory usage overview (JSON object size) =======
AREA                           SIZE(CHARACTERS) COMMENT
Analysis request tracker store          1353672 Use 'analyses' command with '-internals' option for details
Annotation store                              0 See details below
The annotation store does not contain any annotations
======= Request overview =======
TYPE                                      COUNT
Overall request count                       656
Running root requests                        10
Running root requests on standard queue       0
Running root requests on background queue    10
Purged requests                               0
Limit (ignoring purged)                   40000
```

The overall request count usually doesn't matter unless it grows closed to the limit. In that case, some of the requests will get purged from memory. *Running root requests* shows how many requests are running on the *standard* queue and the *background* queue. The following limits apply

Queue	Limit	Scope
Standard	10	Quick scan, analysis in DQ projects, data rules
Background	10	Automated discovery

After an analysis request is submitted, it remains in status SUBMITTED until there is free slot available on the corresponding queue. In the above example, a large automated discovery request is running and all slots of the background queue are in use.

There are additional internal queues for the Java, DataStage, ParallelCompound, FAHadoop, and FAExternalHadoop runtimes. The default limits can be changed but there is usually no need to as the defaults have turned out to work best in most environments.

Resolving specific problem situations

This section discusses some specific problem situations and expains how to resolve them.

Debugging performance issues

Look at the performance metrics discussed in this section above.

Analysis requests fails wit a DataStage related issue

Additional details on the DataStage job failure will be available on the DataStage console.

Analysis requests remain in status SUBMITTED forever.

As long as there are ACTIVE request on the same queue, new requests are expected to remain in status SUBMITTED. However, if there are no active requests on the corresponding queue, newly submitted requests should switch from SUBMITTED to ACTIVE state after a couple of seconds. If that does not happen, you should check the

ODF health as described above. If the `e -m` command shows running requests even though there are no ACTIVE requests shown in the `a -l` output, there may be an issue with the internal counters. Consider to restart the `is-en-conductor` and the `iis-services` pods in order to resolve the issue. Be aware of the implications (see pod reference table).

Analysis requests are not making any progress for several hours

Check if any of the subrequests are remaining in status SUBMITTED even though no other requests are running. Run the `a -l -s -f STATUS=ACTIVE` command which will show all active requests including all the subrequests. If there are no active subrequests, there may be a problem with the internal queue counters. Consider to restart the `is-en-conductor` and the `iis-services` pods in order to resolve the issue. Be aware of the implications (see pod reference table).

If the problem does not go away, check the `/opt/IBM/InformationServer/ASBNode/logs/odfengine.log.0` on the `is-en-conductor-0` pod. If you see messages like the following one, the internal Kafka queues may still be busy with processing outdated requests. This will automatically resolve after some hours.

```
WARNING: Request tracker for request 'odf-pc-request-3f05fbd2-26a0-4b1e-b68c-d07b0da6117e_1527400836580' did not arrive within timeout period of '25,000' milliseconds for service queue 'com.ibm.infosphere.ia.CADataStageJobService'.
```

If the problem persists or if you don't want to wait you may perform the following steps to speed-up the processing of the messages:

1. Logon to terminal of is-en-conductor-0 pod
2. Edit file `/opt/IBM/InformationServer/ASBNode/conf/odf.properties`
3. Add a new line to the end of the file:

   ```
   com.ibm.iis.odf.waitfortracker.msecs=10
   ```

4. Save the file
5. Restart the ODF engine within the pod:

   ```
   service ODFEngine stop
   service ODFEngine start
   ```

6. Start a small discovery request and wait for completion.
7. Go back to step 2, remove the line from the `odf.properties` file again and continue with steps 4 and 5.

Performing specific actions

This section describes specific actions that may need to be requested from the support team.

