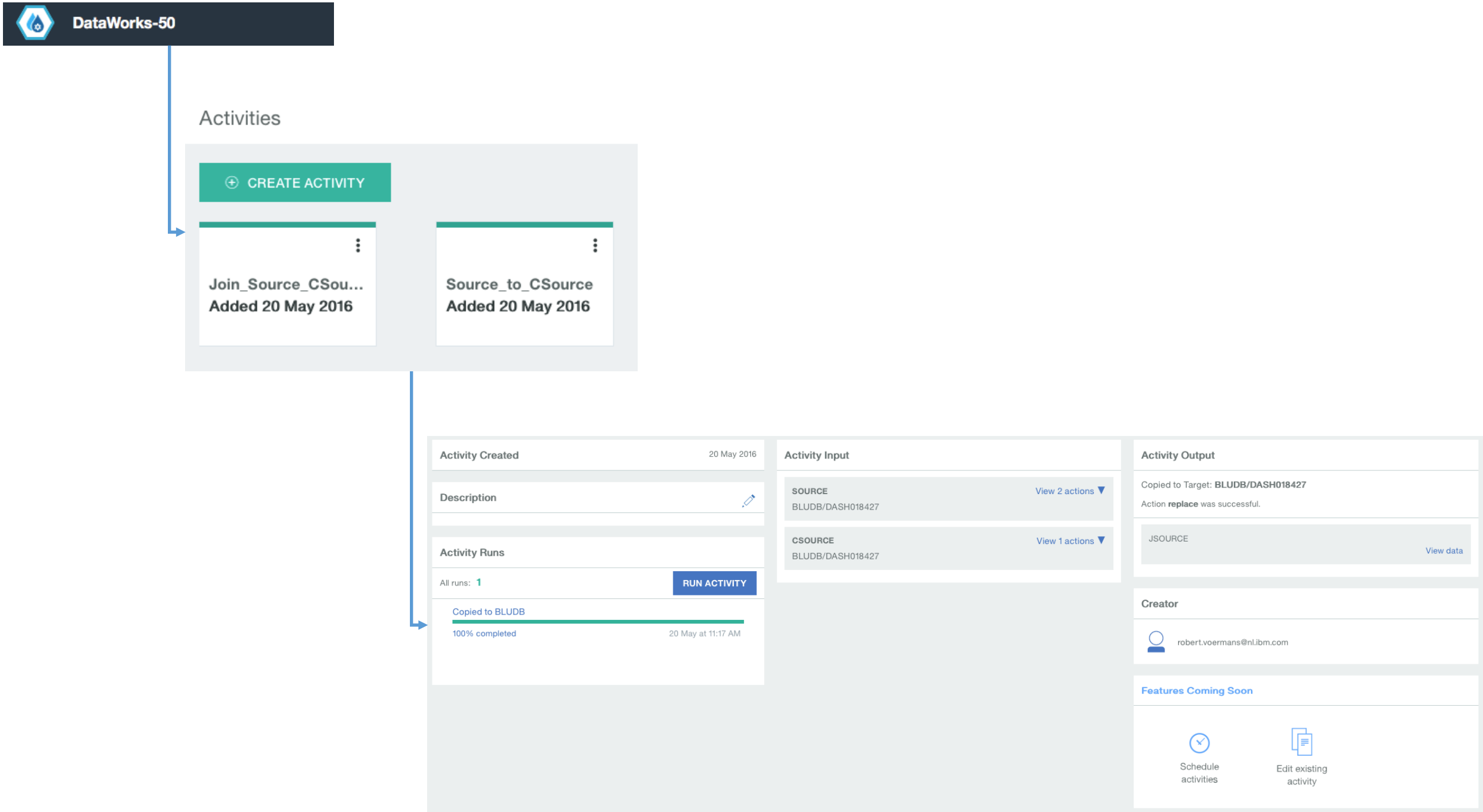
DataWork Lineage

Robert Voermans

Step 1: Hierarchy

- Before loading DW metadata, a structure or hierarchy of the metadata is to be defined


