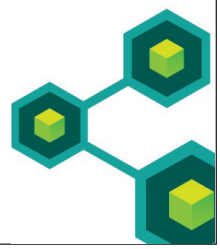
Examine IBM Cognos Services

IBM Cognos BI 10.2.2



Business Analytics software

Business Analytics software

IBM

Objectives

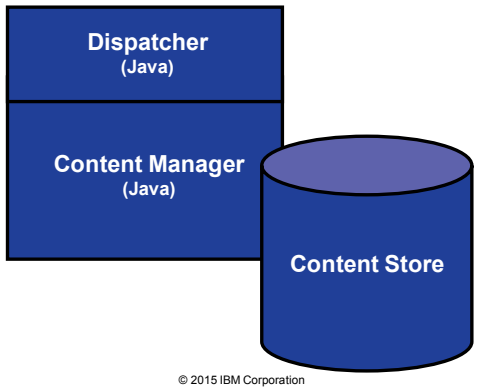
- At the end of this module, you should be able to:
  - identify IBM Cognos Services
  - review collaboration functionality
  - explore the architecture in IBM Cognos 10

Business Analytics software

IBM

What is Content Manager Service?

- implemented by the Content Manager (CM) component
- manages storage of IBM Cognos application data in central repository Content Store (CS)



CM is a Java Servlet and therefore must be deployed to a Servlet Container. If CM is installed separately, the component comes bundled with a dedicated dispatcher which is deliberately configured to not offer the Dispatcher service.

The CM service manages the storage of IBM Cognos application data in a central repository known as the Content Store. Application data contains security (authorization only), configuration data, models, metrics, report specifications, report output, and more. The CM service automatically creates tables and indexes on first startup.

Content Store is connected by JDBC (Type2/Type4). Type 2 JDBC drivers require a local DB client; Type 4 JDBC drivers are standalone and usually only require JAR files. Type 4 JDBC is supported for DB2 in IBM Cognos 10.2.2.

The information stored by Content Manager includes:

- Reports: Specifications, properties, security settings, outputs. This includes analyses in AS, queries in QS, and reports in RS.
- Report Packages: Metadata, reports, folders.
- Metric Packages: Metadata, scorecards, folders.
- Agents: Conditions, schedules, tasks used to monitor events and deliver notifications. This includes the list of recently detected instances of an event.
- Server Configuration: Cognos namespace and information about contacts, directory information, distribution lists, data sources, printers.
- Personal User Information: My Folders and My Pages.
- Language Information: Names, descriptions.

Business Analytics software

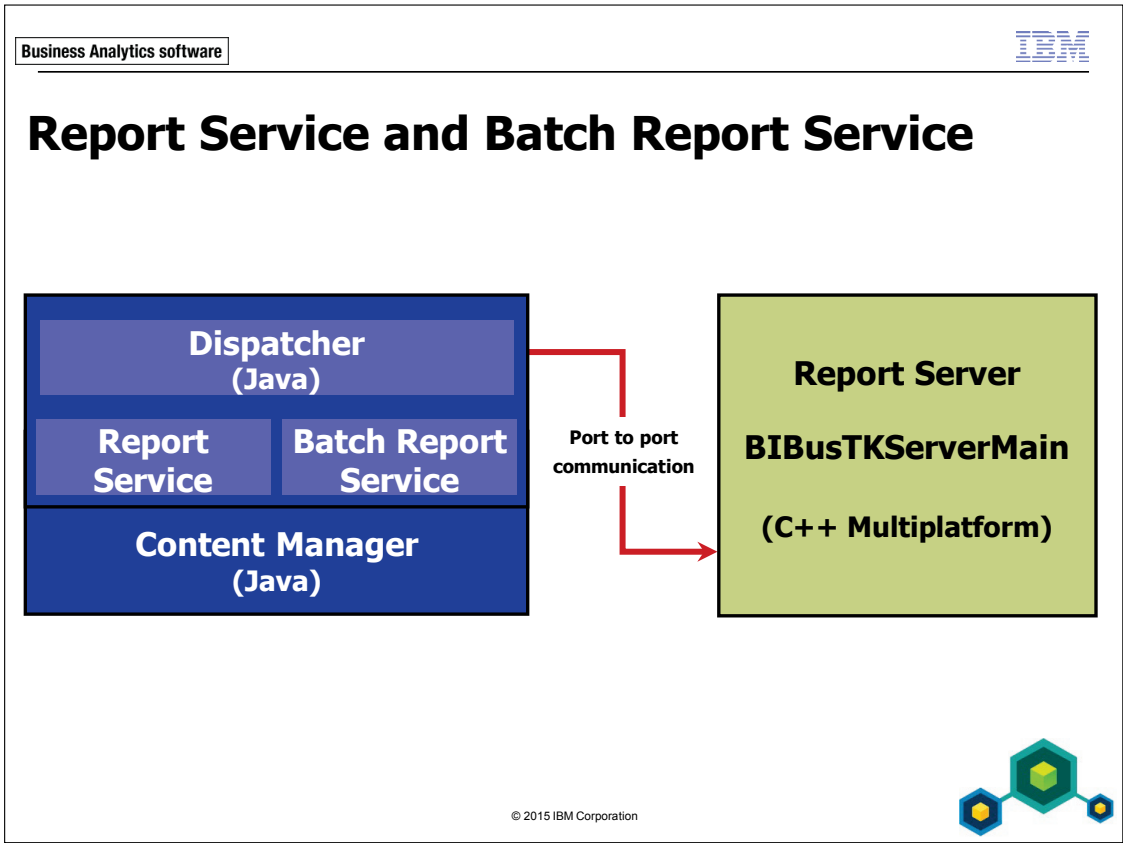
IBM

What Does Content Manager Service Support?

- failover through standby redundancy
  - multiple Content Manager install components installed in single IBM Cognos system
  - on initialization CM registers itself in Content Store database
  - first CM registered becomes the active CM, all others which register subsequently become standby
  - if the active CM becomes unavailable, a standby CM will take over
  - all non-local dispatchers will try to open a socket to the active CM every 5 seconds
  - cluster information update every 30 seconds
  - active CM can be specified using IBM Cognos Administration
- does not support horizontal scaling

Multiple content managers can be installed on the same computer or across different computers. There are many considerations involved in such a decision, factors such as hardware failover and redundancy requirements, number of processors in a machine, and so on.

Horizontal scaling is leveraging multiple systems to work together in parallel with each system has its own operating system and one or more processors. Therefore we can not have multiple primary content managers working at once.



The Report service and Batch report service are implemented by C++ based components which are referred to as report server (functional component).

Report service and Batch report service are two different service registrations which link back to the same implementation in report server. Report service processes interactive report executions, and Batch report service processes report executions triggered by schedules and background requests. They are controlled by settings in the reportservice.xml and batchreportservice.xml files, as well as other configuration files.

Report server instances run inside a separate process managed by the dispatcher. The process name is BIBusTKServerMain, and an instance of BIBusTKServerMain is often referred to as a BIBus. This is different from the BIBus protocol, which is SOAP over HTTP.

Multiple instances of Report server can be spawned by a single dispatcher.

Business Analytics software

IBM

Introduce BIBusTKServerMain

- communication between dispatcher and Report server instances is by TCP port to port connection
- Report server instances
  - can be pre-started
  - get spawned and destroyed dynamically
  - linger; flagged for termination
  - multi-threaded; number of threads is configurable
- Report server threads
  - can be used for internal operation
  - can be dedicated to request processing only
  - called affinity connections

The protocol used for port to port connection is XML-based. A random port from the OS's ephemeral port range is allocated at run-time, for the port-to-port communication. An ephemeral port is a temporary port that can be assigned.

Threads may be distributed among cores in a multi-core CPU. Threads will typically not spawn across CPUs, although exceptions may apply depending on the CPU architecture.

To leverage multiple CPUs, usually multiple processes are required.

Business Analytics software

IBM

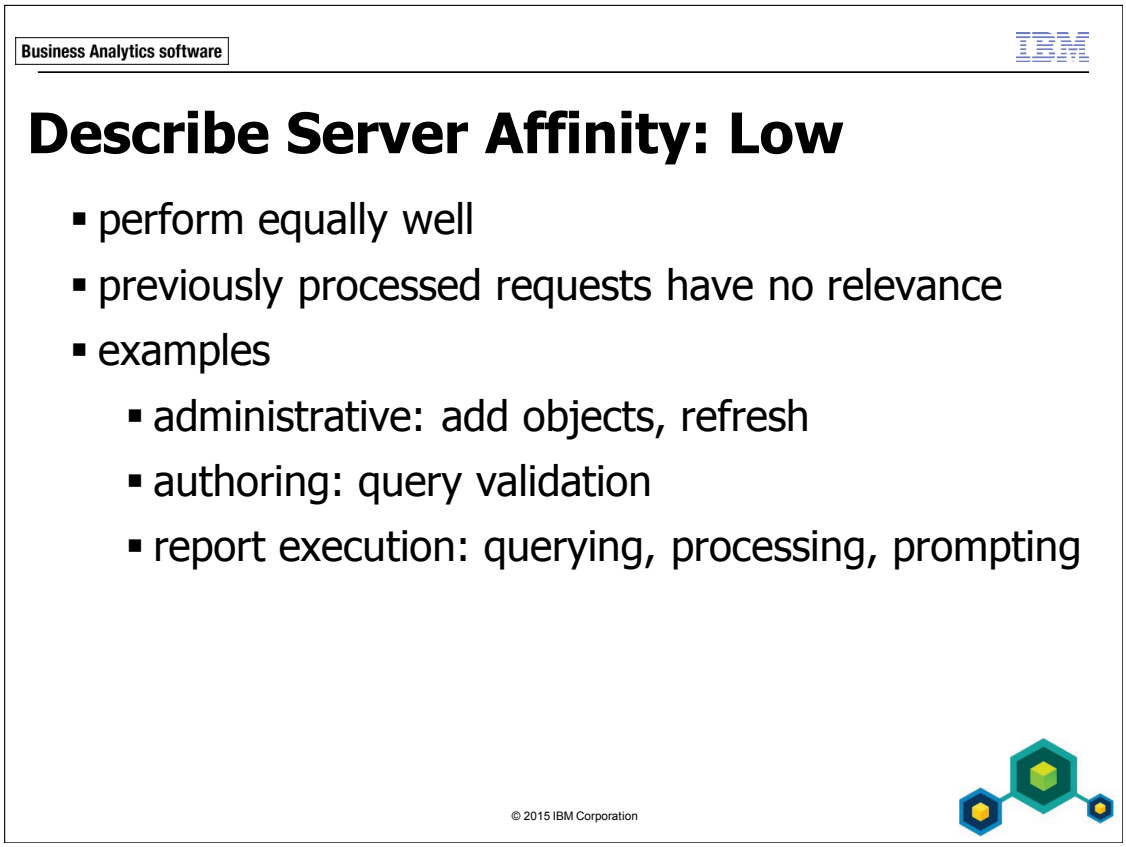
Explain Report Execution in IBM Cognos

- conversation, an operation consisting of multiple steps
- two phases
  - synchronized
  - asynchronous
- each step is a different request
- requests of a conversation have affinity assigned
  - primary request of a conversation: low affinity
  - secondary requests of a conversation: high affinity

Report execution in IBM Cognos is a conversation. Like any other dispatcher conversation, it happens in two phases. In the synchronized phase, the client and server wait for a result for the amount of time specified by the primary wait threshold (the default is 7 seconds) blocking all participating resources. In the asynchronous phase, the request progresses at the server, but the client and the dispatcher only ping for results every now and then (secondary wait threshold, default 30 seconds) and free resources otherwise.

For request routing, the affinity assigned to requests is the basis of routing conversation requests to a specific instance of (Batch)Report Service (request affinity). For the local dispatcher or Report service, this affinity determines to which instance of report server the request has to be assigned (server affinity).

In the context of a conversation, the term server refers to both the target IBM Cognos Service and the dispatcher which routed the request to the service. Both will hold on to a thread for the duration of the primary wait threshold. For more information about conversations, refer to Module 2: Explore the IBM Cognos Dispatcher, on the topic Conversations.



Low affinity requests will operate efficiently on any computer, regardless of the Report server instance they are assigned to. For example, a report request can run on any computer in the IBM Cognos 10 system. A low affinity request is used with the following operations: add, collectParameterValues, execute, getMetadata, getParameters, query, testDataSourceConnection, update, and validate.

Business Analytics software

IBM

Describe Server Affinity: High

- specific requests that can benefit from previously processed requests at this instance
- perform optimally if routed to the same report server instance
- examples
  - report viewing: return, run again
  - HTML report navigation: Top/Bottom page, Page up/down
  - report delivery: E-mail, Save, Save As, Print

IBM Cognos 10 routes high affinity requests to a specific server regardless of the load balancing. A high affinity request is used with the following operations: back, email, firstPage, forward, lastPage, nextPage, previousPage, print, render, save, and saveAs.

Business Analytics software

IBM

Describe Server Affinity: Absolute

- must be routed to a specific instance of report server, otherwise will fail
- can be considered secondary requests which fully depend on a previous request
- examples
  - wait (driving an async conversation)
  - cancel
  - getOutput (retrieve a completed rendered output)
  - release (graceful end of a conversation)

IBM Cognos 10 routes absolute affinity requests to a specific server, regardless of the load balancing. An absolute affinity request is used with the following operations: wait, getOutput, and release.