Performing ODF configuration changes

The configuration options of the ODF engine are stored in file `/opt/IBM/InformationServer/ASBNode/conf/odf.properties` on the `is-en-conductor-0` pod. After performing a configuration change, a restart of the ODF engine is required. This can either be done by restarting the pod or by just restarting the ODF engine service ODFEngine stop itself which is much faster:

1. Logon to the `is-en-conductor-0` pod.
2. Stop the ODF engine using command:

   ```
   service ODFEngine stop
   ```

3. Start the ODF engine again using command:

   ```
   service ODFEngine start
   ```

4. Tail the ODF log to see the engine coming up:

   ```
   tail -f /opt/IBM/InformationServer/ASBNode/logs/odfengine.log.0
   ```

Clean up odf kafka queues without impacting running pods

Sometimes, the only chance to resolve an ODF issue is to cleanup the ODF Kafka queues. This will stop all running analysis requests and will remove the complete history from the ODFAdmin tool. The discovery job history displayed in the UI will not be affected. Perform these steos only if explicitly requested by the support team.

Perform the following steps to clean up all Kafka messages from the ODF queues:

1. Logon to the `kafka-0` pod.
2. Start with `odf-status-topic`.
3. Change retention period to 1 second:

   ```
   /opt/kafka/bin/kafka-topics.sh --zookeeper zookeeper:2181/kafka --alter --topic odf-status-topic --config retention.ms=1000
   ```

4. Show messages on topic:

   ```
   /opt/kafka/bin/kafka-console-consumer.sh --topic odf-status-topic --bootstrap-server kafka:9093 --from-beginning
   ```

5. Repeat until almost no messages left
6. Change retention period back to default value:

   ```
   /opt/kafka/bin/kafka-topics.sh --zookeeper zookeeper:2181/kafka --alter --topic odf-status-topic --config retention.ms=604800000
   ```

7. Repeat steps 3 to 6 for the other topics:
   - odf-admin-topic
   - odf-runtime-AnalysisRequest
   - odf-runtime-BackgroundAnalysisRequest
   - odf-runtime-DataStage
   - odf-runtime-Java
   - odf-runtime-ParallelCompound

Configure ODFAdmin to show older analysis requests

You can change the configuration of the ODFAdmin tool to show analysis requests which are older than 24 hours. Be aware that if there are many old requests, this may drastically slow down the startup of the ODFAdmin tool as well as of the ODF Engine running on the `is-en-conductor-0` pod. Perform the following steps to adjust the configuration:

1. Logon to the `is-en-conductor-0` pod.
2. Edit the `/opt/IBM/InformationServer/ASBNode/conf/odf.properties` configuration file.
3. Add the following line to the file where the number is the number of seconds that should be covered back in time. The value `172800` will show requests of the last two days. The maximum value is seven days.

   ```
   com.ibm.iis.odf.kafka.skipmessages.older.than.secs=172800
   ```

4. Restart the `/opt/IBM/InformationServer/ASBNode/bin/ODFAdmin.sh` tool.

{{site.data.keyword.attributedefs}}

Debugging DataStage related issues

This page describes how to debug DataStage related issues in Watson Knowledge Catalog.

This page is about 80 percent complete. Any contributions are appreciated.

DataStage is being used internally by the following WKC features:

- Automated Discovery
- Data quality analysis in data quality projects
- Data rules and quality rules

Which pods to check

From an architectural perspective, DataStage is represented by the **is-en-conductor-0** pod and by one or multiple **is-en-compute** pods. In the default configuration, each DataStage job is running as a separate process on the **is-en-compute** pod.

Perform a DataStage health check

In order to run a quick health check you can validate whether simple DataStage job is completing successful.

