TopBraid Data Platform

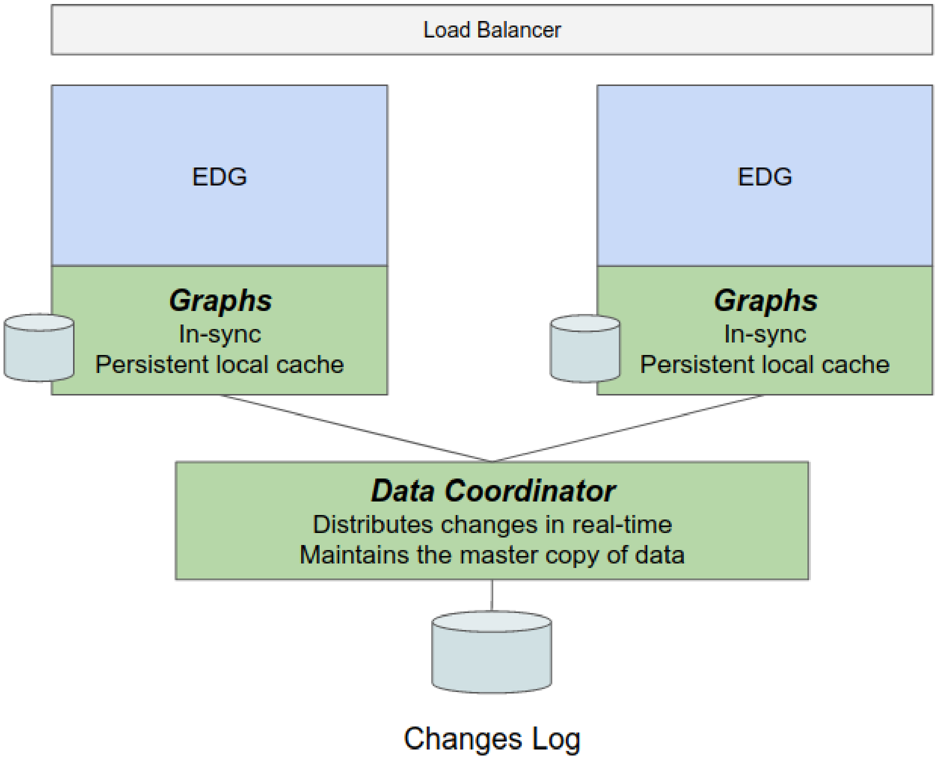
Page Contents

1 TopBraid Data Platform
- 1.1 Obtaining the Software
- 1.2 Creating Shared (DP-managed) Graphs
- 1.3 Create New Asset Collection
- 1.4 Data Coordinator Outages
- 1.5 Data Coordinator Requirements
- 1.6 EDG Failover Configuration
- 1.7 Example RDF Patch
- 1.8 Example Data Platform connector file.
- 1.9 Tutorial
  - 1.9.1 File-based persistent patch storage
  - 1.9.2 Run the patch log server
  - 1.9.3 Configure an EDG server
  - 1.9.4 Create an asset collection in DP
  - 1.9.5 Verify Log Operation

TopBraid Data Platform

The TopBraid Data Platform is a high availability solution for TopBraid EDG servers. It enables continuous operation of business functions by replicating data across a cluster of EDG servers. Each EDG is kept up to date and server can respond to a client/application request. Together with a load balancer to direct requests, user and application access to the EDG data remains available even if some of the servers are offline.

This availability is achieved by having a cluster of EDG servers communicating with a *data coordinator* (DC) server that receives changes from any clustered server whenever it has updates. The DC server propagates the changes to all other EDG servers, keeping them all up-to-date. Additionally, whenever a new EDG server joins the cluster, it is first brought up-to-date before it starts servicing user/application requests. Each EDG server keeps a local, persistent cache of application data so that start-up only involves applying changes missed by that server.



Each EDG or EDG Explorer server has its own copy of a database of all RDF graphs managed by the Data Platform. When changes to the data are made, the changes are recorded locally as RDF Patches.

At the end of the operation (the end of the HTTP request), any changes are sent to the Data Coordinator and made safe. When the patches are confirmed to be safe, the EDG server triple store declares the request has committed and responds to the HTTP request. It is the state of the RDF Patch Log that determines the state of the EDG server for Data Platform backed graphs.

When operations arrive at EDG server, the EDG server checks that its local database is up-to-date and, if it is not, fetched patched from the Data Coordinator and applies them to make sure it is at the latest version.

The Data Coordinator can be a single machine, with file-backed persistence or a cluster of servers with storage in a cloud object store. There are tradeoffs in complexity of operation as discussed below.

The TopBraid Data Platform is based on the open source (Apache License) RDF Delta system.