Business Analytics software

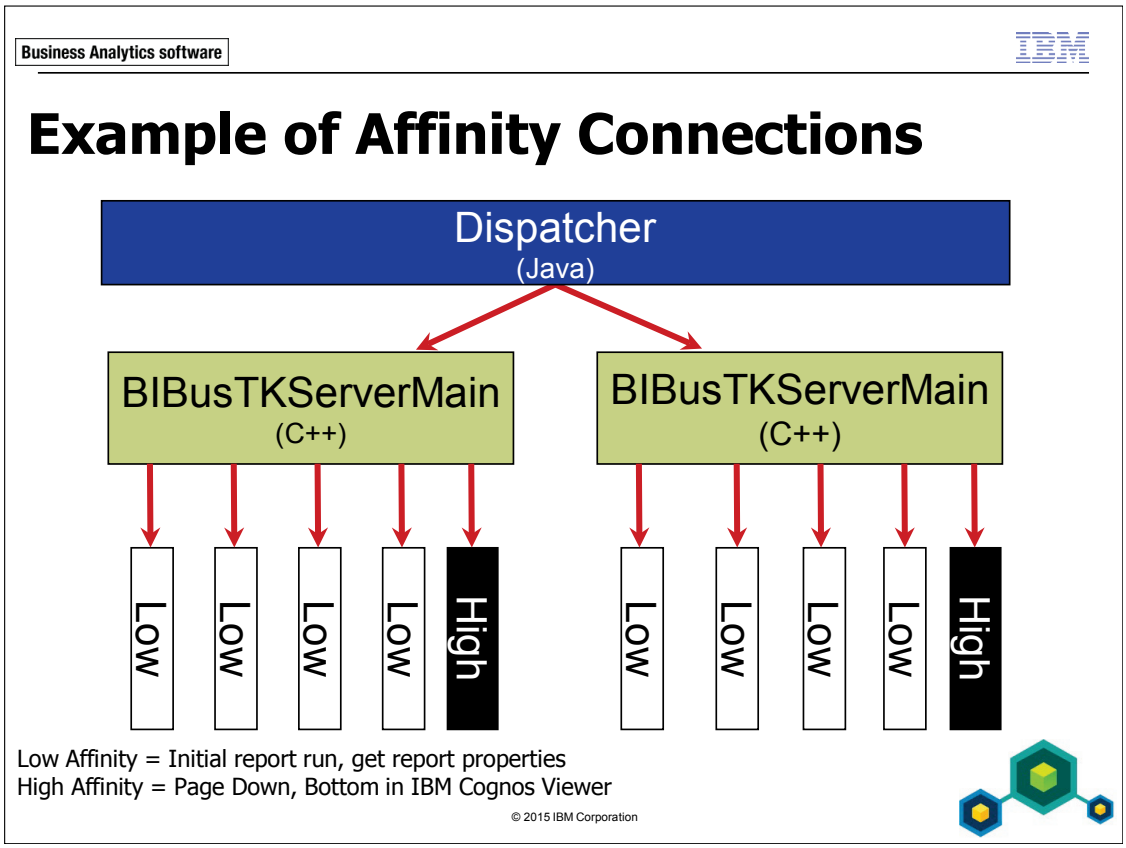
IBM

Describe Affinity Connections

- pooled by request affinity per service instance
- allocation of affinity connections determines the creation of new report server instances
- example
  - by default each report server instance for Report service has 4 low affinity connections, 2 high affinity connections
  - every 3rd low affinity request sent to Report service will trigger a new instance

There are two pools for affinity connections, one for high affinity and one for low affinity, for either Report service and Batch Report service. This means that if both Report service and Batch report service are enabled for an installed instance, then there are four pools. Absolute affinity is handled internally and therefore has no separate pool.

The allocation of affinity connections determines the creation of new Report Server instances. If half the number of the connections of a pool are already allocated, a new instance of Report Server (BIBusTKServerMain) will be spawned for this service if the configured maximum of processes has not been reached yet. The affinity connection of the new instances will get added to the pools thus increasing the sizes of the pool.



There will be many instances of BIBusTKServerMain. Each dispatcher may manage requests for multiple report servers. Low affinity requests can be handled by any report server. Typically, low affinity requests are used when a report or data movement run is initially requested.

High affinity requests are ideally handled by a specific report server. Typically, high affinity requests are for reports that were already requested and may include actions, such as going to the next page in a report.

Business Analytics software

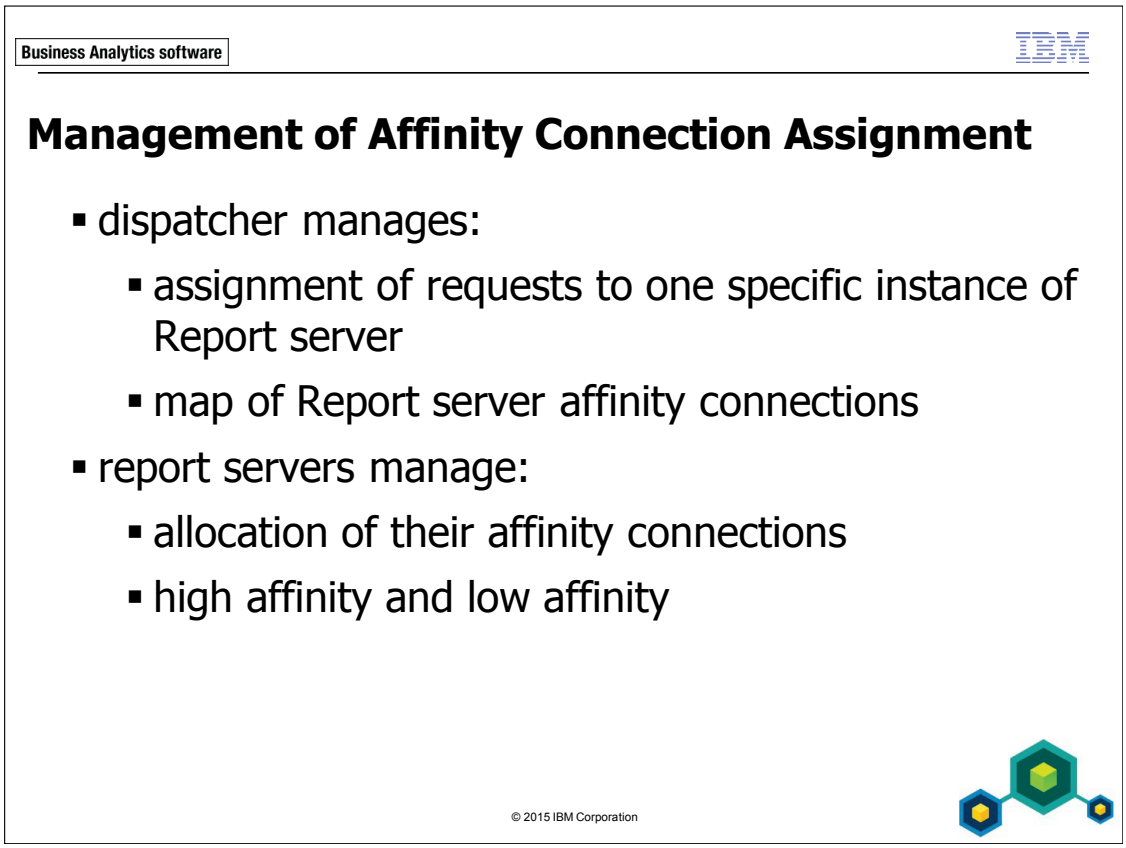
IBM

Describe Affinity Connection Assignment

- eventually report execution requests get assigned to an affinity connection of a report server
- affinity connections remain assigned or unavailable
  - for the duration of the request
  - or the request terminates due to user action or failure

Eventually report execution requests, get assigned to an affinity connection of a report server. Affinity connections do not switch servers in mid-request.

If a user terminates the browser session, cancels a request, or logs out of IBM Cognos 10, the request will terminate.



The dispatcher manages assignment of requests to one specific instance of report server spawned for either Report service or Batch report service, and manages a map of affinity connections on a report server instance, keeping track of assigned connections vs. available connections.

The report servers manage the allocation of their affinity connections, but are otherwise bare of assignment logic.

Business Analytics software

IBM

Explain Affinity Connection Configuration

- configured in IBM Cognos Administration
  - affinity is a tuning setting
  - peak periods and non-peak periods
  - at the dispatcher level
  - at the Report service level or Batch report service level
- must be configured based on solid information
  - monitor the system to learn about load patterns
  - leverage audit database for report usage statistics

The configuration of affinity connections is one of many important settings involved with system performance. It must be configured based on solid information. If this is configured based on uninformed decisions your system can become slow or stop, because of overloading resources. This is not a simple case of more is better.

For more information on leveraging monitoring the system to learn about load patterns, refer to the IBM Cognos System Management Methodology (SMM) on the IBM Web site at http://www.ibm.com/developerworks/data/library/cognos/infrastructure/cognos_specific/page592.html.

Business Analytics software

IBM

Describe Scheduling

- Event Management Service (EMS)
- Monitor Service (MS)
- Delivery Service (DLS)
- Job Service (JS)

Scheduling is handled by several IBM Cognos services.

EMS monitors the global Scheduler Queue and detects when a scheduled task is due to execute. If a background task is due to be run, EMS creates an entry in the global Task Queue using a local JDBC connection to the Notification Database, and executes legacy code which adds a notification about the task to be executed to the JMS queue. Multiple instances of EMS will scale horizontally and can be run independently from other scheduling or target services. Server Groups are irrelevant to EMS.

MS runs background tasks and monitors their execution. It maintains and updates the system global TaskState queue (pending, running, complete, failed). MS uses the capacity mechanism to implement Auto-Load Balancing. It watches the Task queue and takes on tasks if an instance of the required Target Service is available locally and has free capacity. MS will pull tasks from the queue destined for the Server Group it is running in or tasks which have no Server Group assigned.

If there is no instance of a desired Target Service available locally, or there is no capacity the task will not be taken off the Task Queue by this instance of Monitor Service and may remain pending.

DLS sends emails to an external SMTP server on behalf of other services, such as the report service, job service, agent service, or data integration service.

JS handles a set of tasks known as a Job. This service runs jobs by signaling the Monitor service to run job steps in the background. Steps include reports, other jobs, import, exports, and so on.

These scheduling services exchange data through system-global queues stored in relational database tables (the Notification Database):

- Task-state Queue: contains the state of a task passed to MS
- Task Queue: all tasks currently handled by MS
- Scheduler Queue: stores schedule runtime information

Business Analytics software

IBM

Services that Implement the Target Service API (1 of 2)

- services which implement the Target Service API:
  - Batch report service (reports)
  - Job service (jobs, set of reports)
  - Agent service (agents)
  - Delivery service (sending emails)
  - Content Manager service (deployments)
  - IndexUpdate/Data Services (index update/creation)

An API is provided for other IBM Cognos services which want to have their requests enabled for background processing. Services which leverage this API are called Target Services. Examples of Cognos tasks and objects that can be scheduled through the Target Service API are reports, jobs, agents, and email.

Target Services are also required to implement the AsyncAPI so that their requests can be processed in the background through asynchronous conversations. The AsyncAPI defines conversations.

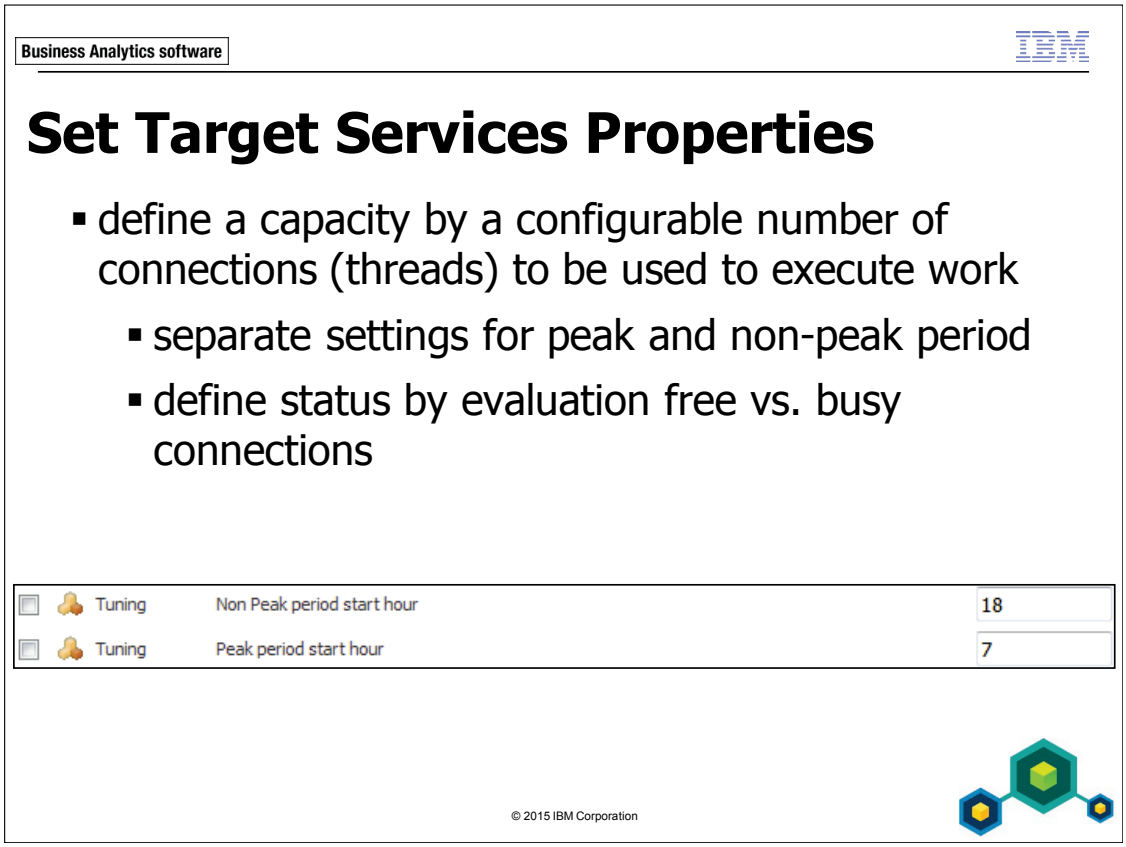
Business Analytics software

IBM

Services that Implement the Target Service API (2 of 2)

