The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee’s responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.
## Contents

### Chapter 1. Reporting and Analysis Architecture Overview

- Hyperion and Enterprise Performance Management ........................................... 11
- About Reporting and Analysis ........................................................................... 11
- Reporting and Analysis Architecture ................................................................ 12
  - Client Layer ...................................................................................................... 12
  - Application Layer ............................................................................................ 14
  - Database Layer ................................................................................................ 19

### Chapter 2. Administration Tools and Tasks

- Understanding Hyperion Home and Install Home ............................................. 21
- Administration Tools .......................................................................................... 22
  - Workspace Configuration Information ............................................................. 22
  - Administer Module ........................................................................................... 23
  - Interactive Reporting Services Manager ......................................................... 23
  - Impact Manager Module .................................................................................. 24
  - Job Utilities Calendar Manager ....................................................................... 24
  - Service Configurators ...................................................................................... 24
  - Servlet Configurator ....................................................................................... 25
- Starting and Stopping Services .......................................................................... 25
  - Before Starting Services .................................................................................. 25
  - Starting Reporting and Analysis Services ....................................................... 26
  - Starting Subsets of Services .......................................................................... 27
  - Starting Services and server.dat .................................................................... 28
  - Starting Interactive Reporting Services Manager .......................................... 28
  - Stopping Services ............................................................................................. 29
  - Service Start-up Dependencies ........................................................................ 29
- Changing Service Port Assignments .................................................................. 30
- Starting the Workspace Servlet ........................................................................... 30
- Configuring Interactive Reporting Services Manager ...................................... 30
  - Interactive Reporting Services Manager Event Thresholds ............................ 30
  - Configuring Interactive Reporting Services .................................................... 33
- Securing Your Workspace Production Environment ......................................... 36
Quick Guide to Common Administrative Tasks ................................................. 36

Chapter 3. Administer Module ................................................................. 39

Overview ................................................................................................... 39
Setting General Properties ........................................................................ 40
  General Properties .................................................................................. 40
  User Interface Properties ....................................................................... 41
Managing Users ....................................................................................... 41
Assigning Reporting and Analysis Default Preferences ....................... 41
Managing Physical Resources .................................................................. 42
  Viewing Physical Resources ................................................................... 42
  Access Permissions for Physical Resources ......................................... 42
  Adding Physical Resources ................................................................... 43
  Modifying Physical Resources .............................................................. 43
  Deleting Physical Resources .................................................................. 43
  Printer Properties .................................................................................. 43
  Output Directory Properties ................................................................... 44
Managing MIME Types ............................................................................. 44
  Defining MIME Types ........................................................................... 44
  Modifying MIME Types ........................................................................ 45
  Inactivating or Re-activating MIME Types ......................................... 45
  Deleting MIME Types ........................................................................... 46
Managing Notifications ............................................................................ 46
  Understanding Subscriptions and Notifications ................................. 46
  Modifying Notification Properties ....................................................... 48
Managing SmartCuts ................................................................................. 49
Managing Row-Level Security .................................................................. 49
Tracking System Usage ............................................................................ 50
  Managing Usage Tracking ..................................................................... 51
  Tracking Events and Documents ............................................................ 51
Sample Usage Tracking Reports ................................................................. 52

Chapter 4. Using Impact Management Services ........................................ 53

Introduction ............................................................................................ 53
About Impact Management Services ....................................................... 54
Impact Management Assessment Services ............................................. 54
About Impact Management Metadata ..................................................... 54
Impact Management Update Services ..................................................... 55
  Update Data Models Transformation ...................................................... 55
  Link Between Data Models and Queries ............................................... 55
<table>
<thead>
<tr>
<th>Chapter 8. Configuring LSC Services</th>
<th>165</th>
</tr>
</thead>
<tbody>
<tr>
<td>About LSC</td>
<td>165</td>
</tr>
<tr>
<td>Starting LSC</td>
<td>166</td>
</tr>
<tr>
<td>Using LSC</td>
<td>166</td>
</tr>
<tr>
<td>Modifying LSC Service Properties</td>
<td>167</td>
</tr>
<tr>
<td>Common LSC Properties</td>
<td>167</td>
</tr>
<tr>
<td>Assessment and Update Services Properties</td>
<td>168</td>
</tr>
<tr>
<td>Hyperion Interactive Reporting Service Properties</td>
<td>168</td>
</tr>
<tr>
<td>Hyperion Interactive Reporting Data Access Service Properties</td>
<td>170</td>
</tr>
<tr>
<td>Hyperion Interactive Reporting Job Service Properties</td>
<td>172</td>
</tr>
<tr>
<td>Modifying Host Properties</td>
<td>172</td>
</tr>
<tr>
<td>Host General Properties</td>
<td>173</td>
</tr>
<tr>
<td>Host Database Properties</td>
<td>173</td>
</tr>
<tr>
<td>Host Shared Services Properties</td>
<td>174</td>
</tr>
<tr>
<td>Host Authentication Properties</td>
<td>175</td>
</tr>
<tr>
<td>Modifying Properties in portal.properties</td>
<td>175</td>
</tr>
</tbody>
</table>
Chapter 9. Configuring the Servlets .................................................... 177
Using Servlet Configurator ................................................................. 177
Modifying Properties with Servlet Configurator ................................. 178
  User Interface Properties ................................................................. 178
  Personal Pages Properties ............................................................... 182
  Internal Properties ........................................................................... 184
  Cache Properties ............................................................................. 186
  Diagnostics Properties ................................................................. 188
  Applications Properties ................................................................. 188
Zero Administration and Interactive Reporting ........................................ 189
  6x Server URL Mapping .................................................................. 190
  Client Processing ............................................................................. 190
Load Testing Interactive Reporting ......................................................... 191
  Data Access Servlet Property ............................................................. 192
  Hyperion Interactive Reporting Data Access Service Property ............. 192
  Hyperion Interactive Reporting Service Property ................................ 192

Chapter 10. Backup Strategies ............................................................. 193
Overview ............................................................................................ 193
What to Back Up ................................................................................. 193
General Backup Procedure .................................................................. 194
Maintenance Mode for System Backups ................................................. 194
Backing Up the Reporting and Analysis File System ............................... 195
  Complete Backup ........................................................................... 195
  Post-Installation ............................................................................. 195
  Weekly Full and Daily Incremental .................................................... 196
  As Needed ...................................................................................... 196
  Reference Table for All File Backups ................................................. 196
Sample Backup Script ......................................................................... 197
Backing Up the Repository Database ..................................................... 198
  Backing Up the Workspace Servlets .................................................. 198