Obtaining the Software

The Data Coordinator (the patch log server) can be downloaded from:

> https://download.topquadrant.com/dp/

EDG server and EDG explorer already have the data platform client software included and no extra software or is required for EDG.

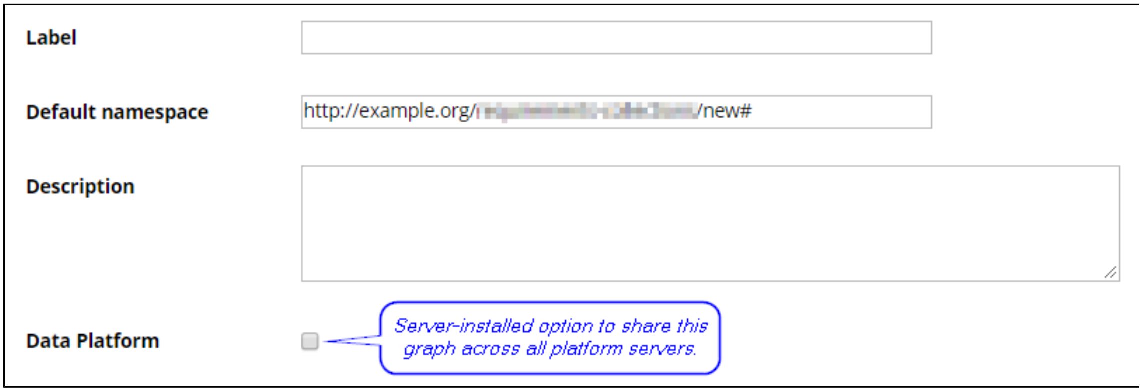
Note that the version corresponding to the EDG version as given in the table on the download page must be used.

Creating Shared (DP-managed) Graphs

When the DP feature is enabled for an EDG server and the data coordinator server is running, creation of new asset collections will show a Data Platform checkbox option. Selecting this will cause the new asset collection to be shared among all DC-connected EDG servers.

Create New Asset Collection

This creates a new Asset Collection with yourself as the manager.



Label

Default namespace http://example.org/[redacted]/new#

Description

Data Platform □

Server-installed option to share this graph across all platform servers.

There may be a few-second delay in propagating collection changes. The DP checkbox option only appears when the DC server is contactable. Asset collections created without the DP option will exist only on the original server.

Data Coordinator Outages

If the Data Coordinator is not running then:

- shared collections can be read (viewed, queried)
- updates to shared collections are not possible
- new, shared collections cannot be created.

Restarting the Data Coordinator re-enables the update and sharing features. Restarting EDG is not necessary.

Data Coordinator Requirements

The DC server is a Java webserver process that coordinates changes across a cluster of EDG servers. For production use, it should be run on a machine separated from all of the clustered EDG machines. For development and experimentation, it can be run on a shared machine.

The DC server stores patches on-disk. The disk storage must be backed-up, which can be done by file backup on a live server. The robustness of the system is determined by the robustness of the file storage so choosing local disk in the DC-server is limiting. It is primarily a patch server for patch files and is not computationally demanding.

EDG Failover Configuration

In a high-availability configuration, two or more EDG servers provide the service. Each has a complete copy of the replicated graphs.

Typically, a deployment will put a load balancer in front of the two servers so that the clients (web browsers or other applications) use the same URL to access either server. It also allows a deployment to move EDG servers between physical hardware for maintenance and system upgrade.

There are two choices for Patch Log Server: either run with a file-backed single server or run with replicated servers for continuous operation.

The file-based patch log server can use any OS-supported filesystem - local storage or remote disk-array and the reliability and durability of files written are determined by the choice of filesystem. If the single server style is used, then operation is interrupted if the patch log server is unavailable. The EDG servers keep running but will be unable to update data until the patch log server restarts. Patch log server startup is very fast. There is no need to restart EDG servers.

The advantage is the simplicity of operation so there is a tradeoff for small deployments of simplicity and continuous operation.

For continuous operation, three or more patch log servers run, on separate hardware; the system uses Apache Zookeeper to manage a system wide database of the patch state of the deployment. Storage of patches is in an object store, Amazon S3. There are publicly available adapters to provide the Amazon S3 API over other storage choices such as Apache Cassandra.