- Job Service (JS)
  - governs the execution of jobs
  - splits the job into steps, passes them to the MS for execution
- Delivery Service (DS)
  - builds SMTP compliant emails from data passed, and passes to a configured mail server
- Agent Service (AS)
  - runs agent actions

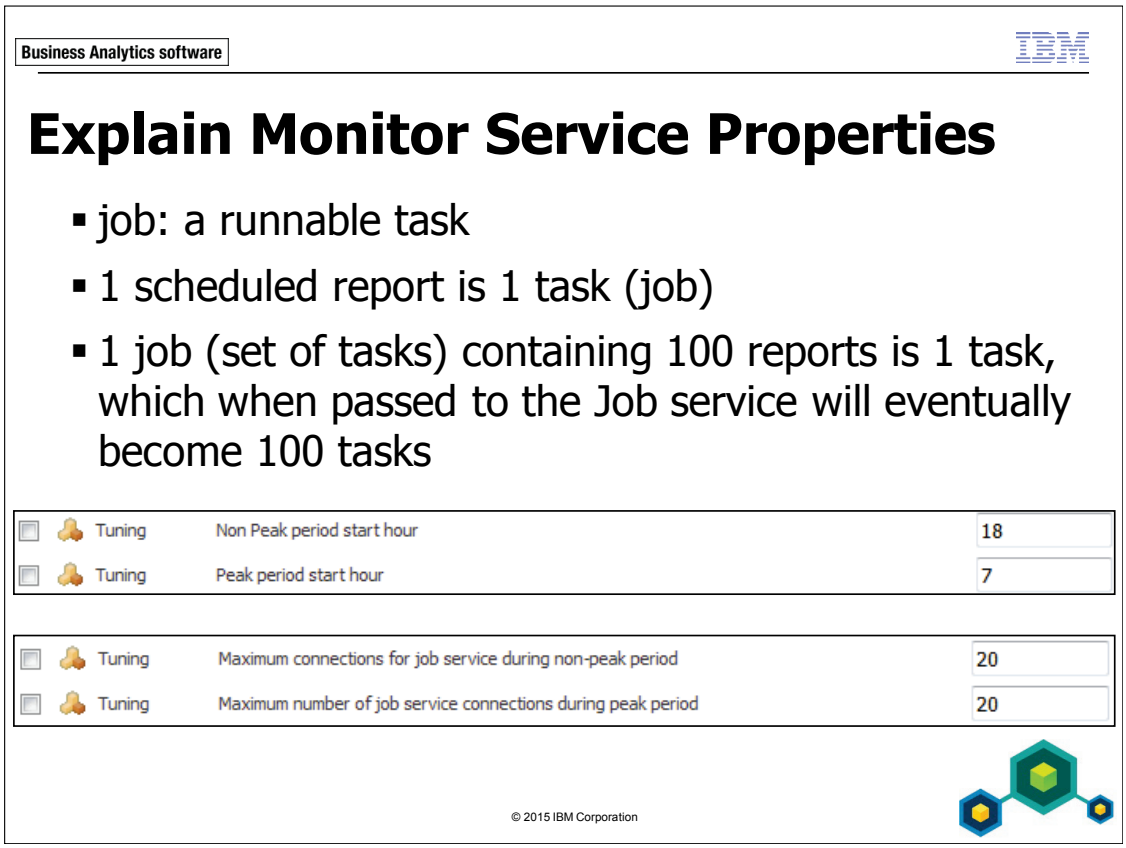
A job is defined as set of tasks, although a task can be another job.



Target Services properties are configurable as Dispatcher properties, and settings can be configured through IBM Cognos Administration at the service level.

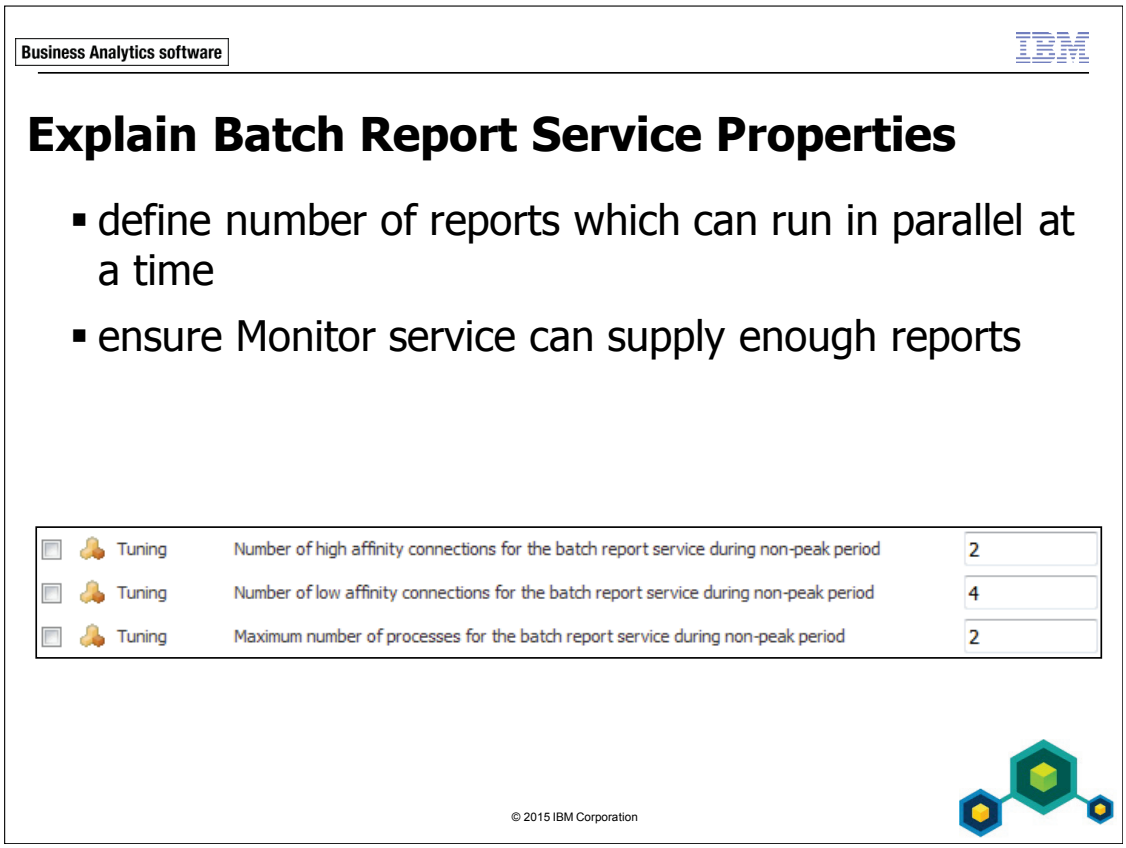
The Target Services require an instance of MS to be assigned work. If separated from MS, they will not be assigned any load for background processing.

A 24 hour day is split into two periods: Peak and Non-Peak.



Administrators can set the start of a peak period and the start of a non-peak period.

This allows configuration of the number of Monitor Runners, or connections, for peak and non-peak periods.



The Batch report service (BRS) executes reports. Administrators can configure the affinity connections and number of processes for the BRS. This effectively defines the number of reports which can run in parallel at a time for the BRS.

For optimal performance make sure the MS can supply enough reports, which is the number of Jobs (Monitor Runners) as specified on the previous page.

In this example, BRS would be able to run 4 reports at a time, of 2 processes with 2 low affinity connections each. The high affinity connections would only be used for secondary requests.

Refer to the Describe Server Affinity: High topic in this module. High affinity could be tuned for requests such as email, print, save.

A good analogy might be to imagine a sink into which water pours from a faucet. If the drain is too small, the sink will overflow. If the drain is too big, the sink will run dry and the faucet cannot provide enough water.

The Monitor Runners are the faucet, the number of batch processes and their affinity connections define the drain size.

If the drain is too small: Too many Monitor Runners and the requests that they send to BRS are backing up on the dispatcher queue.

If the drain is too big: Too many reports get executed at the same time, overloading the available resource.

Business Analytics software

IBM

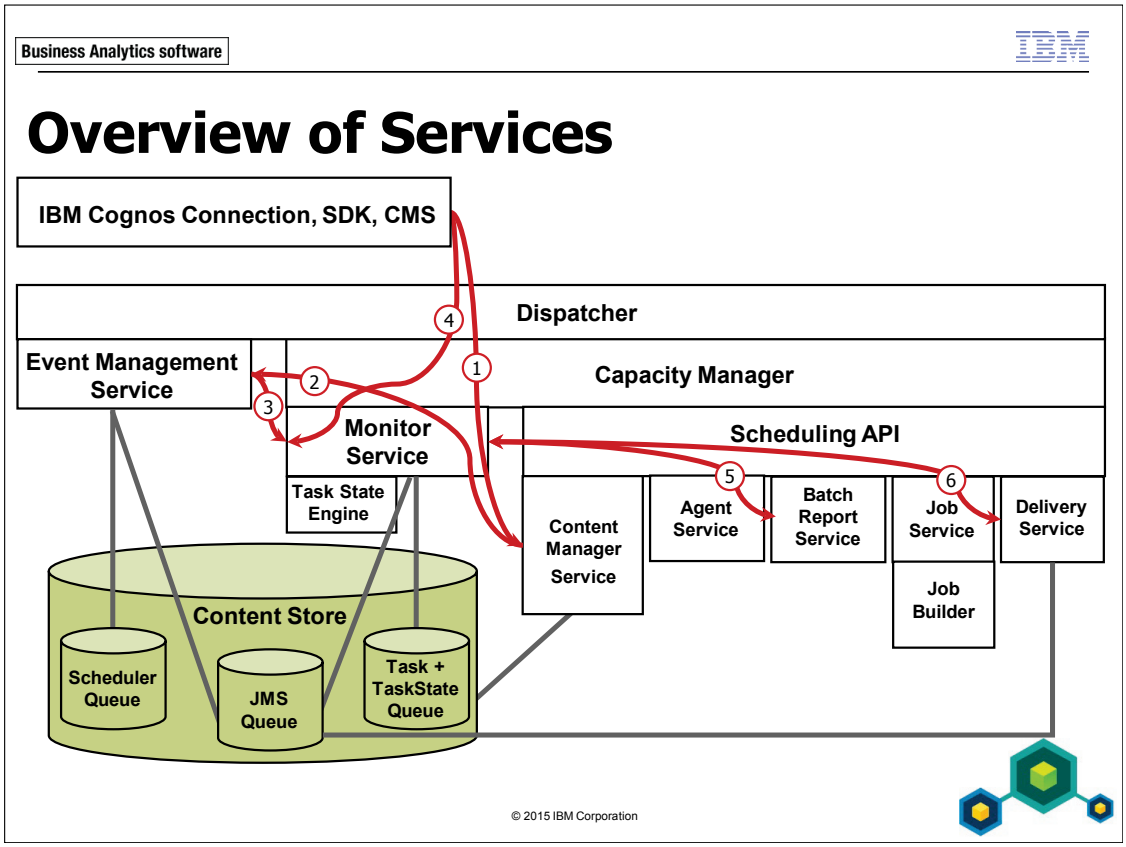
Explain Content Manager Service Properties

- Capacity is defined by a maximum number of connections for ContentManager Service (CM Service)
  - defines the number of deployment/ ContentManager maintenance tasks this particular CM Service instance can handle in parallel, along with all other duties of CM Service

Category	Name	Value	Acquired
Environment	Advanced settings	Edit...	Yes
Logging	Audit logging level for Content Manager service	Minimal	Yes
Tuning	Maximum connections for Content Manager service during non-peak period	4	Yes
Tuning	Maximum number of Content Manager service connections during peak period	4	Yes

The CM Service runs deployments and Content Manager maintenance tasks.

Since there is only one active instance of CM Service, the capacity should be defined cautiously.



The process flow for scheduled tasks is as follows:

1. A user creates a schedule, either through IBM Cognos Connection or some other client. A scheduler object is created in the Content Store by sending a create request to Content Manager Service.
2. The Content Manager Service calls the Event Management Service, replicating the scheduler objects from the Content Store into the Scheduler queue.
3. Each instance of EMS runs a scheduler thread which polls the Scheduler queue. By design, EMS figures out the next runtime of a task once it is added to the queue and waits until due time. Once it wakes up, it starts a transaction to pull the task from the Scheduler queue and move it to the Task queue through the Monitor service. The task will be entered into the Task queue in a PENDING state.
4. Another scenario where tasks are created and passed to MS is where a user selects "Run now" for saved reports (run and save). IBM Cognos Connection will call MS directly, which will lead to the task being queued in PENDING state.

5. MS has a poll interval (30s hardcoded) after which it will investigate for tasks to take on. MS will only take on tasks for locally available Target Services for this particular server group MS runs in which have free capacity. Capacity information for the local target services is gathered through the Capacity Mechanism.

   In addition, any call to the MS will trigger an investigation of the TaskQueue. A third trigger for an MS instance to investigate the TaskQueue is the legacy JMS queue notification being received.

   Once MS picks up a task it is granted that there is a target service with free capacity to assign this task to.

   MS checks for an authenticated session. For scheduled tasks it will create a new session leveraging the trusted credentials saved with the schedule. For "saved reports" the credentials for the current session are used.

   Next the task status is updated to EXECUTING and the task is assigned to the target service which uses the AsyncAPI to execute the task.

   The Monitor Service updates the task state based on the response received from the target service. If a cancel was issued in between the processing ends here and the task is cancelled. The result can be succeeded or failed.
6. If the Report has email output then BRS employs DLS to add an item to the JMS SMTP Queue. This queue is worked by all DLS instances in a system. A DLS instance will pick up the item and pass it to a configured SMTP server after which the runtime state of a task is updated finally.

   Upon completion of the task execution, the Monitor service will write out the history records from its internal store to Content Manager. It keeps the history internally when tasks fail, in case the user wants to restart any failed tasks. Periodically the history records for failed tasks are removed, based on the retention rules stored for that task in Content Manager.