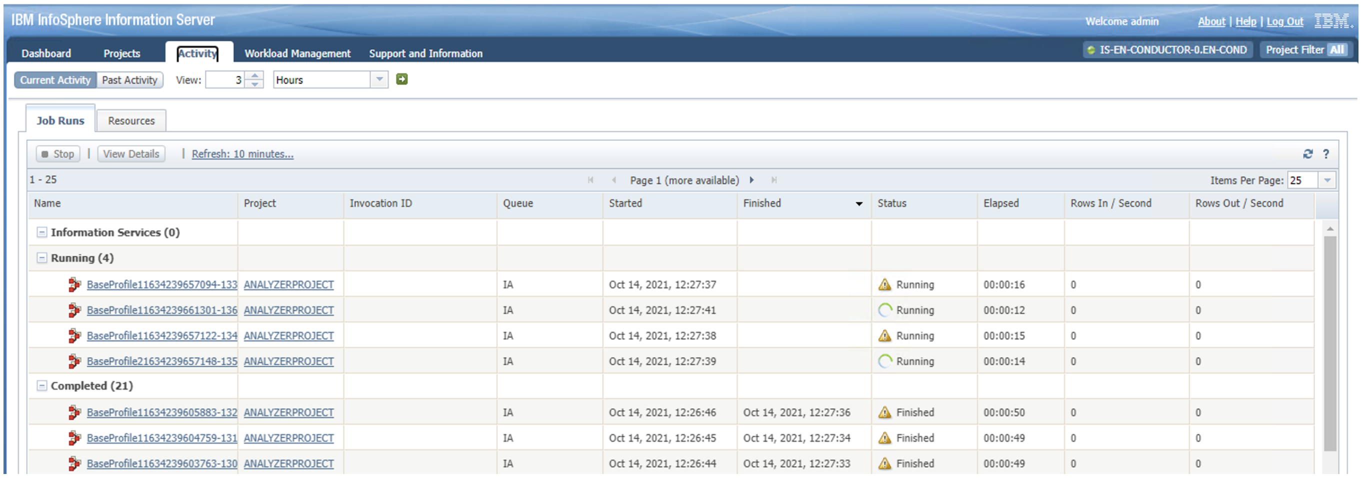
1. Logon to CP4D Web UI.
2. Replace browser URL path by `/ibm/iis/ia/api/configuration/iadb/validate?runTestJob=true` and press enter. Example URL:

   `https://<hostname>/ibm/iis/ia/api/configuration/iadb/validate?runTestJob=true`

3. Successful completion is indicated by message `Validation successful..`.
4. If the validation is unsuccessful, consider to restart the `is-en-conductor-0` pod.

Monitoring DataStage jobs

DataStage jobs can be monitored using the legacy DataStage console which is technically still available in CP4D but which is not officially supported. It is running on the **iis-services** pod. In order to open the DataStage console, logon to the CP4D web UI, then replace the browser URL path by **/ibm/iis/ds/console** and press enter. The complete URL would be **https://<hostname>/ibm/iis/ds/console**.



Name	Project	Invocation ID	Queue	Started	Finished	Status	Elapsed	Rows In / Second	Rows Out / Second
⊟ Information Services (0)									
⊟ Running (4)									
BaseProfile11634239657094-133	ANALYZERPROJECT		IA	Oct 14, 2021, 12:27:37		⚠ Running	00:00:16	0	0
BaseProfile11634239661301-136	ANALYZERPROJECT		IA	Oct 14, 2021, 12:27:41		○ Running	00:00:12	0	0
BaseProfile11634239657122-134	ANALYZERPROJECT		IA	Oct 14, 2021, 12:27:38		⚠ Running	00:00:15	0	0
BaseProfile21634239657148-135	ANALYZERPROJECT		IA	Oct 14, 2021, 12:27:39		○ Running	00:00:14	0	0
⊟ Completed (21)									
BaseProfile11634239605883-132	ANALYZERPROJECT		IA	Oct 14, 2021, 12:26:46	Oct 14, 2021, 12:27:36	⚠ Finished	00:00:50	0	0
BaseProfile11634239604759-131	ANALYZERPROJECT		IA	Oct 14, 2021, 12:26:45	Oct 14, 2021, 12:27:34	⚠ Finished	00:00:49	0	0
BaseProfile11634239603763-130	ANALYZERPROJECT		IA	Oct 14, 2021, 12:26:44	Oct 14, 2021, 12:27:33	⚠ Finished	00:00:49	0	0

For browsing the job log,

1. Navigate to the *Activity* tab in order to see the running DataStage jobs.
2. Select the row of the job to be displayed so it will be highlighted (do not click on the job name).
3. Click on the *View Details* button in order to see the job details.
4. The job log is available in the *Log Messages* tab.
5. Check the *Full Messages* checkbox in order to see all messages. Note that the full messages will only be available if you configure the data quality project to *Retain the analysis jobs and the job logs*. Refer to the following section for details.