DataWorks-50
Activities
CREATE ACTIVITY
Join_Source_CSou...
Added 20 May 2016
Source_to_CSource
Added 20 May 2016
Activity Created
20 May 2016
Description
Activity Runs
All runs: 1
RUN ACTIVITY
Copied to BLUDB
100% completed
20 May at 11:17 AM
Activity Input
SOURCE
View 2 actions
BLUDB/DASH018427
CSOURCE
View 1 actions
BLUDB/DASH018427
Activity Output
Copied to Target: BLUDB/DASH018427
Action replace was successful.
JSOURCE
View data
Creator
robert.voermans@nl.ibm.com
Features Coming Soon
Schedule activities
Edit existing activity

Asset type bundle

- Please review http://www-01.ibm.com/support/docview.wss?uid=swg21699130
- An asset type bundle is a new definition of metadata is the Catalog (incl. structure and hierarchy)
- This is loaded once into the Governance Catalog and maintained via the Governance Catalog Open REST API
- It consists of an archive file (.zip) with an asset_type_descriptor.xml file and a set of icons and labels (for international language support)

Sample directory structure

Name	Date modified	Type	Size
i18n	20-May-16 15:17	File folder	
icons	20-May-16 15:01	File folder	
asset_type_descriptor	20-May-16 18:14	XML Document	5 KB

Name	Date modified	Type	Size
labels.properties	20-May-16 15:17	PROPERTIES File	1 KB

```
family1=DataWorks
tree1=DataWorks Services

class.Service=DataWorks Service
class-plural.Service=DataWorks Services
attr.Service.Plan=Plan
enum.Plan.Starter=Starter
enum.Plan.Personal=Personal
enum.Plan.Professional=Professional
enum.Plan.Enterprise=Enterprise

class.Activity=DataWorks Activity
class-plural.Activity=DataWorks Activities
attr.Activity.LastRun=Last Run

class.Join=DataWorks Join
class-plural.Join=DataWorks Joins

class.Copy=DataWorks Copy
class-plural.Copy=DataWorks Copies

class.Instance=DataWorks Instance
class-plural.Instance=DataWorks Instances
attr.Instance.Location=Location

class.Input=DataWorks Input
class-plural.Input=DataWorks Inputs

class.Output=DataWorks Output
class-plural.Output=DataWorks Outputs

class.Action=DataWorks Instance Action
class-plural.Action=DataWorks Instance Actions
attr.Action.Logging=Logging
```

Name	Date modified	Type	Size
Action-bigIcon	20-May-16 14:07	GIF image	2 KB
Action-icon	20-May-16 14:08	GIF image	2 KB
Activity-bigIcon	20-May-16 14:09	GIF image	2 KB
Activity-icon	20-May-16 14:10	GIF image	2 KB
Copy-bigIcon	20-May-16 14:11	GIF image	2 KB
Copy-icon	20-May-16 14:12	GIF image	2 KB
Input-bigIcon	20-May-16 14:12	GIF image	2 KB
Input-icon	20-May-16 14:13	GIF image	2 KB
Instance-bigIcon	20-May-16 14:45	GIF image	2 KB
Instance-icon	20-May-16 14:45	GIF image	2 KB
Join-bigIcon	20-May-16 14:53	GIF image	2 KB
Join-icon	20-May-16 14:54	GIF image	2 KB
Output-bigIcon	20-May-16 14:55	GIF image	2 KB
Output-icon	20-May-16 14:56	GIF image	2 KB
Service-bigIcon	20-May-16 14:57	GIF image	2 KB
Service-icon	20-May-16 14:58	GIF image	2 KB

2 icons per asset, icon (16x16), bigIcon(32x32)

Sample asset_type_descriptor.xml (partly)