Additional Information:

- JMS: Java Message Service.
- JMS queue: Where scheduled items are stored. The JMS queue is internal but makes use of the data in the NC table NC_JMSQUEUE.
- JMS SMTP queue: This is an internal queue that sits in memory and cannot be seen or manipulated like the JMS queue.
- SMTP: Simple Mail Transfer Protocol.
- EMS has a queue: jmsqueue. The EMS thread which picks the runnables which need to execute now puts them into this queue. At this moment, the activities/runnables are invisible to the user and administrators; they are neither pending nor executing.
- MS has a queue: monitor queue. This queue contains all the activities which are to be executed by monitor runners. At this point they are pending. From this queue, a monitor runner will pick up an activity, which at that time becomes executing.

Business Analytics software

IBM

Service Oriented Architecture Implications

- Target Services will not be assigned tasks if they are separated from Monitor Service (MS)
  - without MS, deployments and Content Manager maintenance tasks will not work
  - there must be at least one instance of MS next to CM
- MS is lightweight with regards to performance; more instances can help overall performance.
- all installed instances running Target Services must run Monitor Service

Server Groups are fully respected; each Server Group requires at least one instance of MS and desired Target Services on the same install.

Target Services can be separated from each other as long as an instance of MS is available next to them, and they remain in the same Server Group.

Example of a single Server Group scenario:

- Dispatcher 1: MS+JS
- Dispatcher 2: MS+DLS
- Dispatcher 3: MS+BRS+EVS

Business Analytics software

IBM

Explain Cognos Graphics Service

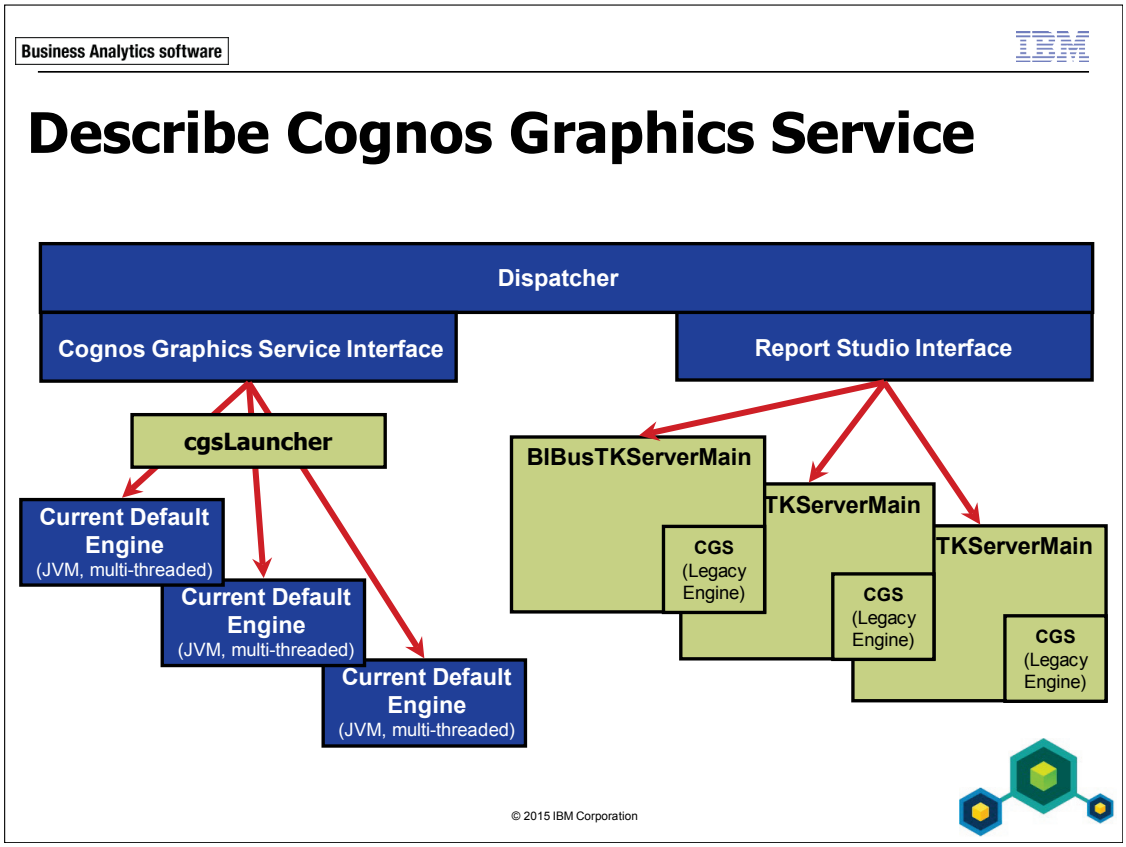
- Legacy Engine
  - runs in-process as part of BIBusTKServerMain
  - used by Analysis Studio and Query Studio
  - can be used by Report Studio
- Current Default Engine
  - runs in own JVM as a separate process
  - used by Cognos Workspace
  - used by Report Studio by default

IBM Cognos 10 offers two alternate graphic engines to use, the legacy engine and the current default engine.

Analysis Studio and Query Studio can only use the legacy engine.

Report Studio can use the legacy engine, but you must specify this in Report Studio Options, Advanced.

Cognos Workspace does not have the option of using the legacy engine.



The current default engine is very much like ReportService. The Cognos Graphics Service (CGS) interface consists of handlers registered to the dispatcher. These handlers are partially using the ones from BIBusTKServerMain, which you are now familiar with the concept.

CGSLauncher is used to spawn JVMs which host the current default engine. In Report service the dispatcher invokes BIBusTKServerMain directly, however for CGS there is some additional redirection. In Windows environments, cgslauncher.exe is called which will call a JVM, and in non-Windows environments cgsServer.sh script is called when needed.

The current default engine will use a configurable number of threads (number of low affinity connections from Service properties).

The Current Default Engine Java process will not be recognizable on *IX, but will be recognizable only by the command which invoked it. On Windows, the cgsLauncher.exe in Task Manager is, in fact, the JRE.

The Current Default Engine Java processes will terminate if idle for 5 minutes (configurable in cgsService.xml).

Business Analytics software

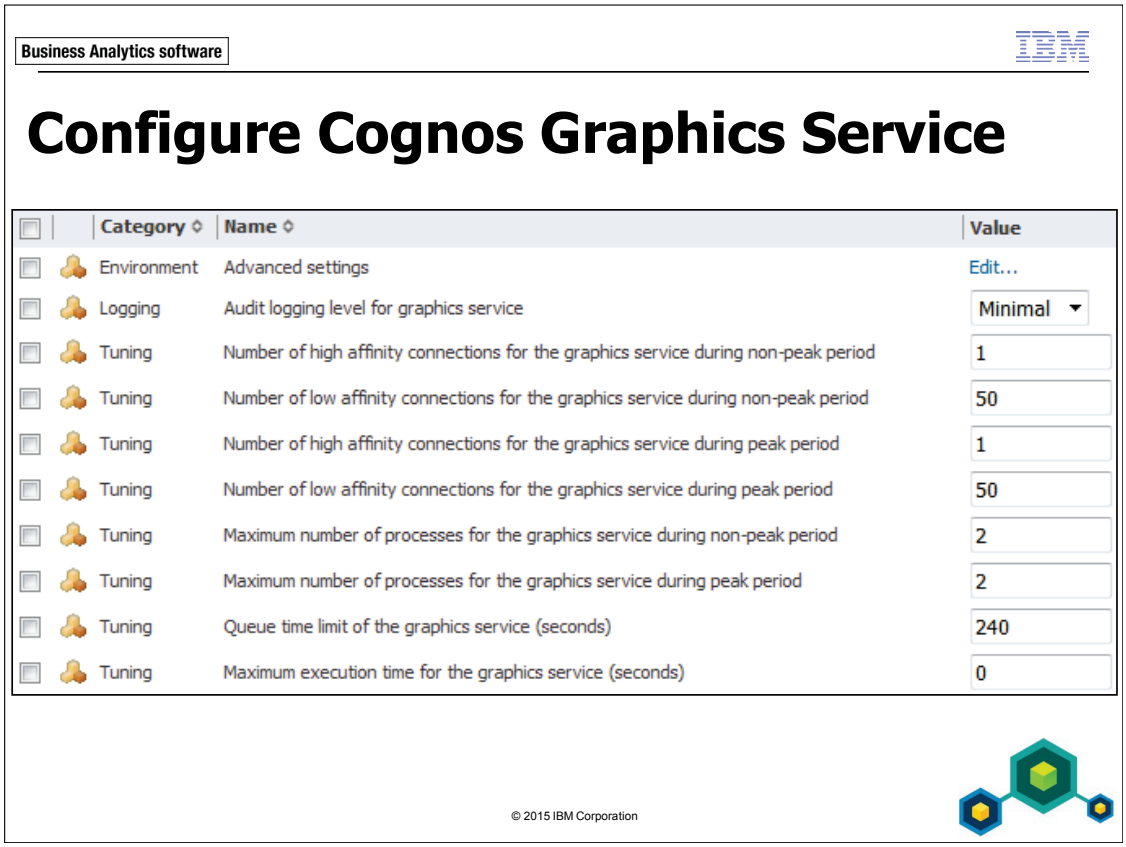
IBM

Describe the Current Default Engine

- invoked through cgsLauncher (cgsLauncher.exe, cgsServer.sh)
  - uses BIBusTKServerMain
- cgsLauncher will start configured number of processes for CGS
  - new instances will get spawned until configured limit for number of processes is reached
  - processes are JVMs
  - JVM arguments can be edited
  - CGS JVMs will use configured number of threads

The current default engine is written in Java, and invoked through cgsLauncher, which is a C++ process that starts when the product starts. cgsLauncher uses the BIBusTKServerMain interface, which is the same interface as Report server, and is controlled by the dispatcher. There is port to port communication through the dispatcher, and it will terminate if it is idle too long (the default is set to 5 minutes).

JVM runs within JRE, and JRE runs within JDK. JVM and JRE sometimes used synonymously.



Requests for Cognos Graphics Service (CGS) are conversations, which is to say that they can go asynchronous and this implies they are high affinity requests. These requests are subject to queuing.

CGS can be configured in IBM Cognos Administration, and the properties are similar to ReportService. Initial requests to CGS are subject to load balancing and advanced routing because they are not affine.

Business Analytics software

IBM

Configure CGS Parameters

- cgsService.xml has several parameters as JVM arguments
  - linger_process
  - prestart_process
  - idle_process_check_interval_ms
  - idle_process_max_idle_ticks
  - report_server_port (should be 0 for dynamic ports)
  - server_port (should be 0 for dynamic ports)

The parameters are in ..\webapps\p2pd\WEB-INF\services\cgsService.xml. (.. is the install directory for IBM Cognos 10.2.2).

Since this is Java there are many tools available to hook to the JRE. Use Java monitoring on the current default engine processes to monitor performance.

- **linger_process**: The minimum number of processes to keep in memory once they're started.
- **prestart_process:** Should the CGSServer be started at initialization time rather than on-demand?
- **idle_process_check_interval_ms:** Time period between checks for idle or expired processes (in ms). Every time we check an idle process we bump its tick count. Every time we use a process the count is reset to 0. If the count ever reaches the value of idle_process_max_idle_ticks (below) the process is destroyed.
- **idle_process_max_idle_ticks:** number of idle check ticks before an idle process is killed.
- **report_server_port and server_port:** The port the CGSServer is listening to. 0 (zero) should be used if the dispatcher spawns the process and uses a dynamically assigned port.

Business Analytics software

IBM

Explain Interactive Discovery Visualization (IDViz) Service

- As of version 10.2.0, IBM Cognos BI supports a "Full Fidelity Publish" (FFP) of an IBM Cognos Insight dashboard to an IBM Cognos BI Workspace.
- The Explore Points from Cognos Insight will become Facet Controls in IBM Cognos Workspace.

IBM Cognos Workspace is the new name for BUX-E (formerly named IBM Business Workspace/IBM Business Insight, which was initially named GO Dashboard when this functionality became included into BI by default as of BI 10.1.0).

The FFP is basically deploying a TM1 data source and a workspace to BI. The dynamic elements of the Cognos Insight dashboard need to be transformed into BI actionable workspace objects as Facets.

The Facet Controls are embedded into Workspace Widgets which display dynamic content retrieved from a content provider in the back-end. Often, the content provider is remote.

In case of an FFP, the content provider is a TM1 data source which is created as part of the publish action to a Workspace.

The Facet Controls communicate with a back-end Facet Service. This is not a service as defined in the IBM Cognos BI context, but rather the IDViz Service is the IBM Cognos Service interface for the Facet Service.

Facet Controls send requests to BI asking for the IDViz Service to provide content. This scales horizontally, as requests are routed by Dispatcher. Requests are not subject advanced routing.

The small resource footprint pulls data from TM1 and serves it up.

IDViz supports IPF logging, and can be disabled if nothing is published to this BI system from Cognos Insight.

If FFP from IBM Cognos Insight is used with a BI system, there should be at least two instances of the IDViz Service in a system to allow fail-over.

