

The two query modes of IBM Cognos Business Intelligence v10

Introduction

By default, IBM Cognos Framework Manager 10.2.x projects and packages are set to the dynamic query mode, which is powered by an eXtensible Query Engine (XQE) written in Java. XQE, the engine behind the Cognos BI query service and the dynamic query mode, embraces the principles of abstraction and extensibility, allowing it to evolve into a more efficient query planner, processor, and executor with every new version of Cognos BI.

However, Framework Manager projects and packages can be set to the compatible query mode, which, for reasons of compatibility with previous versions, is designed to maintain query and data processing behaviors consistent with version 8.4.1 of Cognos BI.

The C++ query engine of version 8.4.1 addressed the analytical challenges of its day, but it was limited in two fundamental ways. The first is that it is 32-bit, which is a problem because effective caching of enterprise data requires 64-bit addressable memory. The second is that its architecture cannot easily adapt to the new trends emerging in the Big Data era.

XQE was developed to address these limitations and accelerate the improvements to query performance that is delivered in new versions of Cognos BI.

Architecture

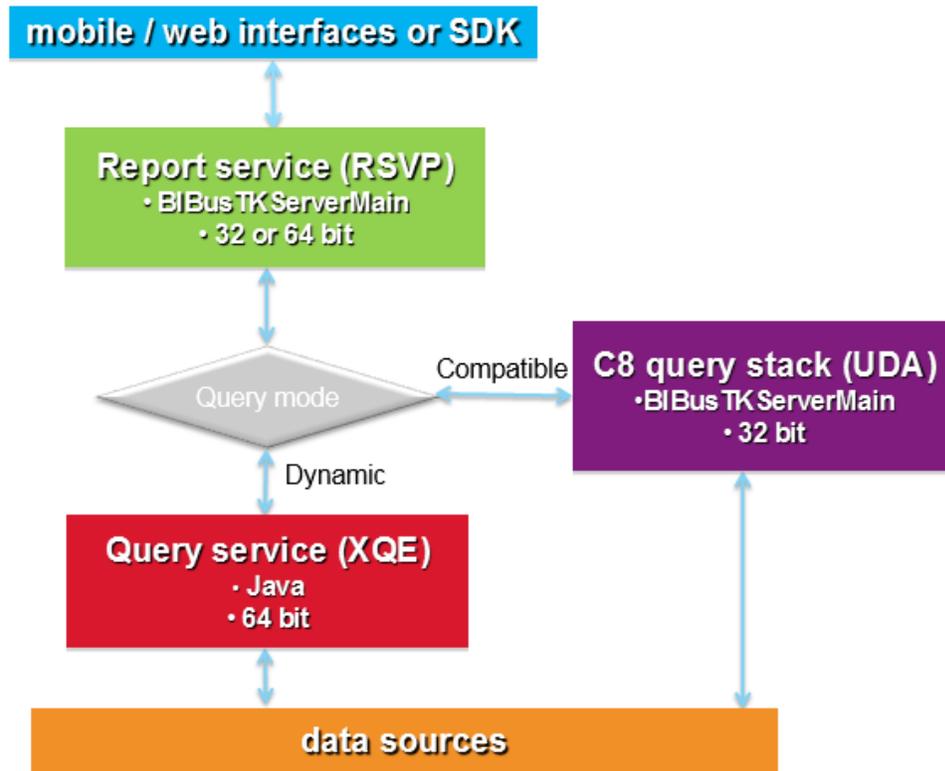
The report service (RSVP) is a rendering engine that accepts XML report specifications from the IBM Cognos BI SDK, web interfaces, and mobile applications, and then generates the desired output which may include visualizations in a pixel perfect layout.

The report service issues data and metadata requests to one of two query engines based on the query mode property of the report's package:

- If the package is set to compatible query mode, then requests are routed to the 32-bit C++ query engine (UDA)
- If the package is set to dynamic query mode, then requests are routed to the 64-bit Java query service (XQE)

The query service and report service operate as independent modules which allows a 64-bit query service to successfully interoperate with either a 32-bit or 64-bit report service. The default configuration of the report service is 32-bit since this configuration supports both the dynamic query mode and the compatible query mode, while a 64-bit report service supports only the dynamic query mode. Note that the report service rarely requires more than 32-bit addressable memory because it typically works with summarized results sets only.

IBM Cognos Business Intelligence 10.x architecture



Statement of Direction

The compatible query mode in Cognos BI version 10 is designed to maintain similar query and data processing with that in version 8. The compatible query mode is not actively being enhanced, but it will continue to be maintained for the foreseeable future including new fixes for defect corrections and added support for some new versions of currently supported data source types.

Reports using the compatible query mode can benefit from new features implemented in other components of the Cognos BI software stack, for example new RAVE visualizations, but do not expect any enhancements to query generation or data processing for the compatible query mode.

There is no intent to deprecate the compatible query mode in the foreseeable future. Relational databases including but not limited to IBM DB2, Oracle, and Teradata will continue to be supported in the compatible query mode for the foreseeable future, however, it has been announced that Cognos BI version 10.2.2 is the last version to support OLAP data sources in the compatible query mode (with the exception of IBM Cognos PowerCubes).

The dynamic query mode is actively being enhanced in new versions of Cognos BI. This includes performance improvements, support for new types of data sources, and new features.

Guidance

It is **highly** preferred to use the dynamic query mode (DQM) for all **new** content.

Enabling DQM on existing BI content is an **optional** migration

- DQM performance may or may not be better
- modifications may or may not be required for any report
- complex reports are more likely to require modifications than simple reports
- DQM enforces certain best practices that are optional in CQM
- DQM will not make certain assumptions that CQM does
- DQM may apply a different order of operations or join path resolution, which can potentially cause changes in a report output

Testing in the IBM Lab and many customer engagements have indicated that the return on investment of migrating a particular set of reports from CQM to DQM is highest for OLAP data sources.

You can use IBM Cognos Lifecycle Manager to gauge the return on investment of migrating a particular set of reports CQM to DQM. Lifecycle Manager allows you to test CQM content as DQM without having to publish the content as DQM.

Additional Information

IBM Cognos Dynamic Query, an IBM Redbooks publication:

<http://www.redbooks.ibm.com/abstracts/sg248121.html>

IBM Cognos Proven Practices: IBM Cognos 10 Dynamic Query Mode Migration Scenarios

http://www.ibm.com/developerworks/data/library/cognos/upgrade_and_migration/bi/page568.html

Lifecycle Manager product documentation on enabling DQM:

http://www-01.ibm.com/support/knowledgecenter/SSEP7J_10.2.2/com.ibm.swg.ba.cognos.ug_upgr_mgr.10.2.2.doc/t_enable_dqm_c10.html%23enable_dqm_c10