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<descriptor xmlns="http://www.ibm.com/iis/igc/asset-type-descriptor"
    bundleId="DataWorks"> <!-- short ID for preventing clashes with other asset bundles -->

<!-- ************************************************************** -->
<!-- ********* GUI Containers: Families, Trees ********** -->

    <family position="2" classRefs="Service, Activity, Instance, Action" >
        <label key="family1" inDefaultLocale="DataWorks" />
    </family>

    <tree position="3" rootClassRefs="Service">
        <label key="tree1" inDefaultLocale="DataWorks Services" />
    </tree>

<!-- ************************************************************** -->
<!-- ********* Structural definitions: Classes, Attributes ********** -->
<!-- *********    Hierarchy: Service - Activity(Join / Copy) - Instance(Input / Output) - Action ********** -->

<!-- ********* Service ********** -->

    <class localId="Service" dataAccessRole="None" canHaveImage="true">
        <label key="class.Service" inDefaultLocale="DataWorks Service" />
        <pluralLabel key="class-plural.Service" inDefaultLocale="DataWorks Services" />

        <headerSection>
            <attribute localId="Plan" type="String" editable="true">
                <label key="attr.Service.Plan" inDefaultLocale="Plan" />
                <validValue localId="Starter">
                    <label key="enum.Plan.Starter" inDefaultLocale="Starter" />
                </validValue>
                <validValue localId="Personal">
                    <label key="enum.Plan.Personal" inDefaultLocale="Personal" />
                </validValue>
                <validValue localId="Professional">
                    <label key="enum.Plan.Professional" inDefaultLocale="Professional" />
                </validValue>
                <validValue localId="Enterprise">
                    <label key="enum.Plan.Enterprise" inDefaultLocale="Enterprise" />
                </validValue>
            </attribute>
        </headerSection>
    </class>
```

Implemented hierarchy

- **Service** (the Bluemix / DW service)
  - **Activity** (Has two instances: Either “Join” or ”Copy”, representing the activity functionality. They will inherit the activity definition)
    - **Instance** (Has two instances: Either “Input” or “Output”, representing source(s) or target(s) . They will inherit the instance definition)
      - **Action**

Metadata attributes

- Service
  - Plan (Either “Starter”, “Personal”, “Professional” or “Enterprise”
- Activity
  - Last Run (Representing the last run date)
- Instance
  - Location (Location information of the source and/or target: Like DB and schema)
- Action
  - Logging (What was done?)

Upload hierarchy

- Using REST-Explorer
  - https://<iis-server-ip>:<iis-server-port>/ibm/iis/igc-rest-explorer
  - For example: https://is-server:9445/ibm/iis/igc-rest-explorer
- Mentioned directory structure and files archived into zip file
  - For example: DataWorks.zip

Sample REST-API call

POST /bundles/ — Register a new asset bundle

Implementation Notes

Registers a new bundle. The bundle file format must follow the guidelines described here.

Hide Examples

Register the DataMass asset bundle.

Requests

```
POST https://localhost:9443/ibm/iis/igc-rest/v1/bundles HTTP/1.1
Accept: application/json
Host: localhost:9443
Accept-Encoding: identity
Content-Length: 49784
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryvKFbEXlDpaTy8RkR
------WebKitFormBoundaryvKFbEXlDpaTy8RkR
Content-Disposition: form-data; name="file"; filename="DataMass.zip"
Content-Type: application/x-zip-compressed
...
------WebKitFormBoundaryvKFbEXlDpaTy8RkR--
```

Response