Business Analytics software

IBM

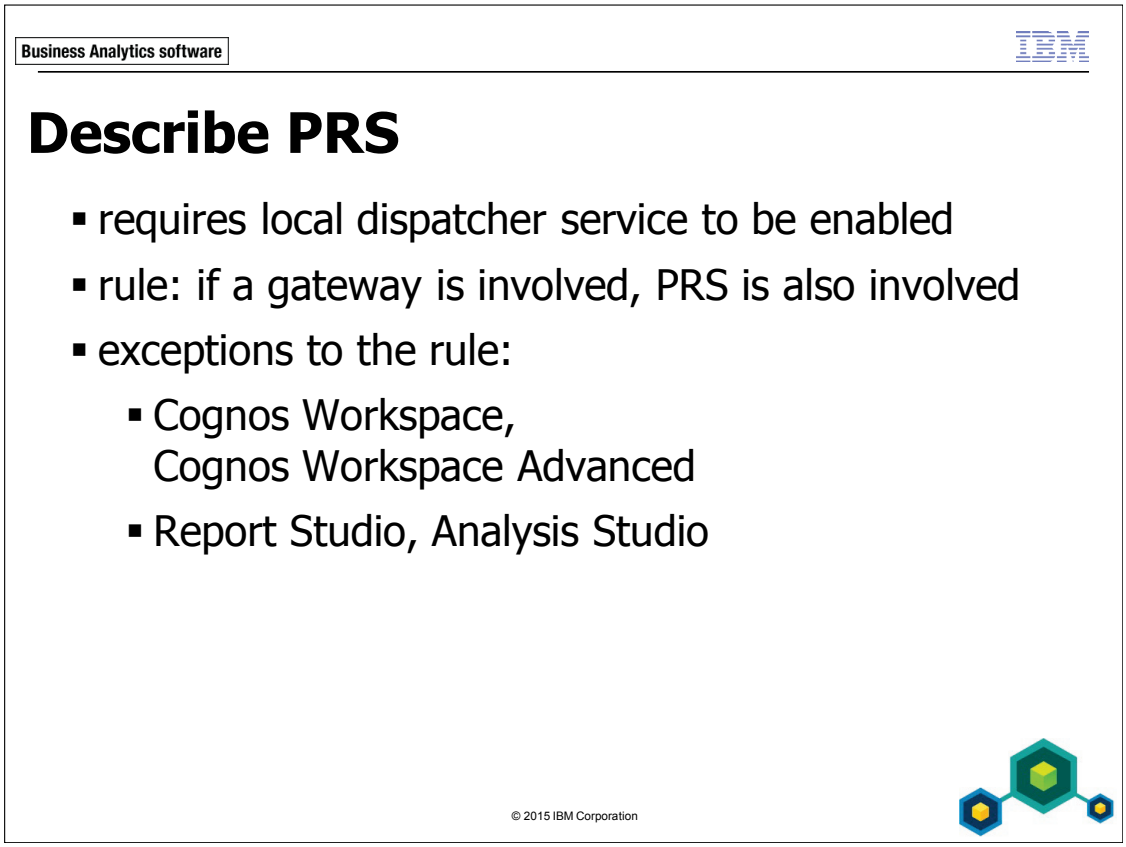
Explain Presentation Service (PRS)

- uses underlying components
  - XTS (generate pages based on XML style templates)
  - Fragments (XML based composition of pages)
- generates HTML output to be viewed in browser client
- proxies requests for other services
  - Report Data Service (RDS): ATOM Feeds
  - Cognos Portal Services (CPS): WSDL, WSRP

PRS uses several underlying components to support different techniques for page generation, like XTS (XML Transformation Service) and Fragments.

PRS generates HTML output to be viewed in browser client for IBM Cognos Connection, Query Studio, Event Studio, IBM Cognos Administration, and IBM Cognos Viewer. XTS is not Event Studio or IBM Cognos Viewer, but those UIs will send back requests to XTS, which will then employ the underlying engines to generate the output for those UIs.

Presentation Service is the central IBM Cognos Service when interacting with browser clients. SDK clients would not require XTS necessarily. XTS is not Presentation Service, but is a server-based service component used to generate HTML pages in IBM Cognos Connection. Sometimes XTS and PRS get used interchangeably which is incorrect. XTS is based on XML and XSLT.



PRS is mandatory for front dispatchers, particularly if Single Sign-On (SSO) is configured. This is because XTS is required when handling User and System Recoverable parameters which are part of the process to render the HTML pages for login and to select a namespace.

PRS is unavailable on CM only installs, as PRS requires the local dispatcher service to be enabled. Therefore a basic rule is that if a gateway is involved, then PRS is also involved. There are exceptions to this rule, as listed in the slide.

Cognos Workspace and Cognos Workspace Advanced are rendered by a different engine (Cognos Workspace is Current default engine only) in a different service (BUX).

Report Studio and Analysis Studio are rendered by AJAX based techniques. AJAX is a design approach and a set of techniques for delivering rich Internet applications (RIAs) using open web formats, for example, HTML, CSS and JavaScript; and rendering using a browser engine.

Business Analytics software

IBM

Explain Report Data Service (RDS)

- RDS exposes IBM Cognos content in Web 2.0 formats
- supports REST and SOAP interfaces
- data can be retrieved as
  - XML (Simple/layout data)
  - HTML including Fragments
  - JSON

RDS is implemented by a Development component called Cognos Content Service (CCS). The API is a separate product known as Cognos Mashup Services (CMS).

RDS exposes IBM Cognos content (reports, workspaces, analyses) in Web 2.0 formats for mashing it up with other applications data.

RDS makes use of Report service, Metric service, PowerPlay service, and Query service.

Representational State Transfer (REST): A software architectural style for distributed hypermedia systems like the World Wide Web. The term is also often used to describe any simple interface that uses XML (or YAML, JSON, plain text) over HTTP without an additional messaging layer such as SOAP.

Business Analytics software

IBM

What Components Use RDS?

- IBM Cognos Workspace
- IBM Cognos for Microsoft Excel
- IBM Cognos Mobile
- IBM Cognos for Microsoft Office

Business Analytics software

IBM

What is Metadata Service (MDS)?

- implemented in C++, runs in separate executable
  - BmtMDProviderMain
- based on the BIBusTKServerMain model
  - has affinity connections (4 low only)
  - a single instance is spawned upon startup by dispatcher
  - terminates if idle for 5 minutes, restarted as required

MDS serves metadata requests for Query, Update, Test data base connections, lineage and more. Theoretically MDS can take up the maximum allowed process space but usually remains small in memory, smaller than Report server instances.

Business Analytics software

IBM

Explain Relational Metadata Service (RMDS)

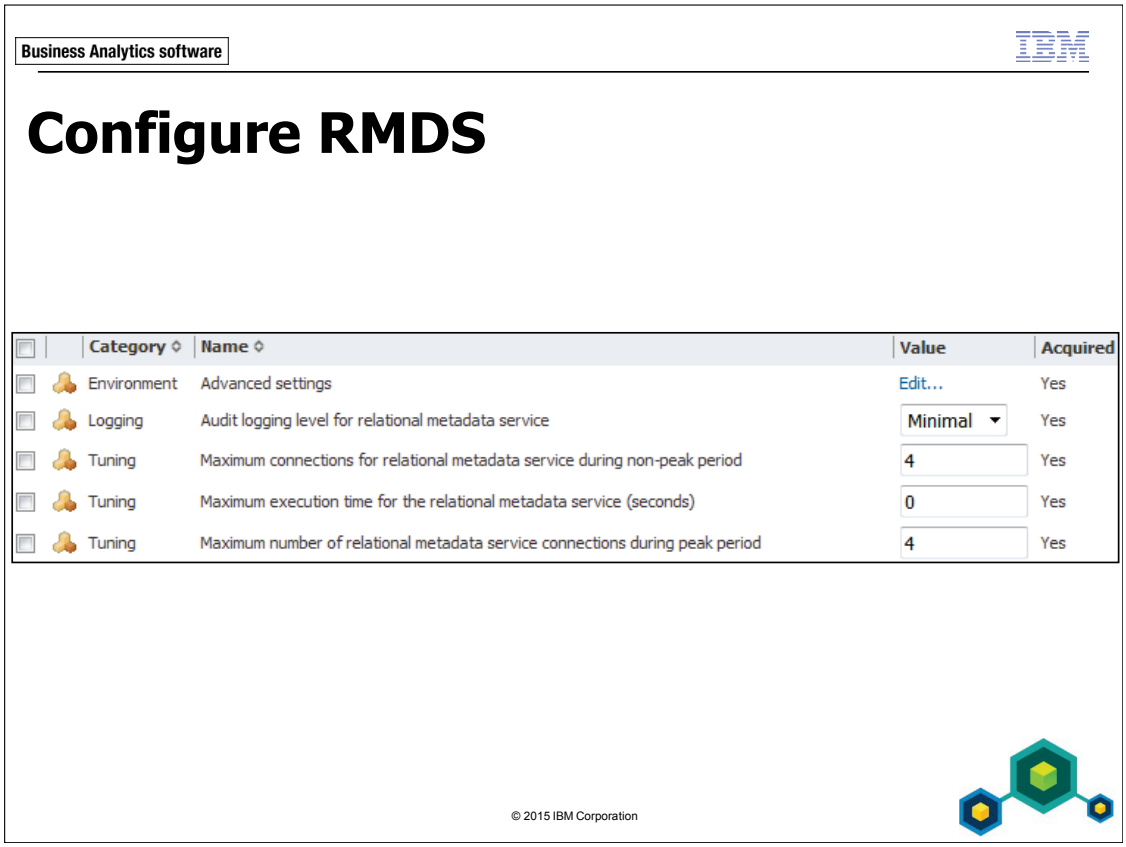
- RMDS interfaces to the component of the DQM engine (RELMD) which connects to data sources via JDBC and provides metadata.
- Used exclusively by:
  - Framework Manager in "dynamic" mode
  - IBM Cognos Cube Designer (CD)
- Queries run on the server, not on the client.
  - No local DB client install is required for installs of FM or CD
  - A JDBC driver is required on Dispatchers running RMDS.
  - The data source requires JDBC connectivity only.
- Use is governed by the "Import relational metadata" capability.

This is a new service as of version 10.2.0 which is logically an interface to RELMD, a component shared by XQE that retrieves metadata from DQM data sources via JDBC. The service is used exclusively by CD and FM.

Using RELMD is quite different from CQM modeling.

- FM in mixed mode uses a local (client) UDA stack and therefore requires DB clients to be installed locally.
- Dynamic mode uses the RMDS which runs the queries on the server:
  - no local (client) DB clients are required
  - requires JDBC connectivity only, running on the Application Tier hosting RMDS
  - requires JDBC driver, although QueryService might be disabled on the same node

The new Import relational metadata capability controls who can pull metadata. The capability does not control availability or start of FM or CD.



With RMDS, there is no horizontal scaling and routing, or fail-over. RMDS is required on Dispatchers referenced in IBM Cognos Configuration for FM and IBM Cognos Cube Designer installations. RMDS does not currently support IPF logging.

RMDS does not require Query Service, and RMDS properties are different than the Metadata Service properties:

- RMDS handles asynchronous conversations
- has batch processing capacity (# connections)

No separate process is being spawned, as work is done in a JVM context of the Dispatcher.

Each instance of RDMS is stand-alone. It uses a local disk cache (../temp directory) and handles requests sent to it through the External Dispatcher URL.

Metadata Service is separate from RDMS and is still required in a system. Metadata Service is required in a system for serving FM mixed mode.

The number of low affinity connections determines how many clients can be served in parallel. Additional requests will possibly be queued at Dispatcher and eventually time out.

RMDS is required on Dispatchers that are referred to by IBM Cognos Cube Designer and FM configurations, nowhere else. RMDS does not require Query Service to be enabled.

Business Analytics software

IBM

What is Dynamic Query Mode?

- Dynamic Query Mode (DQM) is an enhanced Java based query mode which offers the following key capabilities:
  - query optimizations
  - performance improvement through balanced local query processing facilities
  - security-aware caching
  - ability to take full advantage of a 64-bit environment
  - ease of maintenance with query visualization

Query Optimizations: The optimization of the queries is achieved through the advanced application of strict query planning rules. These planning rules incorporate the next generation planning approach which is more streamlined and produces higher quality and faster-to-execute queries. The query planning process is also in itself optimized to make better use of metadata and expression level caches, including plan caches which provide higher application throughput.

Performance Improvement through Balanced Local Processing Facilities: The Dynamic Query Mode makes intelligent, rules based and system load based decisions on which parts of a query should be executed locally in the application server versus remotely in the database server. This ensures that users have the highest functionality possible regardless of whether the underlying data source supports the business intelligence report intent. In addition, the Dynamic Query Mode contains a fine grained metadata and cell data cache which is trickle fed and a higher cache hit ratio than was previously possible. In addition the queries which are sent to remote data sources are further optimised by the execution layer based on cache content and advanced null suppression logic.

Security-Aware Caching: The caching logic available in Dynamic Query Mode is able, when connected to secured metadata sources, to determine the secured access capabilities of each user as they access the data source. This information is then used to optimize the memory usage and internal representation of that user's secured view of the data source metadata. Security can also be setup so that entire OLAP dimensions can be shared providing cache reuse and performance gains.

Ability to take Full Advantage of a 64-bit Environment: The Dynamic Query Mode is able to fully take advantage of a 64-bit environment. When an IBM Cognos 10 64-bit install is deployed to a 64-bit Java Virtual Machine (JVM), the Dynamic Query Mode is substantiated in its own 64-bit JVM. This allows DQM to leverage the 64-bit address space for query processing, metadata caching and data caching. The 64-bit DQM also has the ability to leverage the 64-bit data source client libraries when performing non-JDBC Type 4 connections.

Ease of Maintenance with Query Visualization through IBM Cognos Dynamic Query Analyzer: Query visualization allows system administrators to analyse the queries generated by the Dynamic Query mode and visually see how they will be processed. These visualizations include cost based information derived from the query execution. This information permits the rapid identification of model and query optimizations which could be applied in order to achieve better performance. The visualizations are consumed as a log file by the stand alone IBM Cognos Query Analyzer application.

Business Analytics software

IBM

OLAP Functionality for Dimensionally Modeled Relational (DMR) Packages

- DQM provides users with a true OLAP experience over relational data
- improved ad-hoc analysis capabilities through
  - default member sorting
  - improved suppression
  - aggregate limitations removed
  - nulls as zeros in arithmetic operations
  - FIRST / LAST aggregate rules respect presence of NULL values

DQM applies advanced OLAP caching techniques to enhance performance of dimensionally modeled relational packages. The use of caching reduces the frequency of database queries, thus minimizing the database server workload required to service the IBM Cognos application.

