Red Hat JBoss Data Virtualization 6.3
Glossary Guide

David Sage   Nidhi Chaudhary

Red Hat JBoss Data Virtualization 6.3 Glossary Guide

David Sage
dlesage@redhat.com

Nidhi Chaudhary
nchaudha@redhat.com

Legal Notice

Abstract

This glossary is your guide to Red Hat JBoss Data Virtualization terminology.

Table of Contents

GLOSSARY

1. A

Access Layer

> The access layer is the interface through which applications submit queries (relational, XML, XQuery or procedural) to the virtual database via JDBC, ODBC or Web services.

Admin API

> The Admin API package, **org.teiid.adminapi**, provides methods that allow developers to connect to and configure a Red Hat JBoss Data Virtualization instance at runtime from within other applications.

Administration Tools

> The administration tools are a suite of tools made available for administrators to configure and monitor Red Hat JBoss Data Virtualization.

AdminShell

> The AdminShell provides a script-based programming environment enabling users to access, monitor and control Red Hat JBoss Data Virtualization.

AdminShell GUI

> The AdminShell GUI allows users to administer Red Hat JBoss Data Virtualization through the development and running of scripts in a graphical environment. It lets you write scripts using a GUI text editor featuring syntax highlighting.

Apache CXF

> Apache CXF is an open source framework for developing service-oriented architectures (SOAs). CXF lets users build and develop services using frontend programming APIs, like JAX-WS and JAX-RS. These services can use a variety of protocols such as SOAP, XML/HTTP, RESTful HTTP, or CORBA and work over a variety of transports such as HTTP, JMS or JBI. For more information about CXF, refer to http://cxf.apache.org/docs/

2. B

3. C

Cluster

> A cluster is a group of loosely-connected computers that work together in such a way that they can be viewed as a single system. The components (or nodes) of a cluster are connected to each other through fast local area networks, each node running its own operating system. Clustering middleware orchestrates collaboration between the nodes, allowing users and applications to treat the cluster as a single processor. Clusters are effective for scalable enterprise applications, since performance is improved by adding more nodes as required. Furthermore, if at anytime one node fails, others can take on its load.

Checksum Validation

Checksum validation is used to ensure a downloaded file has not been corrupted. Checksum validation employs algorithms that compute a fixed-size datum (or checksum) from an arbitrary block of digital data. If two parties compute a checksum of a particular file using the same algorithm, the results will be identical. Therefore, when computing the checksum of a downloaded file using the same algorithm as the supplier, if the checksums match, the integrity of the file is confirmed. If there is a discrepancy, the file has been corrupted in the download process.

Connector Development Kit

The Connector Development Kit is a Java API that allows users to customize the connector architecture (translators and resource adapters) for specific integration scenarios.

Connector Framework

Red Hat JBoss Data Virtualization includes a set of translators and resource adapters that enable virtual databases to access their dependent physical data sources. In other words, they provide transparent connectivity between the query engine and the physical data sources. Collectively, this set of translators and adapters is known as the connector framework. You can develop your own custom translators and resource adapters for data sources that are not directly supported by Red Hat JBoss Data Virtualization.

4. D

Dashboard

The Dashboard builder allows you to connect to virtual databases through the DV JDBC driver to visualize the data for testing and Business Analytics.

Data Federation

Data Federation refers to the ability of Red Hat JBoss Data Virtualization to integrate multiple data sources so that a single query can return results from one or more data sources.

Data Provider

Data providers are entities that are configured to connect to a data source (a CSV file or database), collect the required data, and assign them the data type. You can think about them as database queries. The collected data can be then visualized in indicators on pages or exported in a variety of formats such as XLS or CSV.

Data Role

Data roles are sets of permissions used to control users' access to data. They include read-only and read-write access roles.

Data Source

A data source is a repository for data. Query languages enable users to retrieve and manipulate data stored in these respositories.

Database Tool

The database tool is used to configure the database used by Red Hat JBoss Data Virtualization. The database in Red Hat JBoss Data Virtualization is required by three items: ModeShape, the Dashboard Builder and the command/audit logging functionality. If you are using Red Hat JBoss Data Virtualization in conjunction with Red Hat JBoss Fuse Service Works, BPEL, and SwitchYard/jBPM integration also make use of this functionality.