```
HTTP/1.1 200 OK
Content-Type:
Content-Length: 0
```

Parameters

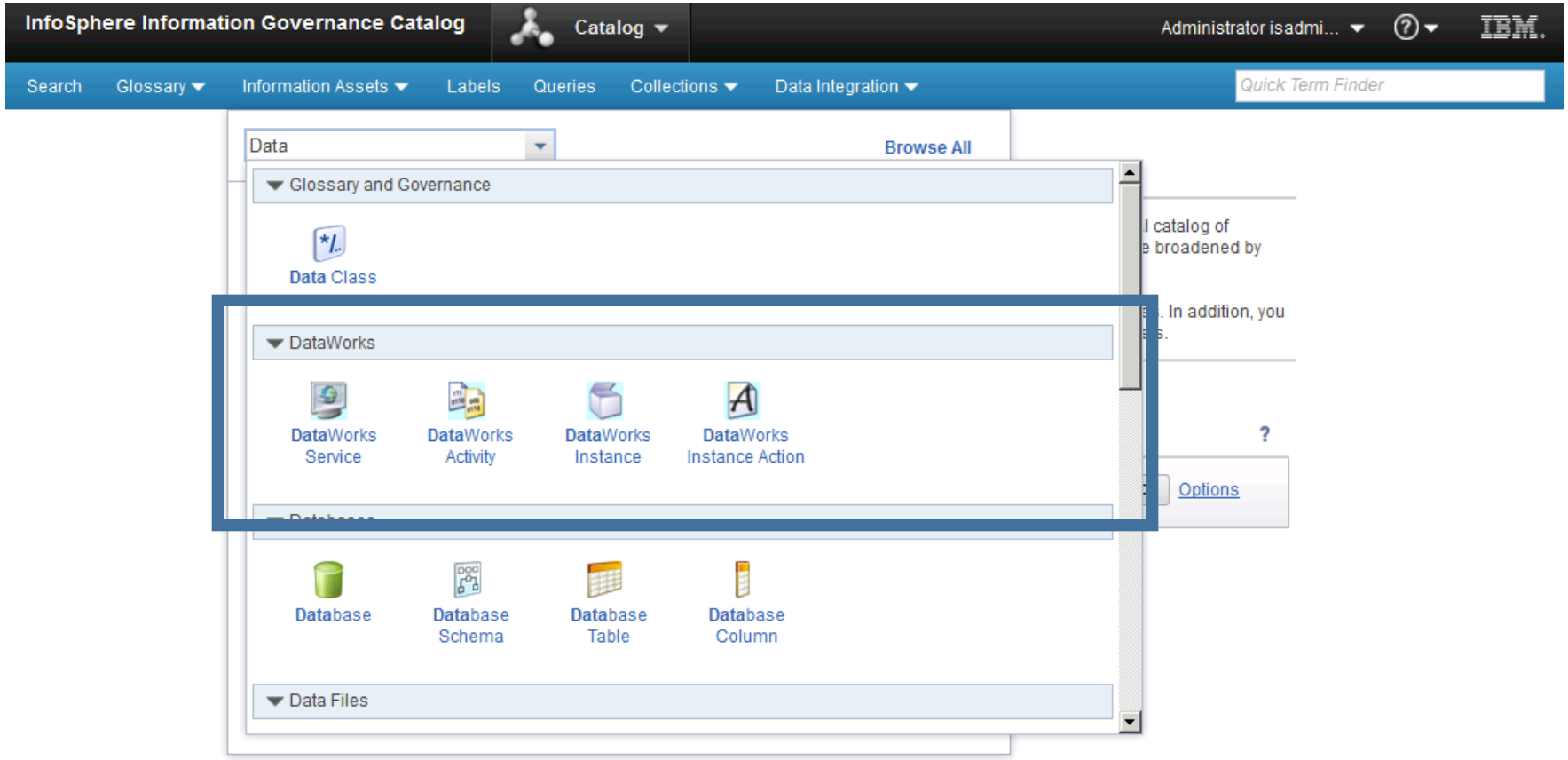
Parameter	Value	Description	Parameter Type	Data Type
file	Browse... DataWorks.zip	**Bundle file**	body	file

Response Messages

HTTP Status Code	Reason	Response Model
400	Bad Request	
401	Unauthorized	

Try it out! Clear

Step 1: Results in the Governance Catalog



The Governance Catalog now has a new section in the Information Assets structure called DataWorks with 4 new asset types called (DataWorks) Service, Activity, Instance and Action

Step 2: Instantiate metadata

- With a hierarchy available we now can upload metadata assets using the new asset type(s) from the bundle
- This is done via the Governance Catalog Open REST API
  - Using REST-Explorer
    - https://<iis-server-ip>:<iis-server-port>/ibm/iis/igc-rest-explorer
    - For example: https://is-server:9445/ibm/iis/igc-rest-explorer
    - Using a specific XML format directly in the call

Sample REST-API call

POST /bundles/assets — Upload assets defined by asset bundles

Implementation Notes

Uploads assets that are defined by a registered bundle by using an XML file format. The XML file format must follow the guidelines described here.

Hide Examples

Publish assets defined by the DataMass asset bundle.

Requests