With OLAP over Relational (ROLAP), DQM specifies a natural order to all result sets. If no specific ordering is specified in the Framework Manager model, the members are sorted by default in ascending order by the member caption. If there are duplicate captions, they those are sorted by business key.

With DQM, suppression is pushed to the OLAP provider. Queries which are sent to remote data sources are optimized by the execution layer based on cache content and advanced null suppression logic. Suppressing rows and columns that contain only null values makes a report easier to read. For example, a product that has no sales for a given quarter may result in a very large report with thousands of cells that contain no data. The time required to evaluate a table to determine which rows and columns contain only null values is mainly determined by the number of cells in the table. Other factors such as the nesting levels on the axes and the use of complex calculated columns might also affect the time.

OLAP over Relational provides the ability to perform complex aggregate computations, and lifts previous aggregation limits which resulted in '--' or 'unknown' being displayed, such as when using Count Distinct as a summary.

Regarding First and Last aggregation rules to respect empty cells, previously, if you asked for the sales from the last period of a quarter and there were no sales for a product in that period, there would be no result to return. Instead of reporting zero, for the period in question, the sales for the last period in which there were sales would have been reported. With DQM, the query will return a zero for the specified period.

An example of performance metrics: with DQM for ROLAP: some tests showed an 85% performance improvement, with an average query time for compatible mode being 125 seconds, improving to 15 seconds average query time for DQM. Results will differ, depending on the environment and the queries being run.

Business Analytics software

IBM

Dynamic Query Mode Connectivity

- supports connectivity to supported OLAP and relational data sources
- OLAP connectivity is done through the native client libraries of the data source vendor
- supported relational data sources can remain in compatible query mode or further modeled dimensionally to provide an OLAP-style experience as dynamic query mode
- relational connectivity is done through JDBC

OLAP data source connections through DQM require a native client install. These native client installs can be either 32-bit or 64-bit as long as they match the bit depth of the DQM. For instance a 32-bit DQM requires a 32-bit SAP BW client library install while a 64-bit DQM would require a 64-bit SAP BW library, if that was the vendor database used.

Relational data source connectivity is done as a JDBC Type 4 connection with the exception of Oracle connections, which can be established as either Type 2 or Type 4 JDBC connections. Type 4 JDBC connections do not require the install of any native client libraries. Connectivity is established through the vendor JDBC driver which is copied to the ..\v5dataserver\lib directory and the p2pd\WEB-INF\lib directory. The Oracle Type 2 JDBC connection also requires the Oracle native library install in conjunction with the Oracle JDBC driver.

Business Analytics software

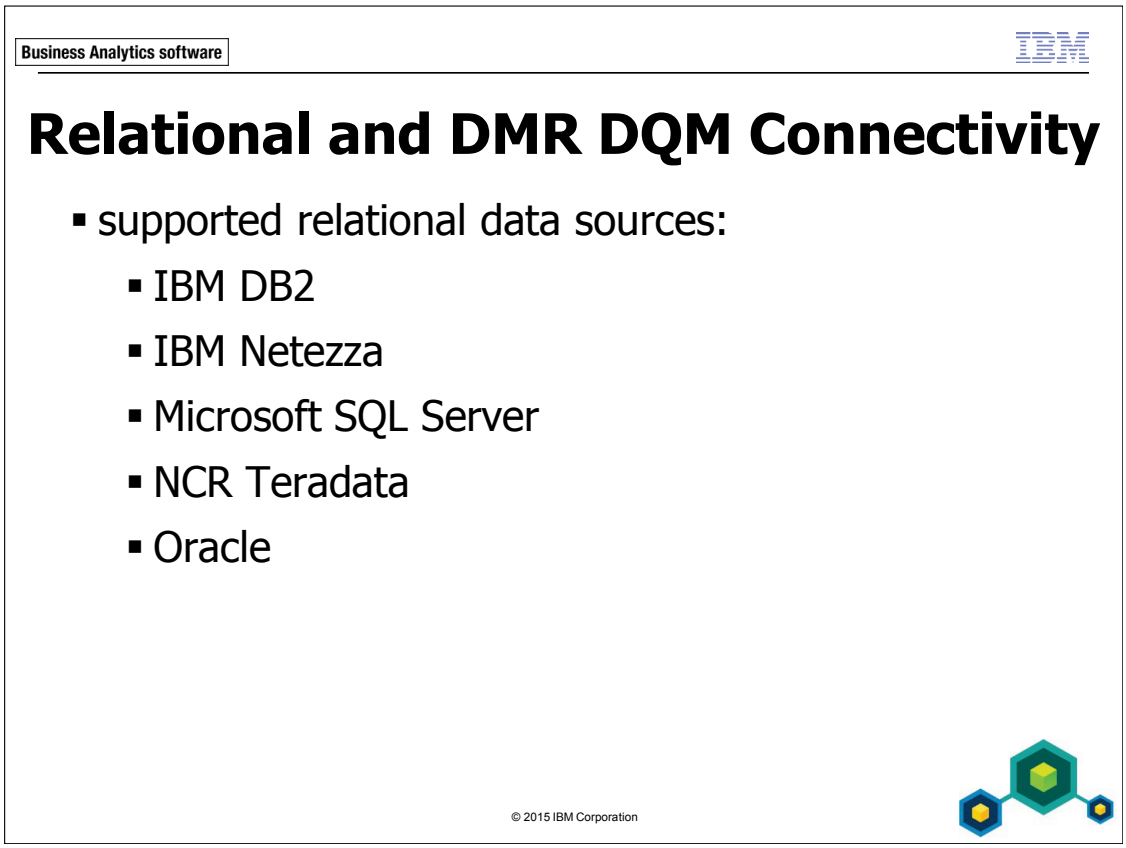
IBM

OLAP DQM Connectivity

- supported OLAP technologies:
  - IBM Cognos Dynamic Cubes
  - IBM Cognos TM1
  - SAP BW
  - Microsoft SSAS
  - Oracle Essbase

The data source specific capabilities files can be located within the ..\configuration\xqe directory. These files help govern the decision making process of DQM during the query planning phase. General OLAP behavior is defined within the OLAP.properties file which is used in conjunction of the data source specific capabilities file, such as bw.properties for SAP BW, eb.properties for Essbase, yk.properties for MSSAS, and so on.

These files should only be changed under the direction of IBM Cognos Support.

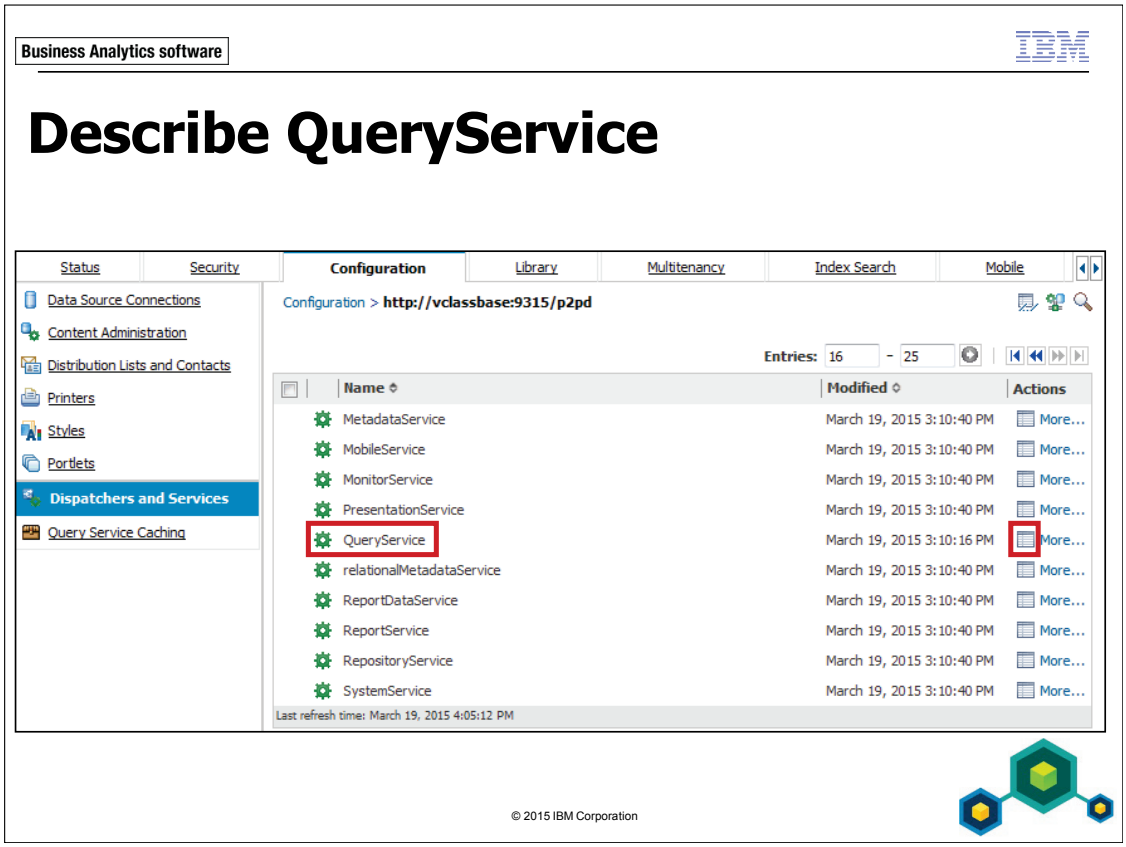


The data source specific capabilities files can be located within the ..\configuration\xqe directory. These files help govern the decision making process of DQM during the query planning phase. For relational queries, DQM uses the general vendor capabilities in conjunction with the vendor version specific capabilities to help with the query planning.

DQM would use:

- db2.sql.properties and db2.sql9.7.properties when connecting to IBM DB2 9.7
- netezza.properties and netezza5.0.properties when connecting to IBM Netezza 5.0

For DMR queries, the DQM behavior is the same as the above in addition to using the dmr.properties file. These files should only be changed under the direction of IBM Cognos Support or Development.



Dynamic Query Mode is also known as QueryService. DQM is coded in Java and runs within its own JVM. By default this JVM is configured to start with 1GB of memory and displays as a separate Java process on the operating system. Although there can be many QueryService services in a multiple server installation, you can only have one QueryService per IBM Cognos 10 dispatcher. Disable the Query Service if you do not need it, to save resources.

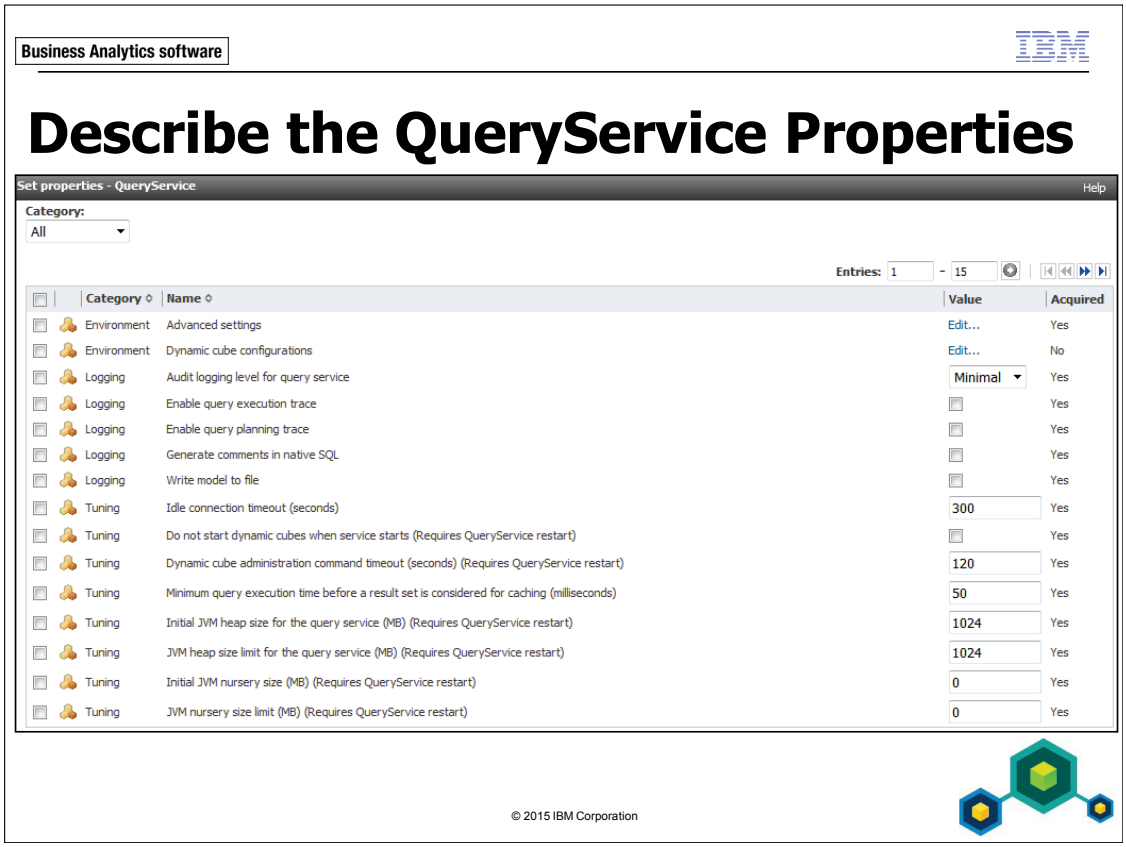
You can determine the Process ID (PID) from the ..\v5dataserver\.XQE file, where the information takes the format of PID@Servername:Port.

The Query Service is used as a secondary service by Report Service and Batch Report Service.

The QueryService receives requests from the BIBusTKServerMain process in the form of a query request, and from this request, plans the query. The planning of the query is where all the decisions are made to build an optimal query. Once the query is planned, the query is executed. The result of the query is passed back the BIBusTKServerMain process which then renders the report requested output.