Example RDF Patch

```
H id <uuid:6e0288f5-ed79-4ab7-b4cc-8386ee32fadf> .
H prev <uuid:3e346531-777e-4169-abcd-284a7a5b22ac> .
TX .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/ns/shacl#NodeShape> .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2002/07/owl#Class> .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/2000/01/rdf-schema#subClassOf>
<http://www.w3.org/2002/07/owl#Thing> .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/2000/01/rdf-schema#label>
"SubClass" .
TC .
```

Example Data Platform connector file.

```
# baseURI: urn:x-evn-master:ontconfig
# imports: http://topbraid.org/sparqlmotionlib
@prefix dp: <http://org.topbraid/dataplatform#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sm: <http://topbraid.org/sparqlmotion#> .
@prefix sml: <http://topbraid.org/sparqlmotionlib#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<urn:x-evn-master:ont> a dp:Graph ;
    dp:coordinator "http://localhost:1066/" ;
    dp:datasource "ont" ;
    dp:localData "ont/data" ;
    dp:localDataLocation "/home/afs/TQ/EDG/TBS/build/workspaces/TBCME/Zone/ont/data" ;
    sml:baseURI "urn:x-evn-master:ont" ;
    sml:graphName "urn:x-evn-master:ont" ;
    sml:imports () ;
    .

<urn:x-evn-master:ontconfig> a owl:Ontology ;
    owl:imports <http://topbraid.org/sparqlmotionlib> ;
    .
```

Tutorial

This section gives a short walkthrough for setting up a EDG system with the Data Platform. It covers the use of the TopBraid Data Platform with file-backed storage of patches. The Data Coordinator is the RDF Delta patch log server.

- File-based persistent patch storage
- Run the patch log server
- Connect an EDG server
- Create an asset collection in DP
- Verify log operation

File-based persistent patch storage

The simplest configuration of a patch log server uses the servers file system for patch storage. This is an OS-supported filesystem - local storage or remote disk-array and the reliability and durability of files written are determined by the choice of filesystem.

Run the patch log server

In production, the patch log server should be run on a separate machine to the server running the TopBraid server, not just a separate virtual machine.

For development and testing, and this tutorial, it can run on the localhost for convenience.

After obtaining the software, run the patch log server with:

java -jar rdf-delta-server-*VERSION*.jar --base *DIRECTORY*

where *VERSION* is the version number of the RDF Delta softare corresponding to the version of the EDG server in use.

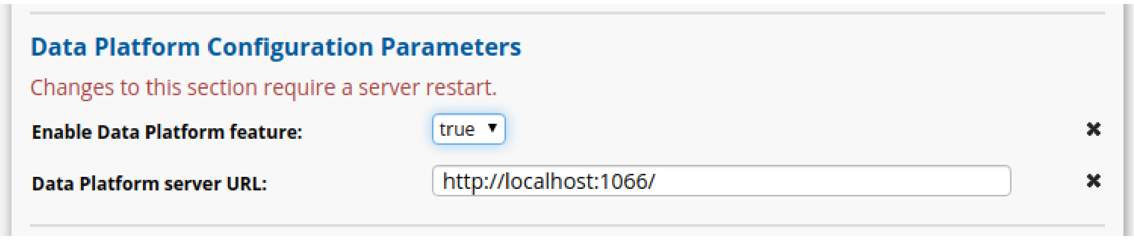
DIRECTORY is an initially empty, existing directory where the patch log server stores the patch logs in files.

Example patch server output:

```
[2019-05-08 10:46:59] Delta INFO Provider: file
[2019-05-08 10:46:59] Config INFO Delta Server port=1066
[2019-05-08 10:46:59] Delta INFO RDF Delta 0.7.0 2019-03-12T14:54:40+0000
[2019-05-08 10:46:59] Delta INFO No data sources
```

Configure an EDG server

The EDG server needs to be setup to use the Data Platform. In "Server Administration" -> "EDG Configuration Parameters", press "Edit" and in the "Data Platform Configuration Parameters" section, set "Enable Data Platform feature" to true and check the "Data Platform server URL" setting which should be "http://localhost:1066/" for this tutorial setup.



Data Platform Configuration Parameters

Changes to this section require a server restart.

Enable Data Platform feature: true ▾ ✖

Data Platform server URL: http://localhost:1066/ ✖

Restart the EDG server.

These settings are recorded in the dynamic/config.ttl file in the workspace:*<workspace/*server.topbraidlive.org/dynamic/config.ttl. If preferred, this file can be edited directly when the EDG server is not running.

```
:configuration a cfg:ServerConfiguration ;
    . . .
    cfg:dpFeature true ;
    cfg:dpServerURL "http://localhost:1066/" ;
    cfg:dpZone "Zone" ;
    . . .
```

Property	Type	Value
`cfg:dpFeature`	Boolean	`true` for Data Platform.
`cfg:dpServerURL`	String	The URL for the patchlog server.
`cfg:dpZone`	String	

The feature setting is set to `true` (no quotes) to enable the Data Platform.

HOST is the DNS name or IP address of the patch log server machine. This should be a separate machine to the server running the TopBraid server.