```
POST /ibm/iis/igc-rest/v1/bundles/assets HTTP/1.1
Accept: application/json
Host: localhost:9443
Accept-Encoding: identity
```

Response

```
HTTP/1.1 200 OK
Content-Type:
Content-Length: 0
```

Parameters

Parameter	Value	Description	Parameter Type	Data Type
string	(required)	**Asset Document XML string**	body	string

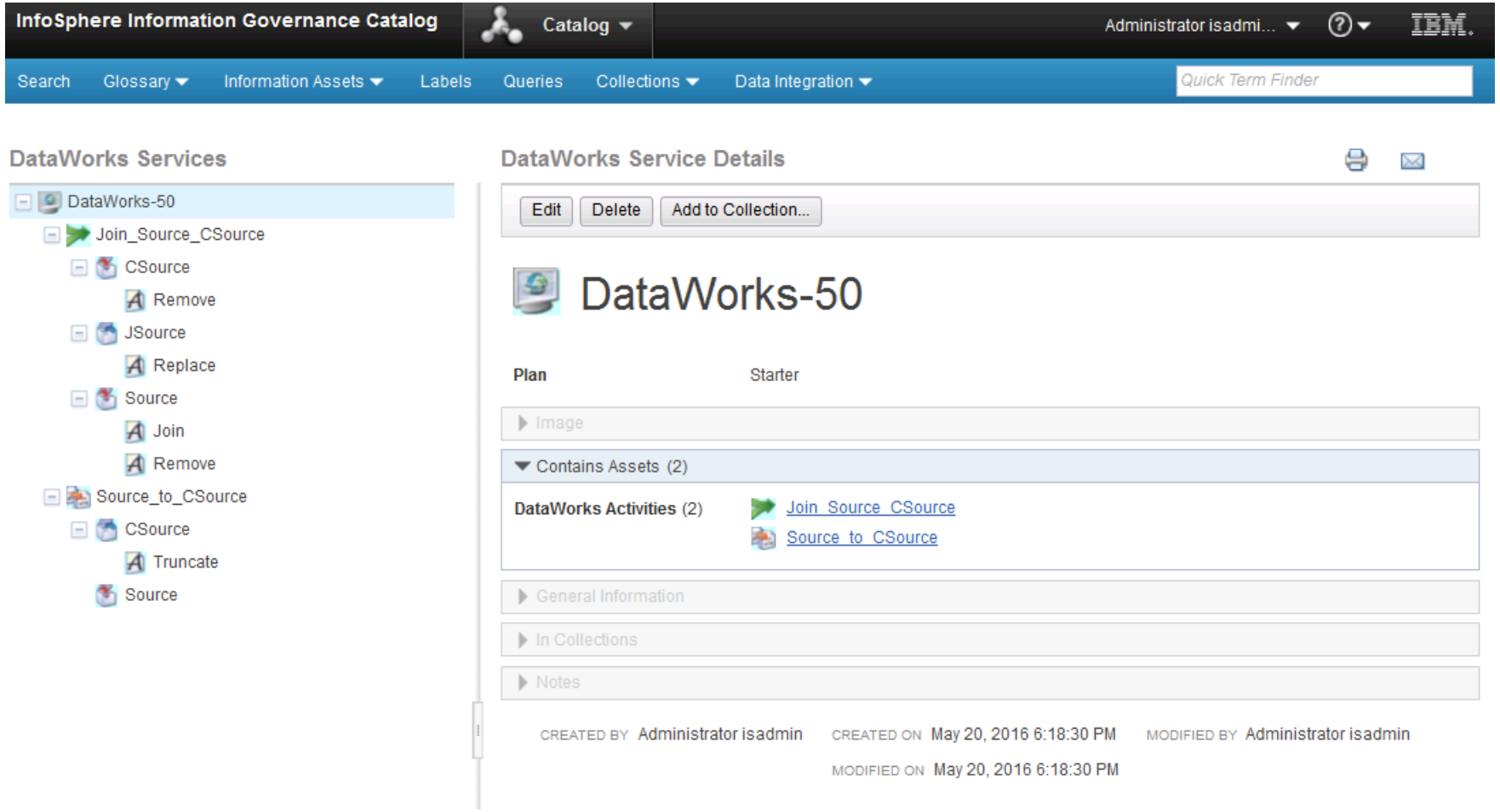
Parameter content type: application/xml

Try it out! Clear

Sample XML (partly)

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<doc xmlns="http://www.ibm.com/iis/flow-doc">
    <assets>
        <asset class="$DataWorks-Service' repr="DataWorks-50" ID="a1">
            <attribute name="name" value="DataWorks-50" />
            <attribute name="$Plan" value="Starter" />
        </asset>
        <asset class="$DataWorks-Join" repr="Join_Source_CSource" ID="a2">
            <attribute name="name" value="Join_Source_CSource" />
            <attribute name="$LastRun" value="20 May 2016 at 11:17 AM" />
            <reference name="$Service" assetIDs="a1" />
        </asset>
        <asset class="$DataWorks-Copy" repr="Source_to_CSource" ID="a3">
            <attribute name="name" value="Source_to_CSource" />
            <attribute name="$LastRun" value="20 May 2016 at 10:24 AM" />
            <reference name="$Service" assetIDs="a1" />
        </asset>

        <asset class="$DataWorks-Input" repr="Source" ID="a4">
            <attribute name="name" value="Source" />
            <attribute name="$Location" value="BLUDB/DASH018427" />
            <reference name="$Activity" assetIDs="a3" />
        </asset>