6. Click on a log entry to see more details.

Retaining OSH scripts and logs after job completion

By default, the DataStage job scripts and their logs related to WKC data discovery and data analysis are automatically deleted after job completion. In order to retain them, a configuration option needs to be adjusted in the data quality projects.

1. Open data quality project used for data discovery or data analysis.
2. In the *Settings* tab, navigate to the *Advanced options* section.
3. Select the *Retain the analysis jobs and the job logs* check box and save the settings. This makes sure that OSH script of the generated DataStage jobs will be kept.
4. Run the scenatios that you want to capture the scripts and logs for.
5. Uncheck the above option again in order to avoid the folders from filling up.

Save the complete log of a job

Perform the following steps to download a complete DataStage job log.

1. Configure the data quality project to *Retain the analysis jobs and the job logs* (see separate section above).
2. Identify the job name from the DataStage console, e.g. `BaseProfile11616517063936-26` (see separate section above).
3. Logon to the `is-en-conductor-0` pod.
4. Run the following command to set the DataStage environment:

   ```
   . /opt/IBM/InformationServer/Server/DSEngine/dsenv
   ```

5. Retrieve the list of available job logs:

   ```
   /opt/IBM/InformationServer/Server/DSEngine/bin/dsjob -ljobs ANALYZERPROJECT
   ```

6. Retrieve the log of the relevant job, e.g.

   ```
   /opt/IBM/InformationServer/Server/DSEngine/bin/dsjob -logdetail ANALYZERPROJECT BaseProfile11616517063936-26
   ```

7. Redirect the the job log to a file, e.g.

   ```
   /opt/IBM/InformationServer/Server/DSEngine/bin/dsjob -logdetail ANALYZERPROJECT BaseProfile11616517063936-26 >
   /tmp/BaseProfile11616517063936-26.log
   ```

Capture the OSH script of a job

The so-called OSH script is the textual representation of a DataStage job. For problem determination, this script will be needed by the support team. Perform the following steps in order to capture the script:

1. Open data quality project used for the problematic scenario and select to *Retain the analysis jobs and the job logs* (see separate section above).
2. Rerun the problematic scenario.
3. Logon to the `is-en-conductor-0` pod.
4. Change to the following folder:

   ```
   cd /opt/IBM/InformationServer/Server/Projects/ANALYZERPROJECT
   ```

5. List folders with name `RT_SC<number>` ordered by creation time:

   ```
   ls -rtl | grep 'RT_SC[0-9]'
   ```

6. Capture the files `OshScript.osh` from the two newest folders, e.g. `RT_SC33` and `RT_SC33`. There will be separate folders for *column analysis* and *data quality analysis*.
7. Uncheck option *Retain the analysis jobs and the job logs* in the data quality project settings again.

Enable debug traces

If requested by the support team, follow the steps below in order to enable the debug logging:

1. Logon to the `is-en-conductor-0` pod.
2. Edit the `/opt/IBM/InformationServer/ASBNode/conf/asbagent-logging.properties` file.
3. Add the following lines to the end of the file:

   ```
   com.ibm.iis.cas.level=ALL
   com.ibm.iis.isf.agent.log.level=ALL
   ```

4. Set the following environment variable within the pod:

   ```
   # export CC_MSG_LEVEL=1
   ```

5. Run the following command to restart the NodesAgent:

   ```
   /opt/IBM/InformationServer/ASBNode/bin/NodeAgents.sh restart
   ```

6. Reproduce the issue
7. Capture all logs from the following folder within the pod:

   ```
   /opt/IBM/InformationServer/ASBNode/logs
   ```