Chapter 11. Troubleshooting ................................................................. 199
Overview ............................................................................................ 199
Logging Architecture .......................................................................... 199
  Log4j ............................................................................................. 200
  Logging Service .............................................................................. 200
  Log Management Helper ............................................................... 200
  Server Synchronization ................................................................... 200
  Log File Basics ............................................................................. 200
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log File Location</td>
<td>201</td>
</tr>
<tr>
<td>Hyperion Interactive Reporting Service and Hyperion Interactive Reporting Data Access Service Local Log Files</td>
<td>201</td>
</tr>
<tr>
<td>Log File Naming Convention</td>
<td>201</td>
</tr>
<tr>
<td>Log Message File Format</td>
<td>203</td>
</tr>
<tr>
<td>Configuration Log</td>
<td>203</td>
</tr>
<tr>
<td>Configuring Log Properties for Troubleshooting</td>
<td>203</td>
</tr>
<tr>
<td>Configuration Files</td>
<td>204</td>
</tr>
<tr>
<td>Configuring Logging Levels</td>
<td>204</td>
</tr>
<tr>
<td>Configuring Appenders</td>
<td>206</td>
</tr>
<tr>
<td>Configuring Log Rotation</td>
<td>207</td>
</tr>
<tr>
<td>Analyzing Log Files</td>
<td>209</td>
</tr>
<tr>
<td>Viewing Log Files</td>
<td>209</td>
</tr>
<tr>
<td>Standard Console Log File</td>
<td>209</td>
</tr>
<tr>
<td>Logs for Importing General Content</td>
<td>209</td>
</tr>
<tr>
<td>Logs for Importing Interactive Reporting Content</td>
<td>210</td>
</tr>
<tr>
<td>Logs for Running Jobs</td>
<td>210</td>
</tr>
<tr>
<td>Logs for Logon and Logoff Errors</td>
<td>211</td>
</tr>
<tr>
<td>Logs for Access Control</td>
<td>211</td>
</tr>
<tr>
<td>Logs for Configuration</td>
<td>212</td>
</tr>
<tr>
<td>Information Needed by Customer Support</td>
<td>212</td>
</tr>
<tr>
<td><strong>Glossary</strong></td>
<td>213</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>237</td>
</tr>
</tbody>
</table>
Hyperion offers a comprehensive Enterprise Performance Management (EPM) system that consists of these products:

- Oracle’s Hyperion® Reporting and Analysis – System 9—Management reporting including query and analysis in one coordinated environment
- Oracle’s Hyperion Applications—Coordinated planning, consolidation, and scorecarding applications
- Oracle’s Hyperion® Foundation Services—Used to ease installation and configuration, provide metadata management, and support a common Microsoft Office interface

Reporting and Analysis is a modular business intelligence platform that provides management reporting, query, and analysis capabilities for a wide variety of data sources in one coordinated environment.

One zero-footprint Web-based thin client provides users with access to content:

- Financial reporting for scheduled or on-demand highly formatted financial and operational reporting from most data sources including Oracle’s Hyperion® Planning – System 9 and Oracle's Hyperion® Financial Management – System 9
- Interactive reporting for ad hoc relational queries, self-service reporting and dashboards against ODBC data sources
- Production reporting for high volume enterprise-wide production reporting
- Web analysis for interactive ad hoc analysis, presentation, and reporting of multidimensional data
Reporting and Analysis, which includes Oracle’s Hyperion® Essbase® – System 9, is part of a comprehensive EPM system that integrates this business intelligence platform with Oracle's Hyperion financial applications and Oracle's Hyperion® Performance Scorecard – System 9.

### Reporting and Analysis Architecture

The Reporting and Analysis environment is organized into three layers:

- “Client Layer” on page 12
- “Application Layer” on page 14
- “Database Layer” on page 19

#### Client Layer

The client layer refers to local interfaces used to author, model, analyze, present, report, and distribute diverse content, and third-party clients, such as Microsoft Office:
● Oracle’s Hyperion® Workspace—Web-based DHTML zero-footprint client that provides the user interface for viewing and interacting with content created by the authoring studios, and enables users to create queries against relational and multidimensional data sources:
  ○ Essbase—High performance multidimensional modeling, analysis, and reporting
  ○ Oracle’s Hyperion® Financial Reporting – System 9—Highly formatted financial reporting
  ○ Oracle’s Hyperion® Interactive Reporting – System 9—Ad hoc query, analysis, and reporting including dashboards
  ○ Oracle’s Hyperion® SQR® Production Reporting – System 9—High volume enterprise production reporting
  ○ Oracle’s Hyperion® Web Analysis – System 9—Advanced interactive ad hoc analysis, presentation, and reporting against multidimensional data sources

● Authoring Studios
  ○ Oracle’s Hyperion® Interactive Reporting Studio—Highly intuitive and easy-to-navigate environment for data exploration and decision making. A consistent design paradigm and robust formatting tools enable users to easily build free-form, presentation-quality reports for broad-scale publishing across their organization.
  ○ Oracle’s Hyperion® Interactive Reporting Web Client—Read-only Web plug-in for viewing Interactive Reporting documents.
  ○ Oracle’s Hyperion® Financial Reporting Studio—Windows client for authoring highly formatted financial reports from multidimensional data sources, which features easy, drag and drop, reusable components to build and distribute HTML, PDF, and hardcopy output.
  ○ Oracle’s Hyperion® Web Analysis Studio—Java applet that enables you to create, analyze, present, and report multidimensional content. The studio offers the complete Web Analysis feature set to designers creating content, including dashboards for information consumers.
  ○ Oracle’s Hyperion® SQR® Production Reporting Studio—Windows client that provides the design environment for creating reports from a wide variety of data sources. Reports can be processed in one pass to produce a diverse array of pixel-perfect output. Processing can be scheduled and independently automated, or designed to use form templates that prompt dynamic user input.
  ○ Oracle’s Hyperion® Dashboard Development Services—Enables creation of dashboards:
    ○ Dashboard Studio—Windows client that utilizes extensible and customizable templates to create interactive, analytical dashboards without the need to code programming logic.
    ○ Dashboard Architect—Windows-based integrated development environment that enables programmers to swiftly code, test, and debug components utilized by Dashboard Studio.
    ○ Performance Scorecard—Web-based solution for setting goals and monitoring business performance using recognized scorecarding methodologies. Provides tools
that enable users to formulate and communicate organizational strategy and accountability structures:

- **Key Performance Indicators (KPIs)**—Create tasks and achievements that indicate progress toward key goals
- **Performance indicators**—Indicate good, acceptable, or poor performance of accountability teams and employees
- **Strategy maps**—Relate high-level mission and vision statements to lower-level actionable strategy elements
- **Accountability maps**—Identify those responsible for actionable objectives
- **Cause and Effect maps**—Depict interrelationships of strategy elements and measure the impact of changing strategies and performance

- **Oracle’s Hyperion® Smart View for Office**—Hyperion-specific Microsoft add-in and toolbar from which users can query Hyperion data sources including Essbase, Financial Management, and Planning. Users can use this environment to interact with Financial Management and Planning forms for data input, and can browse the Reporting and Analysis repository and embed documents in the office environment. Documents are updated by user request.

**Application Layer**

The application layer—a middle tier that retrieves requested information and manages security, communication, and integration—contains two components:

- “Application Layer Web Tier” on page 14
- “Application Layer Services Tier” on page 14

Because the business intelligence platform is modular, it may consist of various combinations of components, configured in numerous ways. The end result is a comprehensive, flexible architecture that accommodates implementation and business needs.