        <asset class="$DataWorks-Output" repr="CSource" ID="a5">
            <attribute name="name" value="CSource" />
            <attribute name="$Location" value="BLUDB/DASH018427" />
            <reference name="$Activity" assetIDs="a3" />
        </asset>

        <asset class="$DataWorks-Action" repr="Truncate" ID="a6">
            <attribute name="name" value="Truncate" />
            <attribute name="$Logging" value="Action truncate was successful" />
            <reference name="$Instance" assetIDs="a5" />
        </asset>

        <asset class="$DataWorks-Input" repr="Source" ID="a7">
            <attribute name="name" value="Source" />
            <attribute name="$Location" value="BLUDB/DASH018427" />
            <reference name="$Activity" assetIDs="a2" />
        </asset>

        <asset class="$DataWorks-Input" repr="CSource" ID="a8">
            <attribute name="name" value="CSource" />
            <attribute name="$Location" value="BLUDB/DASH018427" />
            <reference name="$Activity" assetIDs="a2" />
        </asset>
```

Step 2: Results in the Governance Catalog



New assets are imported using the defined hierarchy. They have their own identity (icon) and attributes

Step 3: Add flows (or lineage)

- When the new metadata assets are available you are now able to create flows (or lineage) with the new assets and existing assets
- This is done via the Governance Catalog Open REST API
  - Using REST-Explorer
    - https://<iis-server-ip>:<iis-server-port>/ibm/iis/igc-rest-explorer
    - For example: https://is-server:9445/ibm/iis/igc-rest-explorer
    - Using a specific XML format directly in the call