8. Remove the two lines from the properties file updated in steps 2 and 3.
9. Restart the pod in order to disable the logging.

Set environment variables for automated discovery

The dsadmin command can be used for creating a new environment variable for DataStage jobs. This can be helpful for generating additional debugging information when running a job. Relevant environment variables and their values include:

- OSH_ECHO=1
- APT_DUMP_SCORE=1
- APT_STARTUP_STATUS=1
- APT_SHOW_COMPONENT_CALLS=1
- APT_NO_PM_SIGNAL_HANDLERS=1
- APT_PLAYERS_REPORT_IN=1
- APT_RECORD_COUNTS=1
- APT_DEBUG_DYNAMIC_POD=1
- APT_PM_SHOW_PIDS=1

In order to set a variable for a specific project:

1. Logon to the `is-en-conductor-0` pod.
2. Run the following command to set the DataStage environment:

   ```
   . /opt/IBM/InformationServer/Server/DSEngine/dsenv
   ```

3. Run the dsadmin command to set the environment variable. Project name would always be ANALYZERPROJECT as this is the default DataStage project for WKC. Example:

   ```
   /opt/IBM/InformationServer/Server/DSEngine/bin/dsadmin -envset OSH_ECHO -value "1" ANALYZERPROJECT
   ```

4. If the variable does not yet exist you can add a new one. Example:

   ```
   /opt/IBM/InformationServer/Server/DSEngine/bin/dsadmin -envadd OSH_ECHO -type STRING -prompt "OSH_ECHO" -value "1" 
   ANALYZERPROJECT
   ```

5. Run the command without parameters to get help information.

Remove old dumps from is-enconductor pod

You may remove old dumps from the is-enconductor pod in order to save disk space. Perform the following steps in order to do so:

1. Logon to the `is-en-conductor-0` pod.
2. Change too the ANALYZERPROJECT folder:

   ```
   cd /opt/IBM/InformationServer/Server/Projects/ANALYZERPROJECT
   ```

3. Run the following commands to remove the dumps: `

   ```
   rm -rf core*.*
   rm -rf javacore*.*
   rm -rf *dump*.*
   rm -rf Snap*.*
   rm -rf verbosegc*.*
   ```

{{site.data.keyword.attributedefs}}

Data quality

This page describes how to debug DataStage related to data quality analysis in Watson Knowledge Catalog.

This page is about 70 percent complete. Any contributions are appreciated.

Overview

From a technical perspective, there is a large overlap between data quality and data discovery. When you start an analysis process in a data quality project it is processed in the exact same way as for automated discovery, i.e. a request is started in the internal *Open Discovery Framework* (ODF) component and the actual analysis is being performed through the internal DataStage engine. Refer to the data discovery architecture for details.

Relationship analyses, data rules, and quality rules are also executed through ODF and DataStage. Therefore, the same debugging recommendations apply as for automated discovery jobs.

Troubleshooting recommendations

The basic troubleshooting process is the exact same as for data discovery. For example, if a data analysis, relationship analysis, data rule, or quality rule fails, refer to section Data discovery job fails..

Known issues

The same known issues apply for data quality analysis as for automated discovery. Refer to page data discovery known issues. There are some specific issues that only apply to data quality. They are listed in the following table.

Problem	Error message or symptom
Datasets are missing from the data quality project	The project does not contain any datasets or datasets are missing.
Data Stewards can't create automation rules	`ERROR - The principal 'XX' doesn't have enough privileges to access the service method User not authorized to:create ActionableRule`
Virtual tables are not supported for BigQuery connections	Limitation.
On Firefox, no details are shown for assets that are affected by an automation rule	`No details to display`
When you add data files to the data quality project, the Tree view doesn't show data files	Tree view does not work.