**Application Layer Web Tier**

The application layer relies upon a J2EE application server and Web server to send and receive content from Web clients. An HTTP connector is required to link the Web server and the application server.


For a complete description of supported Web tier hardware and software, see the *Hyperion Installation Start Here*.

**Application Layer Services Tier**

The application layer services tier contains services and servers that control functionality of various Web applications and clients:
Most services fall into two main groups, depending on the tool used to configure their properties:

- **Local services**—Services in the local Install Home that are configured using the Local Service Configurator (LSC). Referred to as LSC services.
- **Remote services**—Services on a local or remote host that are configured using the Remote Service Configurator (RSC). Referred to as RSC services.

Because most of these services are replicable, you may encounter multiple instances of a service in a system.

**Core Services**

Core Services are mandatory for authorization, session management, and document publication:

- **Repository Service**—Stores Hyperion system data in supported relational database tables, known collectively as the repository. Repository Service is replicable; if replicated, it must be replicated and co-located with Publisher Service.
- **Publisher Service**—Handles repository communication for other LSC services and some Web application requests; forwards repository requests to Repository Service and passes replies back to initiating services. Publisher Service is replicable; if replicated, it must be replicated and co-located with Repository Service.
- **Global Service Manager (GSM)**—Tracks system configuration information and monitors registered services in the system. A system can have only one GSM.
- **Local Service Manager (LSM)**—Created for every instance of an LSC or RSC service, including GSM. When system servers start, they register their services and configuration information with GSM, which supplies and maintains references to all other registered services.
- **Authentication Service**—Checks user credentials at logon time and determines whether they can connect; determines group memberships, which, along with roles, affects what content and other system artifacts (resources) users can view and modify. Authentication Service is replicable and does not need to be co-located with other services.
- **Authorization Service**—Provides security at the level of resources and actions; manages roles and their associations with operations, users, groups, and other roles. A system must have at least one Authorization Service.
- **Session Manager Service**—Monitors and maintains the number of simultaneous system users. Monitors all current sessions and terminates sessions that are idle for more than a specified time period. While Session Manager is replicable, each instance independently manages a set of sessions.
Service Broker—Supports GSM and LSMs by routing client requests and managing load balancing for RSC services. A system can have multiple Service Brokers.

Name Service—Monitors registered RSC services in the system, and provides them with system configuration information from server.xml. Works in conjunction with Service Broker to route client requests to RSC services. A system can have only one Name Service.

Management services—Core Services that collect and distribute system messages and events for troubleshooting and usage analysis:

 Logging Service—Centralized service for recording system messages to log files. A system can have only one Logging Service.

 Usage Service—Records the number and nature of processes addressed by Hyperion Interactive Reporting Service, which enables administrators to review usage statistics such as the number of logons, what the most used files are, what the most selected MIME types are, and what happens to system output. Systems can have multiple Usage Services.

Functional services—Core Services that are specific to various functional modules:

 Job Service—Executes scripts that create reports, which can be prompted by users with permissions or by Event Service. Report output is returned to initiating users or published to the repository. Job Services can be created and configured for every executable.

 Event Service—Manages subscriptions to system resources. Tracks user subscriptions, job parameters, events and exceptions, and prompts Job Service to execute scheduled jobs. Event Service is configured to distribute content through e-mail and FTP sites, and to notify users with subscriptions about changing resources. A system can have only one Event Service.

Oracle’s Hyperion® Impact Management Services—Core Services that enable you to harvest, update, and publish new Interactive Reporting content from old Interactive Reporting repository resources. These services must be used in conjunction with Interactive Reporting services. Both services perform automatic load balancing and fault tolerance when multiple instances are running:

 Assessment (Harvester) Service—Harvests metadata from published Interactive Reporting repository documents.

 Update (Transformer) Service—Updates published and harvested Interactive Reporting documents or publishes new versions to the repository.

Interactive Reporting Services

Interactive Reporting services are Core Services that support Interactive Reporting functionality by communicating with data sources, starting RSC services, and distributing Interactive Reporting client content:

 Hyperion Interactive Reporting Service—Runs Interactive Reporting jobs and delivers interactive HTML content for Interactive Reporting files. When actions involving Interactive Reporting documents are requested, Hyperion Interactive Reporting Service fulfills such requests by obtaining and processing the documents and delivering HTML for display.
● Hyperion Interactive Reporting Data Access Service—Provides access to relational and multidimensional databases, and carries out database queries for the plug-in, Hyperion Interactive Reporting Service, and Interactive Reporting jobs. Each Hyperion Interactive Reporting Data Access Service supports connectivity to multiple data sources, using the connection information in one or more Interactive Reporting database connection files, so that one Hyperion Interactive Reporting Data Access Service can process a document whose sections require multiple data sources. Hyperion Interactive Reporting Data Access Service maintains a connection pool for database connections.

● Extended Access for Hyperion Interactive Reporting Service—Enables users to jointly analyze multidimensional and relational sources in one document. It retrieves flattened OLAP results from Web Analysis documents, SQR Production Reporting job output, or Financial Reporting Batch reports in the Reporting and Analysis repository and imports data into Interactive Reporting documents (.bqv) as Results sections.

Financial Reporting Servers

Financial Reporting servers support Financial Reporting functionality by processing batch requests, generating output, and distributing Financial Reporting client content:

● Hyperion Financial Reporting Server—Generates and formats dynamic report or book results, including specified calculations. Hyperion Financial Reporting Server can handle numerous simultaneous requests for report execution from multiple clients, because each request is run on its own execution thread. Hyperion Financial Reporting Server caches data source connections, so multiple requests by the same user do not require a re-connection. Financial Reporting servers are replicable—the number necessary depends on the number of concurrent users who want to execute reports simultaneously through the clients. Multiple Financial Reporting servers can be configured to report against one repository.

● Hyperion Financial Reporting Communication Server—Provides a Java RMI Registry to which other Financial Reporting servers are bound.

● Hyperion Financial Reporting Print Server—Enables Financial Reporting content to be compiled as PDF output. Runs only on supported Windows platforms, but is replicable to provide scalability for PDF generation.


SQR Production Reporting Service

SQR Production Reporting Service responds to scheduled and on-demand requests by Job Service to run jobs, process data, and generate reports. SQR Production Reporting Service is optimized for high volume reporting through the use of native drivers, array processing for large data sets, and cursor management. It processes time-saving data manipulation operations in one pass of the data source and produces large quantities of reports in online and printed formats. SQR Production Reporting Service is a replicable service.
**Assistive Hyperion Services**

- **Performance Scorecard Services**—Support Performance Scorecard client functionality used in conjunction with Reporting and Analysis.
- **Oracle’s Hyperion® Shared Services**—Supports authentication and user provisioning for all Hyperion products. See the Shared Services documentation set.
- **Smart View Services**—Provide a common Microsoft Office interface for Hyperion products. See the Smart View documentation set.