In IBM Cognos Administration on the Configuration tab, Dispatchers and Services have a QueryService entry, which is used to configure settings for the Dynamic Query Mode.

The QueryService specific configuration can be found in the ..\configuration\xqe_config.xml file. This file should only be changed under direction of IBM Cognos Support or Development.



The QueryService properties allow for the administration of the Dynamic Query Mode. Settings of most interest include:

Enable query execution trace? Enabling this trace setting will allow an administrator to capture the query execution after it has gone through its planning phase. The log this trace produces is called the run tree log written to the ..\logs\XQE directory. This file usually accompanied by the profiling log which tracks the time spent in each of parts of the query execution. As a log is generated for each report that is executed, the log file adheres to the following naming convention.

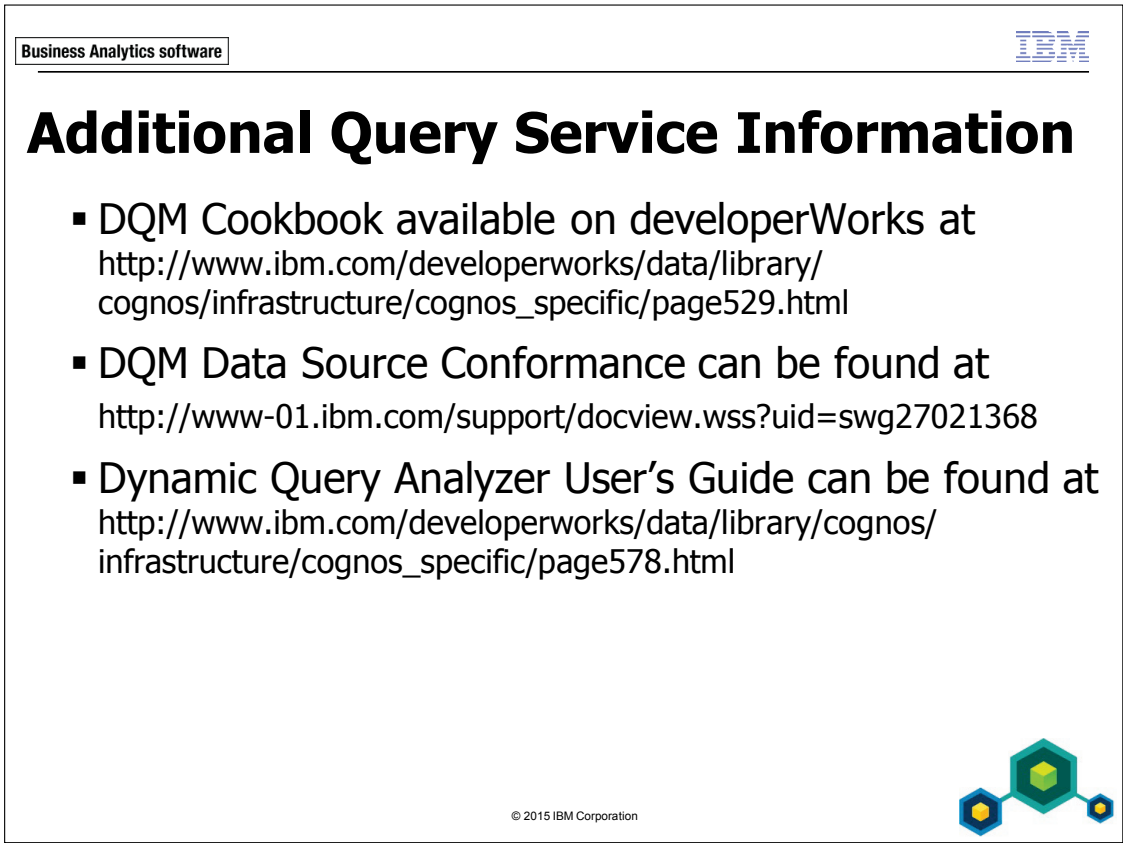
- <date>_<timestamp>_reportName/runtreeLog.xml
- <date>_<timestamp>_reportName/profilingLog-#.xml

As an example, executing a report called Top_Sales would result in a log file named 2015-02-10_11h33m700s_Top_Sales/runtreeLog.xml, and one or several profiler logs named 2015-02-10_11h33m700s_Top_Sales/profilingLog-0.xml, 2015-02-10_11h33m700s_Top_Sales/profilingLog-1.xml, and so on.

These files are best viewed with the Dynamic Query Analyzer application which is described in a later module.

Initial JVM heap size for the query service (MB) This setting defines how much memory the DQM Java Virtual Machine will take on startup. The value set here is passed to the JVM on startup as the –Xms<value> parameter.

JVM heap size limit for the query service (MB) This setting defines the upper memory limit of the DQM Java Virtual Machine during operation. The value set here is passed to the JVM on startup as the –Xmx<value> parameter.



The DQM Cookbook contains a significant amount of detail. It provides background information and troubleshooting tips as well.

Business Analytics software

IBM

What are Index Services?

- adds full-context search capabilities
- indexes
  - must be created once and then maintained
  - saved to file system (..\indexes\csn)
  - can be exported to integrate with other search engines

The former add-on product IBM Cognos Go! Search has been incorporated into IBM Cognos 10 to offer search capabilities for almost all data in the Content Store. The functionality is based on the Apache Jakarta Lucerne project, a text search engine library written entirely in Java.

Using Search requires an index to be available.

Business Analytics software

IBM

Explore Related Index Services

- Index Data Service
- Index Update Service
- Index Search Service

The Index Data service retrieves relevant data for index building from the Content Manager service. The Index Update service manages creating and updating indexes. The Index Search service is employed when searching through indexes.

If you have three or more application tier instances in your IBM Cognos 10 system, you can decide to limit the Index services to just a couple of instances, rather than on all instances. To do this, you can disable (in IBM Cognos Configuration) the Index services on all the instances except the ones you want to include the Index services on. Ideally, the Index services should be enabled as a group for any given instance.

Business Analytics software

IBM

Explain Collaboration Functionality

- annotations and comments
  - only through workspace widgets
  - aimed at business users
- integration with IBM Lotus Connections (LC)
  - link up workspaces with LC activities
  - link out to LC and its capabilities
- assignment of approval tasks to other users of IBM Cognos through Event Studio
  - aimed at report authors

IBM Cognos 10 BI supported adding rich text comments to report output versions. This includes annotations and comments which can be attached at a cell level or to the report as a whole.

You can integrate with IBM Lotus Connections (LC) by linking up workspaces with LC activities, thereby accessing the capabilities of LC.

You can also assign approval tasks to other users of IBM Cognos through Event Studio.

Business Analytics software

IBM

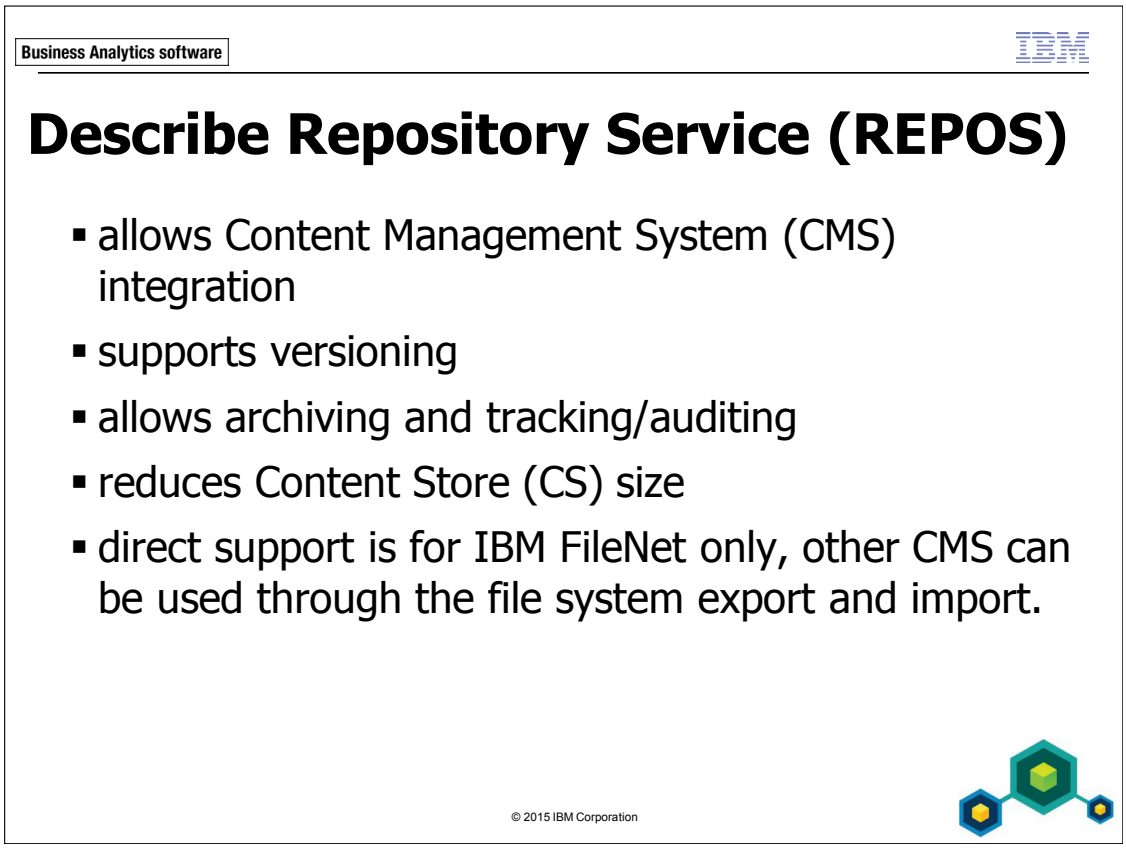
Describe Collaboration Services

- Human Task Service (HTS)
- Annotation Service (AS)
- both services are implemented in Java

Functionality is implemented by two services which belong to the scheduling context, HTS and AS. HTS handles the approval tasks which can be created in Event Studio; it interacts with AS and scheduling. AS manages annotations for reports. HTS is also implemented in My Inbox.

Both services are implemented in Java, and are part of a CM only install, as they run within the JRE hosting the Content Manager component.

Integration with LC is handled by Cognos Workspace and underlying components.



Since IBM Cognos 10.1.1, report outputs can be saved and archived in locations other than in the Content Store.

- The legacy feature of saving to the file system in addition to CS still exists with limited functionality.
- Support in 10.1.1 was limited. Full support is delivered as of 10.2.2.

Repository Service (REPOS) stores report outputs in one of the supported back-end repositories accessed as a Cognos data source:

- configurable path in the file system (local + UNC)

- IBM FileNet Content Manager (ECM):
  - not the same as the Cognos Content Manager component
  - offers a file system like structure, similar to a mounted drive
  - accessed by the user running Cognos Service

The FileNet system is attached by setting up a Cognos data source for writing. The file system is a virtual FileNet driver which writes to a file system path to a URL. FileNet is exposed as a file system, hence the similarity. File system configuration is done in IBM Cognos Configuration under Global Settings\Alias Root. This is an optional feature, as storing in CS is still fine.

The legacy feature which uses "Archive Location File System Root" is only providing file system storage next to storing in CS; the new feature offers both.

Business Analytics software

IBM

Explain REPOS

- part of the Application Tier install
- client components access directly through URLs generated by the Presentation Service
- scales horizontally
- multiple instances will work jointly
- Dispatcher distributes load
- REPOS requests are not subject to advanced routing
- local instances preferred when accessed by other services

REPOS comes with the Application Tier install; it is not provided with a Content Manager only install.

Load balancing is handled by the Dispatcher for REPOS requests. Service registration has a preference of "cluster compatible" mode. REPOS is only used when accessed by other Cognos Services for storing data. That is, a request to store something is unlikely to travel over the BIBus unless it has to (as in the case that no local REPOS instance is available).

There should be at least two instances of REPOS in a system for fail-over.

REPOS can be disabled if no archiving is used, or if output versions are only going to be kept in the Content Store.

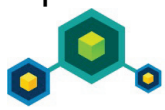
Business Analytics software

IBM

Configure REPOS

Category	Name	Value	Acquired
Environment	Advanced settings	Edit...	Yes
Logging	Audit logging level for repository service	Minimal	Yes
Tuning	Maximum number of seconds reports and report elements can exist in the cache.	1200	Yes
Tuning	Maximum number of reports and report elements that can overflow to disk.	1000	Yes
Tuning	Maximum number of reports and report elements that can be stored in memory.	100	Yes

- REPOS can consume a lot of memory if not configured properly
  - uses a two-level cache, local to every service instance
    - first level of cache is in Java heap (such as the Dispatcher memory)
    - second level of cache is an overflow to the local disk (/temp folder)
- cache control is only by number of objects (reports, report elements)



It is important to know your report output sizes; reduce the number of objects in the memory cache.

100 (objects) * 10 MB = 1 GB

Keep in mind that there is a cache for each REPOS instance The REPOS cache is a potential trap:

- understand your report output size
- outputs will be moved across BIBus and into the memory cache

If 100 users pull 100 different outputs from a single instance of REPOS within 20 minutes, they all are going to be loaded into the memory cache using default settings. Doing the math: 100 * 2 MB = no issue, 100 * 20 MB = possible issues. For two instances, that might cause half the cache to be built for each instance. Over time though, it might build up to two full copies of the cache.

In the initial release of 10.2, there is no control over the cache size. You will need to use the number of objects in your configuration (../configuration/cache/ehcache.xmlis cache settings file).

The overflow location is the ../temp folder of the local installation.

REPOS should be placed on servers with large RAM (> 8Gb).

Do not put REPOS next to CM or JMS as you do not want to take memory away from these services. It is better to put REPOS next to a Report Service or a Query Service, as the hardware that hosts these services usually has larger amounts of RAM.