The Run analysis button cannot be found	Cannot find button.
Analysis request in data quality project remains in status "In Queue" or "In Progress" forever.	Analysis is queued or runs forever.

Detailed instructions for specific issues

This section contains detailed instructions for some of the issues referenced above.

Reset status of analysis request in data quality projects

If the status of a dataset in a data quality project remains "In Queue" or "In Progress" forever but there are no corresponding requests visible in the ODFAdmin tool, there may be a mismatch between the status displayed in the UI and the status of the actual requests. You may use the following steps in order to reset the request status of all in progress and queued jobs to "Error".

1. Logon to the `iis-services` pod
2. Replace the `<project-name>` in the following command and run it. There is no need to replace the user name or the password, as `$ISADMIN_PASSWORD` points to an environment variable.

   ```
   /opt/IBM/InformationServer/ASBServer/bin/IAAdmin.sh -user isadmin -password $ISADMIN_PASSWORD -projectName -resetAnalysisStatus -projectName <project-name> -all
   ```

   Example

   ```
   /opt/IBM/InformationServer/ASBServer/bin/IAAdmin.sh -user isadmin -password $ISADMIN_PASSWORD -projectName -resetAnalysisStatus -projectName UGDefaultWorkspace -all
   ```

{{site.data.keyword.attributedefs}}

Troubleshooting metadata import

This page describes how to debug issues related to metadata import.

This page is work in progress. Any contributions are appreciated.

{{site.data.keyword.attributedefs}}

Troubleshooting governance workflows

This page describes how to debug issues related to governance workflows.

This page is work in progress. Any contributions are appreciated.

{{site.data.keyword.attributedefs}}

Watson Knowledge Catalog APIs

This page provides recommendations for troubleshooting issues related to the use of Watson Knowledge Catalog APIs.

This page is about 60 percent complete. Any contributions are appreciated.

Watson data API

The Watson Data API is supporting the strategic stack of Watson Knowledge Catalog. The API reference is available here. For some areas of the Watson Data API, an API reference is integrated into the product. You can open it by logging in to CP4D in your browser and replacing the URL path accordingly.

- Catalog API reference: `https://<hostname>/v2/cams/explorer/`
- Glossary API reference: `https://<hostname>/v3/glossary_terms/api`

Update a tag on a column of a dataset in a catalog

1. Get Bearer token:

   ```
   curl -k -X GET https:/<hostname>/v1/preauth/validateAuth -H "cache-control: no-cache" -H "content-type: application/json"
   -u admin:password
   ```

2. Add tag to column COLUMN1 of table with id `46149591-1473-408a-8c32-c59906bc74e9` in catalog with id `4754c88f-6f26-44a5-a6e6-849d1216dc6c`:

   ```
   curl -X PATCH --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization:
   Bearer <token>' --header 'X-OpenID-Connect-ID-Token: Bearer ' -d '\
   [{ \
   "op": "add", \
   "path": "/COLUMN1/column_tags", \
   "value": [ \
     "MyTag" \
   ] \
   }]' 'https://<hostname>/v2/assets/46149591-1473-408a-8c32-c59906bc74e9/attributes/column_info?catalog_id=4754c88f-6f26-
   44a5-a6e6-849d1216dc6c'
   ```

   See http://jsonpatch.com/ for details on the request body.