Design Tools

> The design tools are available to assist users in setting up Red Hat JBoss Data Virtualization for their desired data integration solution.

Distributed Caching

> Distributed caching extends the traditional concept of caching on single machines to allow caching across multiple servers. Using distributed cache, applications can treat the cache as a single entity, and cache can grow in size and transactional capacity by adding more servers.

5. E

Execution Environment

> The Execution Environment is used to actualize the abstract structures from the underlying data, and expose them through standard APIs. The Data Virtualization query engine is a required part of the execution environment, to optimally federate data from multiple disparate sources.

6. F

Failover

> Failover is automatic switching to a redundant or standby computer server, system, hardware component or network upon the failure or abnormal termination of the previously active application, server, system, hardware component or network.

7. G

Governance

> Governance uses the hierarchical database for storing additional data and metadata related to Red Hat JBoss Data Virtualization. For example, if a user needs to incorporate information in their JBoss Data Virtualization solution such as images, emails or other semi-structured content, they can store that in the hierarchical database and then access it through JBoss Data Virtualization. Or they can put additional descriptive information about their data in the hierarchical database and combine it with other data.

8. H

9. I

Interactive AdminShell

> The interactive AdminShell provides a command-line interface in which users can issue ad hoc script-based commands for simple administration of JBoss Data Virtualization.

10. J

JBoss Operations Network

JBoss Operations Network (JON) gives administrators a single point of access to view their systems. JON provides a means to develop and monitor a system's inventory. Every managed resource – from platforms to applications to services – is contained and organized in the inventory, no matter how complex the IT environment may be.

JBoss Operations Network centralizes all of its operations in an installed server. The JON server communicates with locally installed JON agents. The agents interact directly with the platform and services to carry out local tasks such as monitoring. The types of resources that can be managed by JON and the operations that can be carried out are determined by the server and agent plug-ins which are loaded in JON.

The relationships between servers, agents, plug-ins, and resources are what define JBoss Operations Network.

11. K

12. L

Load Balancing

Load balancing is a computer networking methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple components with load balancing, instead of a single component, also increases reliability through redundancy.

13. M

Management Command Line Interface

The Management Command Line Interface (CLI) is a command line administration tool for JBoss EAP 6.

Use the Management CLI to start and stop servers, deploy and undeploy applications, configure system settings, and perform other administrative tasks. Operations can be performed in batch mode, allowing multiple tasks to be run as a group.

Management Console

The Management Console is a web-based administration tool for JBoss EAP 6.

Use the Management Console to start and stop servers, deploy and undeploy applications, tune system settings, and make persistent modifications to the server configuration. The Management Console also has the ability to perform administrative tasks, with live notifications when any changes require the server instance to be restarted or reloaded.

In a Managed Domain, server instances and server groups in the same domain can be centrally managed from the Management Console of the domain controller.

Materialized Views

JBoss Data Virtualization supports materialized views. Materialized views are just like other views, but their transformations are pre-computed and stored just like a regular table. When queries are issued against the views through the JBoss Data Virtualization Server, the cached results are used. This saves the cost of accessing all the underlying data sources and re-computing the view transformations each time a query is executed.

Materialized views are appropriate when the underlying data does not change rapidly, or when it is acceptable to retrieve data that is "stale" within some period of time, or when it is preferred for end-user queries to access staged data rather than placing additional query load on operational sources.

Maven

Apache Maven is a distributed build automation tool used in Java application development to build and manage software projects. Maven uses configuration XML files called POM (Project Object Model) to define project properties and manage the build process. POM files describe the project's module and component dependencies, build order, and targets for the resulting project packaging and output. This ensures that projects are built in a correct and uniform manner.

Maven uses repositories to store Java libraries, plug-ins, and other build artifacts. Repositories can be either local or remote. A local repository is a download of artifacts from a remote repository cached on a local machine. A remote repository is any other repository accessed using common protocols, such as **http://** when located on an HTTP server or **file://** when located on a file server.

Modeling Environment

The Modeling Environment is used to define abstraction layers.