### Services Tier Summary

<table>
<thead>
<tr>
<th>LSC or RSC Service</th>
<th>Type</th>
<th>Name</th>
<th>Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSC</td>
<td>Core</td>
<td>Authentication Service</td>
<td>Multiple</td>
</tr>
<tr>
<td></td>
<td>Core</td>
<td>Authorization Service</td>
<td>Multiple</td>
</tr>
<tr>
<td></td>
<td>Core</td>
<td>Global Service Manager</td>
<td>1 per system</td>
</tr>
<tr>
<td></td>
<td>Core</td>
<td>Local Service Manager</td>
<td>Multiple</td>
</tr>
<tr>
<td></td>
<td>Core</td>
<td>Publisher Service</td>
<td>Multiple</td>
</tr>
<tr>
<td></td>
<td>Core</td>
<td>Session Manager</td>
<td>Multiple</td>
</tr>
<tr>
<td>Impact Management</td>
<td>Assessment (Harvester) Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Impact Management</td>
<td>Update (Transformer) Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Interactive Reporting</td>
<td>Extended Access for Hyperion Interactive Reporting Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Interactive Reporting</td>
<td>Hyperion Interactive Reporting Job Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Interactive Reporting</td>
<td>Hyperion Interactive Reporting Data Access Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Interactive Reporting</td>
<td>Hyperion Interactive Reporting Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Management</td>
<td>Logging Service</td>
<td></td>
<td>1 per system</td>
</tr>
<tr>
<td>Management</td>
<td>Usage Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>RSC</td>
<td>Core</td>
<td>Name Service</td>
<td>1 per system</td>
</tr>
<tr>
<td></td>
<td>Core</td>
<td>Repository Service</td>
<td>Multiple</td>
</tr>
<tr>
<td></td>
<td>Core</td>
<td>Service Broker</td>
<td>Multiple</td>
</tr>
<tr>
<td>Functional</td>
<td>Event Service</td>
<td></td>
<td>1 per system</td>
</tr>
<tr>
<td>Functional</td>
<td>Job Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Assistive Hyperion</td>
<td>Common Administration Services</td>
<td>Shared Services</td>
<td>1 per system</td>
</tr>
<tr>
<td>LSC or RSC Service</td>
<td>Type</td>
<td>Name</td>
<td>Instances</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------</td>
<td>-------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Financial Reporting Servers</td>
<td>Financial Reporting Print Server</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Financial Reporting Servers</td>
<td>Financial Reporting Server</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Performance Scorecard Services</td>
<td>Scorecard Module Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQR Production Reporting Service</td>
<td>SQR Production Reporting Service</td>
<td></td>
<td>Multiple</td>
</tr>
<tr>
<td>Smart View Services</td>
<td>Smart View Services</td>
<td></td>
<td>Multiple</td>
</tr>
</tbody>
</table>

**Database Layer**

Architecturally, databases fall into two fundamental groups: repositories that store Hyperion system data; and data sources that are the subject of analysis, presentation, and reporting.

There are two important repositories for information storage:

- **Common repository**—Hyperion system data in supported relational database tables
- **Shared Services**—User, security, and project data that can be used across Hyperion products

**Data Sources:**

- Relational data sources, for example, Oracle, IBM DB2, and Microsoft SQL Server
- Multidimensional data sources, for example, Oracle’s Hyperion® Essbase® – System 9
- Oracle’s Hyperion applications, for example, Oracle’s Hyperion® Financial Management – System 9 and Oracle’s Hyperion® Planning – System 9
- Data warehouses
- ODBC data sources

For a complete description of supported data sources, see the *Hyperion Installation Start Here*. 
Understanding Hyperion Home and Install Home

When multiple Hyperion products are installed on one computer, common internal and third-party components are installed to a central location, called Hyperion Home.

On Windows platforms, the Hyperion Home location is defined in the system environment variable called HYPERION_HOME and the default location is C:\Hyperion.

On UNIX platforms, the Hyperion Home value is stored in .hyperion.<hostname> in the home directory, and the default location is $HOME/Hyperion.

See the Hyperion Reporting and Analysis – System 9 Installation Guides for information on changing the Hyperion Home location.

Hyperion Home contains a \common directory. A Reporting and Analysis installation adds a \BIPlus directory to Hyperion Home, which is the default installation location or Install Home for Reporting and Analysis (that is, C:\HYPERION_HOME\BIPlus on Windows, or $HOME/Hyperion/BIPlus on UNIX).

It is possible to have multiple Install Homes on one physical host. All Java services in an Install Home run in one process space and share a GSM (not necessarily on one host). If a host has multiple Install Homes, each Install Home requires its own separate services process space and is managed by its own GSM. Services in an Install Home are referred to collectively as an Install Home.
Administration Tools

Topics that describe Workspace and Reporting and Analysis administration tools:

- “Workspace Configuration Information” on page 22
- “Administer Module” on page 23
- “Interactive Reporting Services Manager” on page 23
- “Impact Manager Module” on page 24
- “Job Utilities Calendar Manager” on page 24
- “Service Configurators” on page 24
- “Servlet Configurator” on page 25

Workspace Configuration Information

Configuration information about your Workspace installation, including GSM and Shared Services information, is available at this URL:

http://hostname:port/workspace/browse/configInfo

where hostname is the name of the Workspace server, and port is the TCP port on which the application server is listening. The default port for Workspace is 19000 if using Apache Tomcat.

Workspace timeout settings:

- **Application server session timeout**— Time in minutes after which the application server times out a session if no requests are received for that session
  - In `<Workspace deployment\WEB-INF\web.xml>`:
    ```xml
    <session-config>
    <session-timeout>60</session-timeout>
    </session-config>
    ```
  - Default is 60 minutes

  **Note:**
  All Reporting and Analysis Web applications session timeouts should be greater than 10 minutes.

- **Workspace timeout**—Time in minutes after which Workspace warns and logs out inactive users

  Workspace sends “keep alive” requests to all the application servers (Workspace, Financial Reporting, Web Analysis, and Oracle's Hyperion® Performance Scorecard – System 9). If there is no activity in Workspace per the session timeout setting, the user is warned that the session is about to end. If the user does not take action within 5 minutes, the user is logged out.
  ```
  KeepAliveInterval=10
  SessionTimeout=30
  ```
• Default is 10 minutes for keep alive and 30 minutes for session timeout

For information on disabling the configuration URL, see “Securing Your Workspace Production Environment” on page 36.

**Administer Module**

Properties managed using the Administer module (accessed using Navigate):

- General properties
- Your organization, including adding and modifying users, groups, and roles, through the Oracle's Hyperion® Shared Services User Management Console
- Physical resources including printers and output directories
- MIME types
- Notifications
- SmartCuts
- Row-level security
- Usage tracking
- Event tracking

For detailed information on managing these items, see “Administer Module” on page 39.

For information about common user-interface features among the modules, see the *Hyperion Workspace User’s Guide*.

**Interactive Reporting Services Manager**

Interactive Reporting Services Manager enhances the process monitoring capabilities for these Interactive Reporting services and processes:

- Hyperion Interactive Reporting Service
- Hyperion Interactive Reporting Data Access Service
- Hyperion Interactive Reporting Job Service
- Hyperion Interactive Reporting Service Helper Process—Manages logging and usage tracking for all Interactive Reporting services

Interactive Reporting Services Manager eliminates the need for using separate process monitors to manage multiple Interactive Reporting services.