Example: Get a list of all assets in a catalog

This is an example for retrieving a list of all data assets from a WKC catalog or details about a specific data asset. Use the following procedure:

1. Get bearer token

   ```
   curl -k -X POST \
   https://<hostname>/icp4d-api/v1/authorize \
   -H 'cache-control: no-cache' \
   -H 'content-type: application/json' \
   -d '{"username":"admin","password":"password"}'
   ```

   x
2. Get list of catalogs

   ```
   $ curl -k -X GET -H "Authorization: Bearer <token>" -H "cache-control: no-cache" "https://<hostname>/v2/catalogs"
   ```

3. Next step is to run a search on the catalog in order to retrieve the desired assets of a certain type. For data_assets that would be

   ```
   $ curl -k -X POST -H 'content-type: application/json' -H "Authorization: Bearer <token>" -H "cache-control: no-cache"
   "https://<hostname>/v2/asset_types/data_asset/search?catalog_id=a6a69cf6-6493-4c02-92ef-bc7e0a13ef63" -d
   '{"query":"customer"}'
   ```

4. And finally you can get the desired asset based on its asset_id:

   ```
   curl -k -X GET -H "Authorization: Bearer <token>" -H "cache-control: no-cache" "https://<hostname>/v2/assets/57d708a9-
   6b6c-406e-8519-64b89442b9ba?catalog_id=a6a69cf6-6493-4c02-92ef-bc7e0a13ef63"
   ```

5. Use `{"query":"%"}` if you want to return all assets of a given type from the catalog.

Return more than 10k assets

There is in a limitation in the underlying ElasticSearch engine that does not allow for more that 10k assets to be returned by a search. Use the following workaround if you need to retrieving more results.

The way to go around this limitation is by using "search-after" https://www.elastic.co/guide/en/elasticsearch/reference/current/paginate-search-results.html#search-after.

1. Sort & run query using v3/search API, including "sort" into request body, for example

   ```
   {
      "size": 10000,
      "query": {
          "wildcard": {"metadata.name": "*"}
      },
      "sort": [
          {
              "last_updated_at": "asc"
          }
   ```

```
    ]
}
```

2. Now that you have got the sorted response, get the last_updated_at attribute of the last element in the result list (for example it is `[1597311295934]`) and include it into "search_after" to get the next batch. Example:

```
{
   "size": 10000,
   "query": {
       "wildcard": {"metadata.name": "*"}
   },
   "sort": [
       {
           "last_updated_at": "asc"
       }
   ],
   "search_after": [1597311295934]
}
```

3. Keep in mind that the sort order of search results is no longer valid if you use this technique but again. You are clearly not "searching" any more, rather "retrieving" all of the content by query.

IGC REST API

The Information Governance Catalog (IGC) REST API provides programmatic access to the Information assets catalog. A detailed API reference is available on your cluster under the following URL: `https://<hostname>/ibm/iis/igc-rest-explorer`.

Tutorials:

- https://developer.ibm.com/recipes/tutorials/interact-with-your-governance-metadata-in-igc-using-rest-apis-with-curl/
- http://wiki.glitchdata.com/images/1/11/Ibm_governance_catalog_rest_api_examples1.pdf
- https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.bg.doc/topics/c_bgapi_intro_CP.html).

The following sections provide examples for specific scenarios:

Get all databases

```
curl -k -u admin:password -X GET "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-rest/v1/search/database"
-H 'cache-control: no-cache' -H 'content-type: application/json'
curl -k -u admin:password -X GET "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-
rest/v1/search/database_column" -H 'cache-control: no-cache' -H 'content-type: application/json'
```

Query for column only

```
curl -k -u admin:password -X POST "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-rest/v1/search"  -H
'cache-control: no-cache' -H 'content-type: application/json' -d '{"properties":["_id","name"],"types":["database_column"]}'
```

Query for column and table name

```
curl -k -u admin:password -X POST "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-rest/v1/search"  -H