14. N

Non-interactive AdminShell

The non-interactive AdminShell allows you to administer Red hat JBoss Data Virtualization by executing previously developed scripts from the command-line. This mode is especially useful to automate testing and to perform repeated configuration/migration changes to a JBoss Data Virtualization instance.

15. O

Open Database Connectivity

Open Database Connectivity (ODBC) is a standard C programming language middleware API for accessing Database Management Systems (DBMSes). The designers of ODBC aimed to make it independent of database systems and operating systems; an application written using ODBC can be ported to other platforms, both on the client and server side, with few changes to the data access code.

16. P

Pages

Pages are units that live in a workspace and provide space (dashboard) for panels. By default, you can display a page by selecting it in the Page dropdown menu in the top panel.

Every page is divided in two main parts: the lateral menu and the central part of the page. The parts are divided further (the exact division is visible when placing a new panel on a page). Note that the lateral menu allows you to insert panels only below each other, while in the central part of the page you can insert panels below each other as well as tab them.

A page also has a customizable header part and logo area.

17. Q

Query Engine

When applications submit queries to a Virtual Database via the access layer, the query engine produces an optimized query plan to provide efficient access to the required physical data sources as determined by the SQL criteria and the mappings between source and view models in the VDB. This query plan dictates processing order to ensure physical data sources are accessed in the most efficient manner.

18. R

Resource Adapters

A resource adapter is a deployable Java EE component that provides communication between a Java EE application and an Enterprise Information System (EIS) using the Java Connector Architecture (JCA) specification. A resource adapter is often provided by EIS vendors to allow easy integration of their products with Java EE applications.

An Enterprise Information System can be any other software system within an organization. Examples include Enterprise Resource Planning (ERP) systems, database systems, e-mail servers and proprietary messaging systems.

A resource adapter is packaged in a Resource Adapter Archive (RAR) file which can be deployed to JBoss EAP 6. A RAR file may also be included in an Enterprise Archive (EAR) deployment.

19. S

Server

The server is positioned between business applications (consumers) and one or more data sources. An enterprise ready, scalable, manageable, runtime for the Query Engine that runs inside JBoss AS that provides additional security, fault-tolerance and administrative features.

Service

A Red Hat JBoss Data Virtualization service is positioned between business applications and one or more data sources, and co-ordinates the integration of those data sources for access by the business applications at runtime.

Source Model

Source Models represent the structure and characteristics of physical data sources and the source model must be associated with a translator and a resource adapter.

20. T

Teiid Designer

Teiid Designer is a plug-in for Red Hat JBoss Developer Studio, providing a graphical user interface to design and test virtual databases. The Teiid Designer produces Relational, XML and Web Service views. It is designed to resolve semantic differences, create virtual data structures at a physical or logical level and use declarative interfaces to integrate, aggregate, and transform the data from source to a target format for application compatability.

Translator

In Red Hat JBoss Data Virtualization, a translator provides an abstraction layer between the query engine and the physical data source. This layer converts query commands into source specific commands and executes them using a resource adapter. The translator also converts the result data that comes from the physical source into the form that the query engine requires.

21. U

22. V

View Model

View Models represent the structure and characteristics you want to expose to your consumers. These view models are used to define a layer of abstraction above the physical layer. This enables information to be presented to consumers as a business model rather than as a representation of how it is physically stored. The views are defined using transformations between models. The business views can be in a variety of forms: relational, XML or Web Services.

Virtual Database

A virtual database (VDB) is a container for components that integrate data from multiple disparate data sources, allowing applications to access and query the data as if it is in a single database and, therefore, using a single uniform API.

A VDB is composed of various data models and configuration information that describes which data sources are to be integrated and how. In particular, source models are used to represent the structure and characteristics of the physical data sources, and view models represent the structure and characteristics of the integrated data exposed to applications.

23. W

WebUI

The WebUI allows building of Data Services through the browser so no local installation is required.

Workspace

A workspace is a container for pages with panels or indicators. Every workspace uses a particular skin and envelope, which define the workspace's graphical properties. By default, the Showcase workspace is available.

24. X

25. Y

26. Z