Sample REST-API call

POST /flows/upload — Upload flow graph to the flows model

Implementation Notes

Upload flow graph. The graph file format must follow the guidelines described here.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
string	(required)	**Flow Document XML string**	body	string

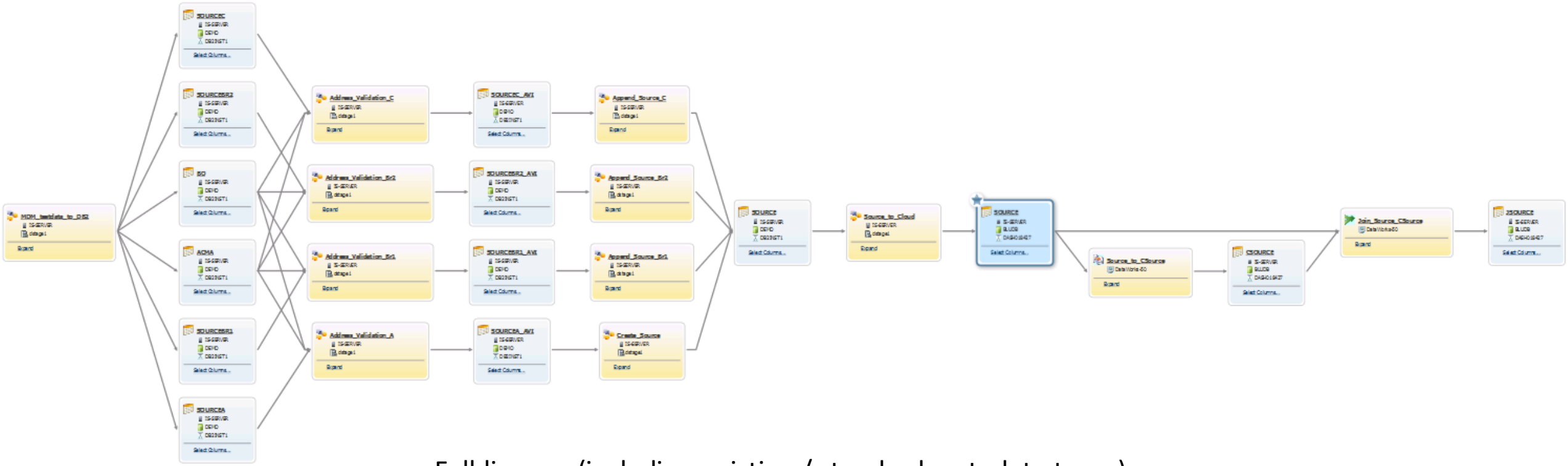
Parameter content type: application/xml

Try it out! Clear

Sample XML (partly)