'cache-control: no-cache' -H 'content-type: application/json' -d '{"properties":["_id","name"],"types":
["database_column"],"where":{"conditions":[{"value":"CARD_DEBT","property":"name","operator":"="},
{"value":"MORTGAGE_JOIN","property":"database_table_or_view.name","operator":"="}],"operator":"and"}}'
```

Query for column, table, and schema name

```
curl -k -u admin:password -X POST "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-rest/v1/search"  -H
'cache-control: no-cache' -H 'content-type: application/json' -d '{"properties":["_id","name"],"types":
["database_column"],"where":{"conditions":[{"value":"CARD_DEBT","property":"name","operator":"="},
{"value":"MORTGAGE_JOIN","property":"database_table_or_view.name","operator":"="},
{"value":"DB2INST1","property":"database_table_or_view.database_schema.name","operator":"="}],"operator":"and"}}'
```

Query for column, table, schema, and database name

```
curl -k -u admin:password -X POST "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-rest/v1/search"  -H
'cache-control: no-cache' -H 'content-type: application/json' -d '{"properties":["_id","name"],"types":
["database_column"],"where":{"conditions":[{"value":"CARD_DEBT","property":"name","operator":"="},
{"value":"MORTGAGE_JOIN","property":"database_table_or_view.name","operator":"="},
{"value":"DB2INST1","property":"database_table_or_view.database_schema.name","operator":"="},
{"value":"mortgage","property":"database_table_or_view.database_schema.database.name","operator":"="}],"operator":"and"}}'
```

Update short an long descriptions

```
curl -k -u admin:password -X PUT "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-rest/v1/assets"  -H
'cache-control: no-cache' -H 'content-type: application/json' -d '{"ids":
["b1c497ce.60641b50.1ibn7rs1f.kn1snau.g7a58c.08163csmv99ttpuur9lls"],"_type":"database_column","properties":
{"short_description":"updated short description","long_description":"updated long description"}}'
```

When updating the descriptions you may increase the performance by updating multiple assets in a single step using the assets/patch API and a json body like this one:

```
{
  "patchUnits": [
    {
      "id": "b1c497ce.60641b50.1ibn7rs1f.kn1snau.g7a58c.08163csmv99ttpuur9lls",
      "path": "short_description",
      "value": "updated short description"
    },
    {
      "id": "b1c497ce.60641b50.1ibn7rs1f.kn1snau.g7a58c.08163csmv99ttpuur9lls",
      "path": "long_description",
      "value": "updated long description"
    },
    {
      "id": "b1c497ce.60641b50.1ibn8dh27.jr7qj7n.utmhug.4eotabi84r4ahb9ft0r38",
      "path": "short_description",
      "value": "updated short description"
    },
    {
      "id": "b1c497ce.60641b50.1ibn8dh27.jr7qj7n.utmhug.4eotabi84r4ahb9ft0r38",
      "path": "long_description",
      "value": "updated long description"
    }
  ]
}
```

Example:

```
curl -k -u admin:password -X POST "https://cpd-cpd-cpd.apps.cpd-demo.demo.ibmcloudpack.com/ibm/iis/igc-rest/v1/assets/patch"  -
H 'cache-control: no-cache' -H 'content-type: application/json' -d '{"patchUnits":
[{"id":"b1c497ce.60641b50.1ibn7rs1f.kn1snau.g7a58c.08163csmv99ttpuur9lls","path":"short_description","value":"updated short
description"},
{"id":"b1c497ce.60641b50.1ibn7rs1f.kn1snau.g7a58c.08163csmv99ttpuur9lls","path":"long_description","value":"updated long
description"},
{"id":"b1c497ce.60641b50.1ibn8dh27.jr7qj7n.utmhug.4eotabi84r4ahb9ft0r38","path":"short_description","value":"updated short
description"},
{"id":"b1c497ce.60641b50.1ibn8dh27.jr7qj7n.utmhug.4eotabi84r4ahb9ft0r38","path":"long_description","value":"updated long
description"}]}'
```

Information Analyzer REST API

The Information Analyzer (IA) REST API provides programmatic access to data quality projects. An API reference and examples are available at either of the following locations:

- https://www.ibm.com/docs/en/iis/11.7?topic=analyzer-developing-applications-http-api
- https://www.ibm.com/support/knowledgecenter/SSZJPZ_11.7.0/com.ibm.swg.im.iis.ia.restapi.doc/topics/r_restapi_run_data_rule.html
- https://www.ibm.com/support/pages/information-analyzer-rest-api-usage