Business Analytics software

IBM

What is Cognos Access Manager Service?

- internal service that handles authentication
  - CAM_AsyncAAService
- implemented by CAM, a Java component
- authentication providers remain C++ code
  - except Custom Java Authentication Providers (CJAPs)
- single instance in a system only for IBM Cognos 10

In IBM Cognos 10, authentication is handled by an internal service. Authentication is implemented by the CAM service, which is now a Java component. You should be aware that some authentication providers remain in C++ code, with the exception of Custom Java Authentication Providers (CJAPs).

In an IBM Cognos 10 system, only a single instance of CAM service will exist, and it must be run where the Content Manager runs.

Business Analytics software

IBM

Describe Authentication Providers

- each authentication provider executed in own C++ process
- 32-bit and partial true 64-bit support
- processes use BIBusTKServer interface
  - started by dispatcher upon startup (CAM_LPSvr (Legacy Provider Server))
  - terminated if idle for 10 min
  - processes will be re-spawned as required
  - user authenticated session retained even if the process is terminated

Each authentication provider is now executed in its own separate C++ process, with one process per configured namespace, with the exception of the Cognos namespace.

Having a full range of process memory available to the provider means an increase in support for the number of active users. On 32-bit installs, 2-3 GB C++ heap per configured namespace is available, and on 64-bit installs, > 4GB C++ heap per configured namespace is available.

Business Analytics software

IBM

Overview of IBM Cognos Architecture

- IBM Cognos 10 introduced several significant changes
  - more components moved to Java (CAM, DQM, current default engine)
  - more processes spawned for a single install (CAM_LPSvr, JRE for current default engine, JRE for DQM)
  - more services available
- architecture concepts remain the same mostly due to SOA
  - SOA will help manage new resource requirements
  - extendibility of the product due to SOA
  - added new services

IBM Cognos 10 includes more services such as workspaces, Search, and new functionality including DQM, current default engine, HTS, AS, and Cognos Workspace.

If you have experience with IBM Cognos 8, the architectural knowledge is still applicable for IBM Cognos 10.

Business Analytics software

IBM

Summary

- At the end of this module, you should be able to:
  - identify IBM Cognos Services
  - review collaboration functionality
  - explore the architecture in IBM Cognos 10

Workshop 1: Observe the Effects on a Dispatcher when Running Multiple Report Server Instances

As administrator, you want to run multiple Report Server instances to observe the results in your environment. To do this you will invoke multiple BIBusTKServerMain processes by running a number of reports simultaneously. You will run reports which contain current default engine charts, and then monitor the results.

You will do the following:

- ensure that both dispatchers are running
- launch Windows Task Manager, and select the Processes tab for monitoring
- login to IBM Cognos 10, to the LDAP_Dev namespace with admin/Education1 credentials, and launch Report Studio with any package; select Blank when prompted for an initial report object type
- ensure that Report Studio is using the current default engine (Tools\Options\Advanced, clear Use legacy chart authoring), and then close Report Studio
- create and save a job in My Folders to run the following reports (which contain charts) from Public Folders\Samples_DQ\Models\GO Data Warehouse (analysis)\Report Studio Report Samples:
  - Returns by Failed Orders in 2012_DQ
  - Returns by Order Method_DQ
- monitor the results in Windows Task Manager
- create and save a job in My Folders to run the following reports (which contain charts) from Public Folders\Samples\Models\GO Data Warehouse (analysis)\Report Studio Report Samples:
  - Returns by Failed Orders in 2012
  - Returns by Order Method
  - Sales Growth Year Over Year
- monitor the results in Windows Task Manager
- close all open windows

Question: What has to be done to monitor this if more than one dispatcher is running?

Answer: You could use routing rules to force the activity to one dispatcher, or through configuring load balancing, or other configurations. This environment uses routing rules.

For more information about where to work and the workshop results, refer to the Tasks and Results section that follows.

Workshop 1: Tasks and Results

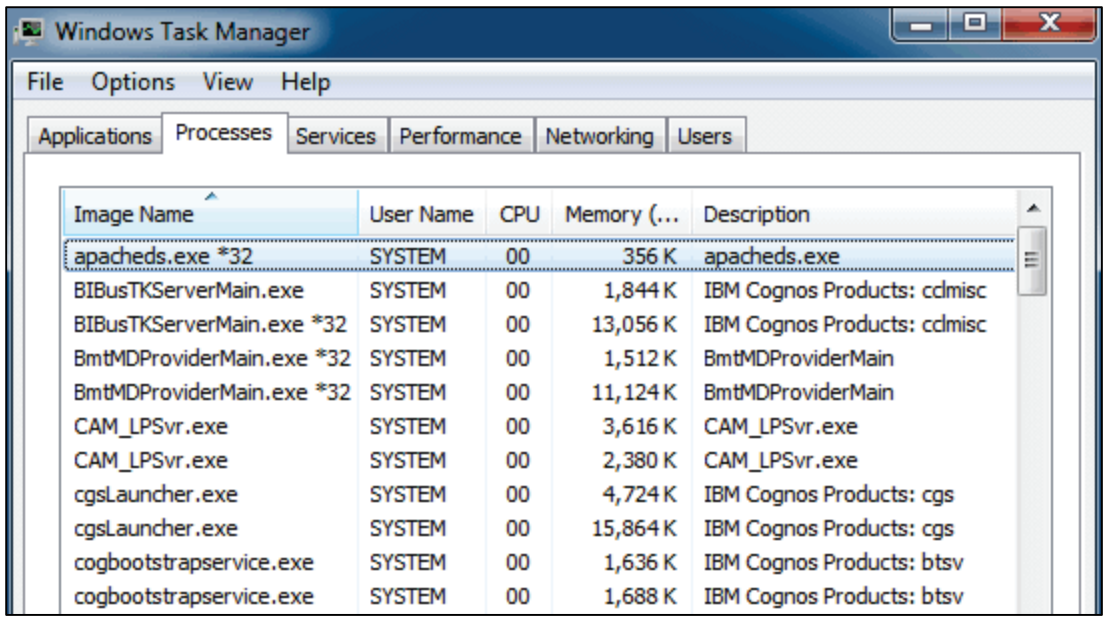
The IBM Cognos 10 Full:9315 dispatcher and the IBM Cognos 10 DispCM:9320 dispatcher should be running at the beginning of this workshop.

Task 1. Start Windows Task Manager.

- Ensure that both dispatchers are running, and then right-click the **Taskbar**, and click **Start Task Manger**.
- Click the **Processes** tab, click the **Image Name** column title to sort alphabetically ascending, and then review the processes that are running.

  Look for instances of BIBusTKServerMain.exe. Also identify instances of java.exe.

  A section of results which may appear are as follows. You may have different services running in addition to these:



  Keep this window open and visible on the desktop as you proceed with the next task.

Task 2. Run multiple reports.

- Launch **Internet Explorer**, go to **http://vclassbase:88/C10Full**, and then log on to the **LDAP_Dev** namespace with **admin/Education1** credentials.
- Click **Author advanced reports** to launch **Report Studio**, and when prompted for a package, navigate to **Public Folders\Samples_DQ\Models**.

- Click **GO Sales (analysis)**, click **Create new**, and then double-click **Blank** when prompted for a report object type. It will take a few moments for the package to load in Report Studio.

  You could have selected any package, as you have opened Report Studio to ensure that legacy charts are turned off, thereby using current default engine charts in your reports. This will cause current default engine JREs to execute as reports with current default engine charts are running.

- To review the **Report Studio** configuration, from the **Tools** menu, click **Options**, on the **Advanced** tab ensure that the **Use legacy chart authoring** check box is not selected, click **OK**, and then close **Report Studio**.

  You will create jobs to run multiple reports at the same time, so that you can monitor the results. One job will be for dynamic query based reports, and the other job will be for compatible query based reports.

- On the **IBM Cognos Software** page, click **IBM Cognos content** to launch IBM Cognos Connection, click the **My Folders** tab, and then on the toolbar, click **New Job**  .

  It may take a few moments for the My Folders tab to load.

- In the **Name** box, type **Run Multiple Chart Reports DQM**, and then click **Next**.

- Under the **Steps** pane, click **Add**, and then under the **Available entries** section on the left side, click **Cognos** and navigate to **Public Folders\Samples_DQ\Models\GO Data Warehouse (analysis)\Report Studio Report Samples**.

- Click the check boxes for the following reports (which contain charts), click **Add** (right yellow arrow) to add the selected entries, and then click **OK**:

  - **Returns by Failed Orders in 2012_DQ**
  - **Returns by Order Method_DQ**

  You may have to navigate to the next page of reports to get both selections.

- To accept the default settings for submission of steps click **Next**, on the **Select an action** page, click **Save only**, and then click **Finish**.

- On the **My Folders** tab, on the toolbar, click **New Job**.

- In the **Name** box, type **Run Multiple Chart Reports CQM**, and then click **Next**.
- Under the **Steps** pane, click **Add**, and then under the **Available entries** section click **Cognos** and navigate to **Public Folders\Samples\Models\GO Data Warehouse (analysis)\Report Studio Report Samples**.
- Add the following reports (which contain charts):
  - **Returns by Failed Orders in 2012**
  - **Returns by Order Method**

  You may have to navigate to the next page of reports to get both selections.
- Click **OK** to close the **Select entries** page.
- To accept the default settings for submission of steps click **Next**, on the **Select an action** page, click **Save only**, and then click **Finish**.

Task 3. Run the jobs, and observe the processes in Windows Task Manager.

- Ensure that you can see both the **Windows Task Manager** and **IBM Cognos Connection\My Folders** windows on your desktop.

  Resize and reposition the windows to see both, as necessary.
- In **Windows Task Manager**, observe the processes that are running, and notice in particular, if any instances of BIBusTKServerMain.exe and java.exe are running.

- In the browser window, in the **Actions** column for the **Run Multiple Chart Reports CQM** entry, click **Run with options**, click **Run**, and then click **OK**.

  1. Observe the number of instances of BIBusTKServerMain that are spawned.
  2. Question: How many instances were spawned?

  More instances of java.exe should be spawned for Graphics service, and for BIBusTKServerMain.

  The results appear similar to the following:

Image Name	User Name	CPU	Memory (...	Description
apacheds.exe...	SYSTEM	00	396 K	apacheds...
BIBusTKServe...	SYSTEM	00	26,068 K	IBM Cogn...
BIBusTKServe...	SYSTEM	00	184,972 K	IBM Cogn...
BIBusTKServe...	SYSTEM	00	3,048 K	IBM Cogn...
BIBusTKServe...	SYSTEM	00	171,124 K	IBM Cogn...
BmtMDProvid...	SYSTEM	00	1,464 K	BmtMDPr...
BmtMDProvid...	SYSTEM	00	2,404 K	BmtMDPr...
CAM_LPSvr.exe	SYSTEM	00	3,400 K	CAM_LPS...
CAM_LPSvr.exe	SYSTEM	00	2,324 K	CAM_LPS...
cgsLauncher....	SYSTEM	00	9,800 K	IBM Cogn...
cgsLauncher....	SYSTEM	00	5,748 K	IBM Cogn...
cogbootstrap...	SYSTEM	00	1,752 K	IBM Cogn...
cogbootstrap...	SYSTEM	00	1,844 K	IBM Cogn...

- If you have time, you can continue to observe the processes until they are complete, and they are no longer displayed.

  You may refresh the display by periodically pressing F5.

- When you have finished your observations, in the browser window, in the **Actions** column for the **Run Multiple Chart Reports DQM** entry, click **Run with options**, click **Run**, and then click **OK**.

- Observe the number of instances of BIBusTKServerMain and other services that are spawned.

  The results appear similar to the following:

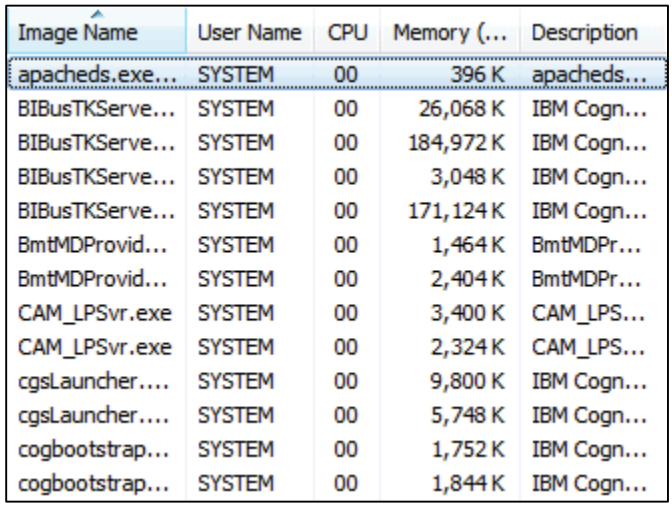


Image Name	User Name	CPU	Memory (...	Description
apacheds.exe...	SYSTEM	00	396 K	apacheds...
BIBusTKServe...	SYSTEM	00	26,068 K	IBM Cogn...
BIBusTKServe...	SYSTEM	00	184,972 K	IBM Cogn...
BIBusTKServe...	SYSTEM	00	3,048 K	IBM Cogn...
BIBusTKServe...	SYSTEM	00	171,124 K	IBM Cogn...
BmtMDProvid...	SYSTEM	00	1,464 K	BmtMDPr...
BmtMDProvid...	SYSTEM	00	2,404 K	BmtMDPr...
CAM_LPSvr.exe	SYSTEM	00	3,400 K	CAM_LPS...
CAM_LPSvr.exe	SYSTEM	00	2,324 K	CAM_LPS...
cgsLauncher....	SYSTEM	00	9,800 K	IBM Cogn...
cgsLauncher....	SYSTEM	00	5,748 K	IBM Cogn...
cogbootstrap...	SYSTEM	00	1,752 K	IBM Cogn...
cogbootstrap...	SYSTEM	00	1,844 K	IBM Cogn...

- When you have finished your observations, close all open windows.