See “Configuring Interactive Reporting Services Manager” on page 30.
Impact Manager Module
Impact Manager module enables users to replace Interactive Reporting data models. Changing the data model enables global changes across all Interactive Reporting documents, without requiring that every document which references a data source be edited individually.

See “Using Impact Management Services” on page 53.

Job Utilities Calendar Manager
You create, modify, and delete custom calendars using Job Utilities Calendar Manager. You can create calendars to schedule jobs based on fiscal or other internal or organizational calendars.

See “Viewing Calendar Manager” on page 123.

Service Configurators
All Reporting and Analysis services have configurable properties that you modify using Local Service Configurator (LSC) or Remote Service Configurator (RSC). LSC and RSC handle different services.

RSC
RSC provides a graphical interface to manage a subset of Reporting and Analysis service types referred to as RSC (or remote) services. You use RSC to configure services on all hosts in the system:
  ● Modify or view RSC service properties
  ● Ping services
  ● Add, modify, or delete hosts
  ● Add, modify, or delete database servers in the system
  ● Delete services

See “Configuring RSC Services” on page 143.

LSC
LSC enables you to manage a subset of Reporting and Analysis services on a local host, referred to as LSC (or local) services:
  ● View or modify properties of LSC services
  ● View or modify properties of the local Install Home
  ● Configure pass-through settings

See “Configuring LSC Services” on page 165.
Servlet Configurator

Servlet Configurator enables you to customize the Browse, Personal Pages, Scheduler, and Administration servlets for your organization. The many settings include the length of time to cache various types of data on the servlets, colors of various user interface elements, the locale, and language.

See “Configuring the Servlets” on page 177.

Starting and Stopping Services

To start Workspace, you start services in each Install Home, and start each installation of Workspace servlets (usually on a Web server).

This section focuses on how to start services of an Install Home, except for a discussion at the end about starting various Install Home services and hosts of a distributed system.

In an Install Home, you can start all installed services, a subset of them, or an individual service. Interactive Reporting Services Manager always should be started separately.

How you start services depends on your operating system, Reporting and Analysis system configuration, and objectives. How you stop services depends on how you started them.

Topics that explain prerequisites and methods for starting and stopping services:

- “Before Starting Services” on page 25
- “Starting Reporting and Analysis Services” on page 26
- “Starting Subsets of Services” on page 27
- “Stopping Services” on page 29
- “Service Start-up Dependencies” on page 29

Before Starting Services

Before starting services, ensure that all required network resources are available to the services. For example, Interactive Reporting services may need to create job output on printers or file directories belonging to network hosts other than where the services are running. These connections must be established before Interactive Reporting services can start.

For Windows, a service may need to log on as a user account rather than as the local system account to establish connections to shared resources on the network. ODBC data sources must be configured as system data sources rather than user data sources. Consult with the site’s network administrators to configure the correct environment.

For UNIX platforms, all necessary environmental settings should be made prior to starting services. Consult with the site’s network administrators to create the necessary software configuration.

Regardless of your method for starting Reporting and Analysis services, you must first start the repository database.
Starting Reporting and Analysis Services

Regardless of whether you installed the complete set of services in an Install Home, a few services, or one service, the methods presented in these topics can be used to start Reporting and Analysis services of a given Install Home:

- "start_BIPlus Method" on page 26
- “startCommonServices Method” on page 26
- “Windows Service Methods (Windows Only)” on page 27

For a usable system, all Core Services must be started (see “Core Services” on page 15).

Note:

Interactive Reporting Services Manager must be started separately (see “Starting Interactive Reporting Services Manager” on page 28). Hyperion recommends that you restart your Web server after restarting Reporting and Analysis services. If you do not restart the Web server, a delay of several minutes occurs before users can log on.

start_BIPlus Method

To start all Reporting and Analysis services (that is, Core Services, Interactive Reporting services, Financial Reporting services) and Web applications (Workspace, Financial Reporting, and Web Analysis), run the start_BIPlus script in \BIPlus\bin:

- UNIX—start_BIPlus.sh
- Windows—start_BIPlus.bat

startCommonServices Method

To start only Core Services, run the startCommonServices script in \BIPlus\bin:

- UNIX—startCommonServices.sh
- Windows—startCommonServices.bat

startCommonServices starts the Java services in an Install Home, except for inactivated ones. Inactivating services is discussed in “Starting Subsets of Services” on page 27.

<table>
<thead>
<tr>
<th>startCommonServices Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Dminimum_password_length</td>
<td>Length of database passwords. Default is 5.</td>
</tr>
<tr>
<td>-Ddisable_htmlemail</td>
<td>Format for e-mails (HTML or text file). Default is HTML format.</td>
</tr>
<tr>
<td>-DPerformance.MaxSTWorkers</td>
<td>Number of job worker threads. Determines the speed at which jobs are built and sent to Job Service. Configure based on number of Job Services, schedules, and events; and size of connection pool for the repository. Default is 2.</td>
</tr>
<tr>
<td><strong>startCommonServices Flag</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-DPerformance.SchedulerBatchSize</td>
<td>Number of schedules processed at one time by the scheduler worker thread. Default is 15.</td>
</tr>
<tr>
<td>-DPerformance.SchedulerDelay</td>
<td>Number of seconds job execution is delayed when Job Services are busy. Default is 300.</td>
</tr>
<tr>
<td>-Djob_limit</td>
<td>Number of concurrent jobs per each Job Service. No default limit.</td>
</tr>
<tr>
<td>-DMCL.JDBFetchMaxRowSize</td>
<td>Number of artifacts returned for a listing, number of Interactive Reporting database connection files, number of items returned from a search, and the artifacts listed in a Personal Page. Default is 100.</td>
</tr>
</tbody>
</table>

**Windows Service Methods (Windows Only)**

On Windows, the preferred method for starting Core Services is from Windows Services or from the Start menu.

➤ To start Core Services, use one method:

- Select Start > Programs > Hyperion > Reporting and Analysis > Utilities and Administration > Start Reporting and Analysis Core Services.
- From Administrative Tools, select Services, select Hyperion S9 BI+ x.x Core Services n, and click Start.

**Starting Subsets of Services**

You can start subsets of LSC and RSC services by inactivating those you do not want to start.

➤ To start a subset of Reporting and Analysis services:

1. **Inactivate services that you do not want to start:**
   - Interactive Reporting services—In `\BIPlus\common\config\server.xml`, set the runtype under each SERVICE node to start or hold.
   - LSC services—Using LSC, set Runtype to Hold for each service.
   - RSC services—In `\BIPlus\common\config\server.dat`, delete the names of services you want to inactivate. Before modifying this file, save a copy of the original.

   Details about `server.dat` are provided in “Starting Services and server.dat” on page 28.

2. **Run the start_CommonServices script or start Core Services.**

   See “Starting Reporting and Analysis Services” on page 26.
Starting Services and server.dat

When Core Services start, only RSC services listed in `\BIPlus\common\config\server.dat` are started.