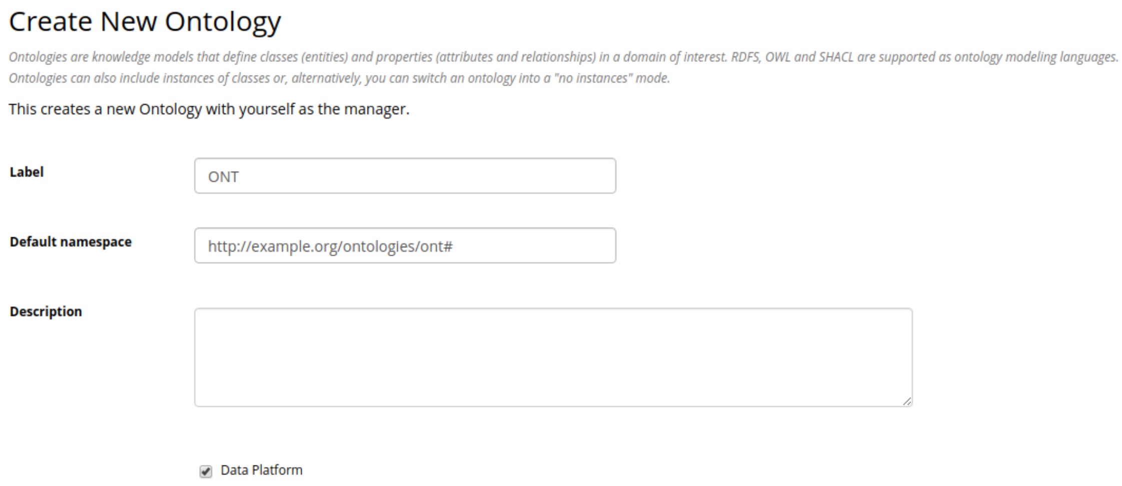
The "Zone" is the place within the server workspace where the Data Platform software keeps the local state information. This does not usually needed to be changed.

The log file (`catalina.log`) should show the Data Platform is running including an entry:

```
DataPlatform: Data Platform : server http://localhost:1066/
```

Create an asset collection in DP

The EDG UI will have an additional checkbox to allow placement of the graph into the DataPlatform.



Create New Ontology

Ontologies are knowledge models that define classes (entities) and properties (attributes and relationships) in a domain of interest. RDFS, OWL and SHACL are supported as ontology modeling languages. Ontologies can also include instances of classes or, alternatively, you can switch an ontology into a "no instances" mode.

This creates a new Ontology with yourself as the manager.

Label ONT

Default namespace http://example.org/ontologies/ont#

Description

☑ Data Platform

Changes to the ontology will be recorded in the patch log. Patches are handled by the system and not displayed to the user but here is an example patch creating a new subclass in an ontology:

```
H id <uuid:6e0288f5-ed79-4ab7-b4cc-8386ee32fadf> .
H prev <uuid:3e346531-777e-4169-abcd-284a7a5b22ac> .
TX .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http
://www.w3.org/ns/shacl#NodeShape> .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http
://www.w3.org/2002/07/owl#Class> .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/2000/01/rdf-schema#subClassOf> <http
://www.w3.org/2002/07/owl#Thing> .
A <http://example.org/ontologies/ont#SubClass> <http://www.w3.org/2000/01/rdf-schema#label>
"SubClass" .
TC .
```

Verify Log Operation

To illustrate that the EDG server is indeed running with the Data Platform, stop TopBraid Composer, restart with a clean workspace, and restart EDG "personal edition". If running EDG in Tomcat, stop Tomcat, delete the following directories and files, or completely delete the EDG workspace as if reinstalled:

```
<Workspace>/Zone/ont/
<Workspace>/Zone/ont.tch/
<Workspace>Repositories/ont.dpc
<Workspace>Repositories/ont.tch.dpc
```

When the EDG: server starts up, the log will show ontology (called "ont" below) restored from the patch log server.

```
[TopBraid Info] 2019-05-08T10:56:27+0100 - DataPlatform: New graph: [id:969bb7, ont.tch, <urn:x-evn-
master:ont.tch>]
[TopBraid Info] 2019-05-08T10:56:27+0100 - DataPlatform: Create DP Graph [id:969bb7] : datasource=ont.
tch, baseURI=urn:x-evn-master:ont.tch, coordinator=http://localhost:1066/
[TopBraid Info] 2019-05-08T10:56:27+0100 - DataPlatform: Attach graph ont.tch in 0.166s
[TopBraid Info] 2019-05-08T10:56:27+0100 - DataPlatform: New graph: [id:84d7ff, ont, <urn:x-evn-
master:ont>]
[TopBraid Info] 2019-05-08T10:56:27+0100 - DataPlatform: Create DP Graph [id:84d7ff] :
datasource=ont, baseURI=urn:x-evn-master:ont, coordinator=http://localhost:1066/
[TopBraid Info] 2019-05-08T10:56:27+0100 - DataPlatform: Attach graph ont in 0.261s
```

The graph has been created and rebuilt from the patch log.