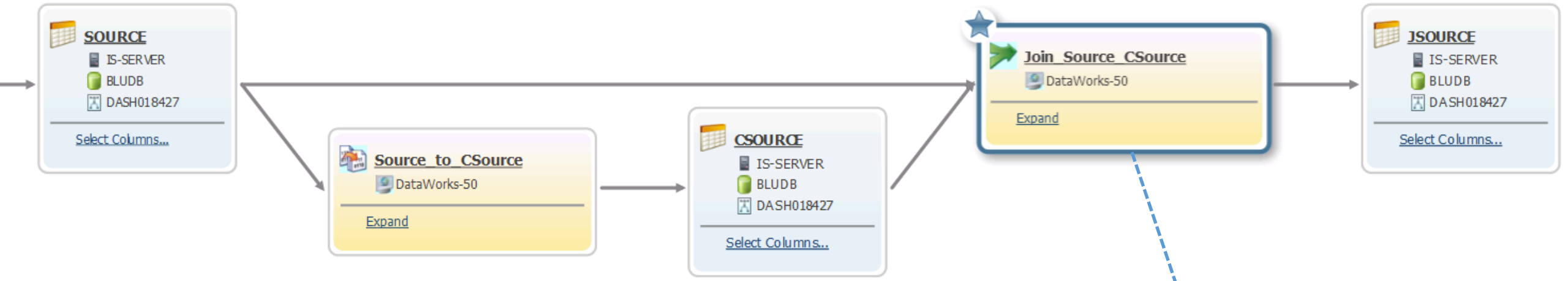
```
    <flowUnits>
        <flowUnit assetID="a2">
            <subFlows flowType="DESIGN" comment="Join sources into target">
                <flow sourceIDs="b1c497ce.54bd3a08.7pi59j9do.oqk44pi.l1kvf4.kikqefoc5q03ok78t04dl" targetIDs="a7"/>
                <flow sourceIDs="b1c497ce.54bd3a08.7pi59jbpg.vr73n7h.97i5d0.e7omrssnliitcad5rcfg3" targetIDs="a8"/>
                <flow sourceIDs="a9" targetIDs="b1c497ce.54bd3a08.7pi59jbpg.vr77c31.njjd9e.kgisdk7fsr03kjvj0i7jn"/>
                <flow sourceIDs="a7 a8" targetIDs="a9"/>
            </subFlows>
        </flowUnit>
        <flowUnit assetID="a3">
            <subFlows flowType="DESIGN" comment="Copy source to target">
                <flow sourceIDs="b1c497ce.54bd3a08.7pi59j9do.oqk44pi.l1kvf4.kikqefoc5q03ok78t04dl" targetIDs="a3"/>
                <flow sourceIDs="a5" targetIDs="b1c497ce.54bd3a08.7pi59jbpg.vr73n7h.97i5d0.e7omrssnliitcad5rcfg3"/>
                <flow sourceIDs="a3" targetIDs="a5"/>
            </subFlows>
        </flowUnit>
    </flowUnits>
</doc>
```

Step 3: Results in the Governance Catalog



Full lineage (including existing / standard metadata types)

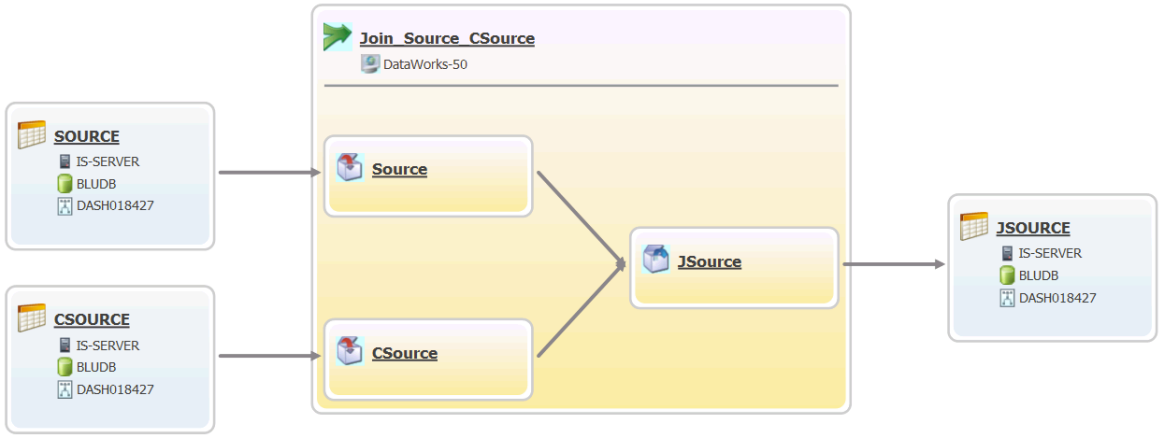
Step 3: Results in the Governance Catalog



Partly lineage (zoomed into DataWorks components)

Click Expand on Join

Expanded lineage for Join



Summary

- It takes three REST API calls to define / import new metadata assets and create lineage with these assets (also using existing metadata assets if required)