Each line in `server.dat` is formatted as:

`serviceType:serviceName`

`serviceType` must be one of the strings shown in column one of this table:

<table>
<thead>
<tr>
<th>Service Type Field in <code>server.dat</code></th>
<th>Official Service Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sqribe.transformer.NameServerImpl</td>
<td>Name Service</td>
</tr>
<tr>
<td>com.sqribe.transformer.RepositoryAgentImpl</td>
<td>Repository Service</td>
</tr>
<tr>
<td>com.sqribe.transformer.MultiTypeServerAgentImpl</td>
<td>Event Service</td>
</tr>
<tr>
<td>com.sqribe.transformer.SQRJobFactoryImpl</td>
<td>Job Service</td>
</tr>
<tr>
<td>com.sqribe.transformer.ServiceBrokerImpl</td>
<td>Service Broker</td>
</tr>
</tbody>
</table>

The `serviceName` is the service name in the form:

`serviceAbbrev#_localHost`

where:

- `serviceAbbrev` is an abbreviation listed in “Managing Services” on page 144
- `#` is a number uniquely identifying the service
- `localHost` is the name of the computer where the service is installed, in the form of `hostname.domain.com`.

For example, to inactivate only Service Broker and Event Service on host apollo, remove these lines from `server.dat`:

```
com.sqribe.transformer.ServiceBrokerImpl:SB1_apollo.Hyperion.com
com.sqribe.transformer.MultiTypeServerAgentImpl:ES1_apollo.Hyperion.com
```

Starting Interactive Reporting Services Manager

You must start Interactive Reporting Services Manager individually. This is true whether the services are installed in an Install Home with the Reporting and Analysis services or alone in its own Install Home.

Note:

When you connect to a computer to start Interactive Reporting Services Manager on Windows, make sure the color property setting for the display is 16 bits or higher. If the color property setting is less than 16 bits, users may encounter extremely long response times when opening Chart sections of Interactive Reporting documents in Workspace. This is an important prerequisite, especially when starting the services remotely (for example using VNC, Terminal...
Services, Remote Administrator or Timbuktu, and so on), because many remote administration clients connect with only 8-bit colors by default.

➤ To start Interactive Reporting Services Manager:

1 In LSC, verify that Run Type for Interactive Reporting Services Manager is set to Start.
2 Start Core Services.
3 Start Interactive Reporting Services Manager.

Stopping Services

You stop all Reporting and Analysis services and services started individually by stopping their processes. Do so at each service’s host computer. In all cases, stopping the services constitutes a hard shutdown and causes the services to stop immediately. In the event of a hard shutdown, all work in progress stops.

The method for stopping a service must match how it was started:

- Services started with a start script—Run its stop script.

Caution!

Use a stop script only if the service was started with its start script. A stop script cannot be used to terminate one service within a multi-service process. The stop script stops all services running in that process.

Note:

On Windows systems, for Interactive Reporting Services Manager, type shutdown in the Interactive Reporting Services Manager console to stop Interactive Reporting services.

- Process running in a console window—Use a shut down command, such as shutdown or Ctrl+C on Windows. Using an operating system kill command (such as kill on UNIX) to stop the Reporting and Analysis services does not cause damage to the system; however, do not use kill -9.

- Windows service—Use the Stop command in the Services tool.

If you are running services as different servers (that is, as separate processes), you must stop Repository Service last.

Service Start-up Dependencies

Core Services, when launched using Windows Services, try for two minutes to launch; however, Core Services depend on other Hyperion components for normal startup in this order:

- Database
- Shared Services
Because of this order dependency, and because Windows imposes a two-minute maximum time limit for service startup, if your repository database runs on the same computer as Core Services, you may not want to validate service startup to avoid a system shut-down.

You select whether to validate service startup by setting the CHECK_SERVICE_STARTUP property in the \BIPlus\common\config\RMService8.properties file. Default is true.

If Core Services do not start, check stdout_console.log in \BIPlus\logs for information to help identify which dependent object is not available.

### Changing Service Port Assignments

*See the Hyperion Installation Start Here.*

### Starting the Workspace Servlet

Start the Workspace servlet according to the instructions given in your Web server documentation. Make the URL available to your system’s end-users.

For Workspace, enter the following URL:

http://hostname:port/workspace

where *hostname* is the name of the Workspace server, and *port* is the TCP port on which the application server is listening. The default port for Workspace is 19000 if using Apache Tomcat.

### Configuring Interactive Reporting Services Manager

You configure Interactive Reporting Services Manager properties in \BIPlus\common\config\server.xml. Interactive Reporting Services Manager manages up to *n* Interactive Reporting services:

- “Interactive Reporting Services Manager Event Thresholds” on page 30
- “Configuring Interactive Reporting Services” on page 33

### Interactive Reporting Services Manager Event Thresholds

You set threshold events to trigger Interactive Reporting Services Manager to stop and restart services. Threshold events are in server.xml in a property list called BQ_EVENT_MONITOR_PROPERTY_LIST. Set the first property, EVENT_MONITORING, to ON to enable threshold event usage. Comment out or delete the thresholds not in use.

<table>
<thead>
<tr>
<th>Property and Events</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVENT_MONITORING</td>
<td>Set to ON to use the following events</td>
</tr>
<tr>
<td>MAXIMUM_DOCUMENTS_THRESHOLD</td>
<td>Number of Interactive Reporting documents retrieved</td>
</tr>
<tr>
<td>Property and Events</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MAXIMUM_JOBS_THRESHOLD</td>
<td>Number of Interactive Reporting jobs run</td>
</tr>
<tr>
<td>MAXIMUM_UP_TIME_THRESHOLD</td>
<td>Total service running time since its first request</td>
</tr>
<tr>
<td>SPECIFIC_SHUTDOWN_THRESHOLD</td>
<td>Time of day that the service is not available, in minutes after midnight, for example, 150 means 2:30 A.M.</td>
</tr>
<tr>
<td>GRACE TIME</td>
<td>Time at which a process gracefully shuts down; default is Midnight</td>
</tr>
</tbody>
</table>

Sample Interactive Reporting Services Manager Information in server.xml

```xml
<SERVICES app="irj">
  <LSM>
    <fedName>
      <instanceid>00000113ba19c09e-0000-0f7e-aclb19a8</instanceid>
      <typeid>0ad72298-0000-0ccc-00000e750e616d5</typeid>
      <versionid>00000000-0000-0000-0000000000000000</versionid>
      <pathname>LSM_BI1_bi-win18.hyperion.com</pathname>
      <servicedesc>Generated service name for BI LSM</servicedesc>
    </fedName>
    <dbconfig>
      <username>biwin18</username>
      <password>9003f5bc66e8c9fbc16692ef880caadd</password>
      <dburl>jdbc:hyperion:oracle://bi-win6.hyperion.com:1521;SID=six1020</dburl>
      <dbdriver>hyperion.jdbc.oracle.OracleDriver</dbdriver>
    </dbconfig>
    <login>
      <username>Server1 on host bi-win18.hyperion.com</username>
      <password>768d2c1965b5f5345ed8b20ceeada7922</password>
    </login>
    <properties>
      <property name="directory">C:\barnes232\BIPlus\</property>
      <property name="GsmSyncTime">30</property>
      <property name="ServiceTestInterval">15</property>
      <property name="ServiceRegistrationInterval">15</property>
    </properties>
  </LSM>
  <service type="IRJob">
    <runtype>start</runtype>
    <fedName>
      <instanceid>00000113ba19c0bc-0000-0f7e-aclb19a8</instanceid>
      <typeid>00000107dd55157a-0000-12ba-ac1b10af</typeid>
      <versionid>00000000-0000-0000-0000000000000000</versionid>
      <pathname>BI1_bi-win18.hyperion.com</pathname>
      <servicedesc>BI1_bi-win18.hyperion.com</servicedesc>
    </fedName>
    <classInfo>
      <infotype>com.brio.one.services.brioquery.info.BrioQueryServiceInfo</infotype>
      <properties>
      </properties>
    </classInfo>
  </service>
</SERVICES>
```
The following property list defines the events that will be tracked by BI service in the evergreen mode. The first property enables event monitoring. If this property is ON, then even if no values are specified for the events, the BI service will start a graceful shutdown at midnight by default. To start the graceful shutdown process at a different time of day, refer to the comments on how to set the time.

```xml
<propertylist defid="0ad70321-0002-08aa-000000e738090110" name="BQ_EVENT_MONITOR_PROPERTY_LIST">
  <property defid="0ad70321-0001-08aa-000000e738090110" name="EVENT_MONITORING">OFF</property>
  <property defid="0ad70321-0001-08aa-000000e738090110" name="MAXIMUM_DOCUMENTS_THRESHOLD">10000</property>
  <property defid="0ad70321-0001-08aa-000000e738090110" name="MAXIMUM_JOBS_THRESHOLD">10000</property>
  <property defid="0ad70321-0001-08aa-000000e738090110" name="MAXIMUM_UP_TIME_THRESHOLD">180</property>
  <property defid="0ad70321-0001-08aa-000000e738090110" name="SPECIFIC_SHUTDOWN_TIME">1290</property>
  <property defid="0ad70321-0001-08aa-000000e738090110" name="GRACE_TIME">3</property>
</propertylist>

This property list is used to define the events that will be tracked by the BI service in the evergreen mode. The first property enables event monitoring. If this property is ON, then even if no values are specified for the events, the BI service will start a graceful shutdown at midnight by default. To start the graceful shutdown process at a different time of day, refer to the comments on how to set the time.

Max Concurrent IR Jobs To Run

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="MAX_CONCURRENT_JOB_REQUESTS">5</property>
```

Directory name where the service temporary files go

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="CACHE_LOCATION">C:\barnes232\BIPlus\data\cache</property>
```

Maximum number of concurrent requests this service supports. All other requests will be blocked.

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="MAX_CONCURRENT_REQUESTS">5000</property>
```

Minimum available disk space (Mb) required to service new request

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="MIN_DISK_SPACE">10</property>
```

Inactive time (seconds), after which document can be unloaded from memory to conserve resources

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="DOCUMENT_UNLOAD_TIMEOUT">900</property>
```

Document unloading mechanism gets activated when number of open documents exceed this number

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="DOCUMENT_UNLOAD_THRESHOLD">15</property>
```

Time (seconds) the polling thread sleeps between two polls

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="POLLING_INTERVAL">180</property>
```

Configuration file URL

```xml
<property defid="0ad70321-0001-08aa-000000e738090110" name="configurationURL">file:C:\barnes232\BIPlus\common\config\log4j\remoteServiceLog4jConfig.xml</property>
```
Configuring Interactive Reporting Services

You configure service type, event, and event threshold information in BIPlus\common\config\irmconfig.xml

In irmconfig.xml, there are multiple SERVICE nodes, one for each Interactive Reporting process:

- DataAccess
- BI
- IRJob
- IRLoggingUtility

Note:

Other service properties are configured using LSC. See “Modifying LSC Service Properties” on page 167.

<table>
<thead>
<tr>
<th>Configurable Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX_NUMBER_OF_MANAGED_SERVICES</td>
<td>Number of processes to manage</td>
</tr>
<tr>
<td></td>
<td>- BI and DataAccess—Only one managed process is active; other processes are in shutdown mode</td>
</tr>
<tr>
<td></td>
<td>- IRJob—All managed processes are started and active</td>
</tr>
<tr>
<td></td>
<td>- IRLoggingUtility—Only one managed process is active (maximum value is 1)</td>
</tr>
<tr>
<td>MONITOR_THREAD_INTERVAL</td>
<td>Interval for polling the internal status of the service in seconds. Minimum and default is 30, maximum is 300.</td>
</tr>
<tr>
<td>MONITOR_THREAD_TIMEOUT</td>
<td>Number of seconds the service is stopped if the polling is not working. Minimum and default is 300, maximum is 600.</td>
</tr>
<tr>
<td>HARD_SHUTDOWN_TIMEOUT</td>
<td>Number of seconds the process continues before a hard shutdown. Maximum and default is 30, no minimum.</td>
</tr>
<tr>
<td>GRACEFUL_SHUTDOWN_TIMEOUT</td>
<td>Number of seconds the process continues during a graceful shutdown. Allows a service to continue processing in the background. Default is 14400 (4 hours), minimum is 3600 (1 hour), maximum is 86400 (1 day).</td>
</tr>
<tr>
<td>IOR-FILE_NAME</td>
<td>Path to services generated data file. Default is BIPlus\logs\bijobs_ior.txt</td>
</tr>
<tr>
<td>SERVICES_STDOUT_FILE_PATH</td>
<td>Path to services standard output file location. Default is BIPlus\logs\bijobs_stdout.txt.</td>
</tr>
<tr>
<td>SERVICE_STDERR_FILE_PATH</td>
<td>Path to services standard error file location. Default is BIPlus\logs\bijobs_stderr.txt.</td>
</tr>
</tbody>
</table>
Sample irmconfig.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<IRMCONFIG>
  <SERVICE type="DataAccess">
    <!--DAS Configuration-->
    <!--This section is configured with defaults during system installation-->
    <!--Number of Services -->
    <PROPERTY name="MAX_NUMBER_OF_MANAGED_SERVICES" value="2" />
    <!-- Port Range, comma separated range, range is equal to MAX_NUMBER_OF_MANAGED_SERVICES -->
    <PROPERTY name="PORT_RANGE" value="6811,6812" />
    <!-- service monitoring interval (seconds) -->
    <PROPERTY name="MONITOR_THREAD_INTERVAL" value="30" />
    <!-- service monitoring timeout duration (seconds) -->
    <PROPERTY name="MONITOR_THREAD_TIMEOUT" value="300" />
    <!-- time service has to live on a hard kill (seconds) -->
    <PROPERTY name="HARD_SHUTDOWN_TIMEOUT" value="30" />
    <!-- time service has to live while in graceful shutdown state (seconds, 86400 is 1 day) -->
    <PROPERTY name="GRACEFUL_SHUTDOWN_TIMEOUT" value="86400" />
    <!-- generated data file location -->
    <PROPERTY name="IOR_FILE_NAME" value="C:\barnes232\BIPlus\logs\das_ior.txt" />
    <!-- service standard output file location -->
    <PROPERTY name="SERVICE_STDOUT_FILE_PATH" value="C:\barnes232\BIPlus\logs\das_stdout.txt" />
    <!-- service standard error file location -->
    <PROPERTY name="SERVICE_STDERR_FILE_PATH" value="C:\barnes232\BIPlus\logs\das_stderr.txt" />
  </SERVICE>
  <SERVICE type="BI">
    <!--BI Configuration-->
    <!--This section is configured with defaults during system installation-->
    <!--Number of Services -->
    <PROPERTY name="MAX_NUMBER_OF_MANAGED_SERVICES" value="2" />
    <!-- Port Range, comma separated range, range is equal to MAX_NUMBER_OF_MANAGED_SERVICES -->
    <PROPERTY name="PORT_RANGE" value="6813,6814" />
    <!-- service monitoring interval (seconds) -->
    <PROPERTY name="MONITOR_THREAD_INTERVAL" value="30" />
    <!-- service monitoring timeout duration (seconds) -->
    <PROPERTY name="MONITOR_THREAD_TIMEOUT" value="300" />
    <!-- time service has to live on a hard kill (seconds) -->
    <PROPERTY name="HARD_SHUTDOWN_TIMEOUT" value="30" />
    <!-- time service has to live while in graceful shutdown state (seconds, 86400 is 1 day) -->
    <PROPERTY name="GRACEFUL_SHUTDOWN_TIMEOUT" value="86400" />
    <!-- generated data file location -->
    <PROPERTY name="IOR_FILE_NAME" value="C:\barnes232\BIPlus\logs\bi_ior.txt" />
    <!-- service standard output file location -->
    <PROPERTY name="SERVICE_STDOUT_FILE_PATH" value="C:\barnes232\BIPlus\logs\bi_stdout.txt" />
    <!-- service standard error file location -->
    <PROPERTY name="SERVICE_STDERR_FILE_PATH" value="C:\barnes232\BIPlus\logs\bi_stderr.txt" />
  </SERVICE>
  <SERVICE type="IRJob">
    <!--IRJOB Configuration-->
    <!--This section is configured with defaults during system installation-->
    <!--Number of Services -->
    <PROPERTY name="MAX_NUMBER_OF_MANAGED_SERVICES" value="2" />
    <!-- Port Range, comma separated range, range is equal to MAX_NUMBER_OF_MANAGED_SERVICES -->
    <PROPERTY name="PORT_RANGE" value="6811,6812" />
    <!-- service monitoring interval (seconds) -->
    <PROPERTY name="MONITOR_THREAD_INTERVAL" value="30" />
    <!-- service monitoring timeout duration (seconds) -->
    <PROPERTY name="MONITOR_THREAD_TIMEOUT" value="300" />
    <!-- time service has to live on a hard kill (seconds) -->
    <PROPERTY name="HARD_SHUTDOWN_TIMEOUT" value="30" />
    <!-- time service has to live while in graceful shutdown state (seconds, 86400 is 1 day) -->
    <PROPERTY name="GRACEFUL_SHUTDOWN_TIMEOUT" value="86400" />
    <!-- generated data file location -->
    <PROPERTY name="IOR_FILE_NAME" value="C:\barnes232\BIPlus\logs\ir_job_ior.txt" />
    <!-- service standard output file location -->
    <PROPERTY name="SERVICE_STDOUT_FILE_PATH" value="C:\barnes232\BIPlus\logs\ir_job_stdout.txt" />
    <!-- service standard error file location -->
    <PROPERTY name="SERVICE_STDERR_FILE_PATH" value="C:\barnes232\BIPlus\logs\ir_job_stderr.txt" />
  </SERVICE>
</IRMCONFIG>
```
<!--This section is configured with defaults during system installation-->

<!----Number of Services -->

<PROPERTY name="MAX_NUMBER_OF_MANAGED_SERVICES" value="1" />

<!-- Port Range, comma separated range, range is equal to

MAX_NUMBER_OF_MANAGED_SERVICES -->

<PROPERTY name="PORT_RANGE" value="6815" />

<!-- service monitoring interval (seconds) -->

<PROPERTY name="MONITOR_THREAD_INTERVAL" value="30" />

<!-- service monitoring timeout duration (seconds) -->

<PROPERTY name="MONITOR_THREAD_TIMEOUT" value="300" />

<!-- time service has to live on a hard kill (seconds) -->

<PROPERTY name="HARD_SHUTDOWN_TIMEOUT" value="30" />

<!-- time service has to live while in graceful shutdown state (seconds, 86400 is 1 day) -->

<PROPERTY name="GRACEFUL_SHUTDOWN_TIMEOUT" value="86400" />

<!-- generated data file location -->

<PROPERTY name="IOR_FILE_NAME" value="C:\barnes232\BIPlus\logs\bijobs_ior.txt" />

<!-- service standard output file location -->

<PROPERTY name="SERVICE_STDOUT_FILE_PATH" value="C:\barnes232\BIPlus\logs\bijobs_stdout.txt" />

<!-- service standard error file location -->

<PROPERTY name="SERVICE_STDERR_FILE_PATH" value="C:\barnes232\BIPlus\logs\bijobs_stderr.txt" />

</SERVICE>

<SERVICE type="IRLoggingUtility">

<!--IRLoggingUtility Configuration-->

<!--This section is configured with defaults during system installation-->

<!----Number of Services -->

<!-- Port Range, comma separated range, range is equal to

MAX_NUMBER_OF_MANAGED_SERVICES -->

<PROPERTY name="PORT_RANGE" value="6810" />

<!-- service monitoring interval (seconds) -->

<PROPERTY name="MONITOR_THREAD_INTERVAL" value="30" />

<!-- service monitoring timeout duration (seconds) -->

<PROPERTY name="MONITOR_THREAD_TIMEOUT" value="300" />

<!-- time service has to live on a hard kill (seconds) -->

<PROPERTY name="HARD_SHUTDOWN_TIMEOUT" value="30" />

<!-- time service has to live while in graceful shutdown state (seconds, 86400 is 1 day) -->

<PROPERTY name="GRACEFUL_SHUTDOWN_TIMEOUT" value="60" />

<!-- generated data file location -->

<PROPERTY name="IOR_FILE_NAME" value="C:\barnes232\BIPlus\logs\logutil_ior.txt" />

<!-- service standard output file location -->

<PROPERTY name="SERVICE_STDOUT_FILE_PATH" value="C:\barnes232\BIPlus\logs\logutil_stdout.txt" />

<!-- service standard error file location -->

<PROPERTY name="SERVICE_STDERR_FILE_PATH" value="C:\barnes232\BIPlus\logs\logutil_stderr.txt" />

</SERVICE>

</IRMCONFIG>
Securing Your Workspace Production Environment

For security purposes, administrators can disable these Workspace features as they are primarily used for test and debugging purposes:

- **Configuration URL**—http://hostname:19000/workspace/browse/configInfo (see “Workspace Configuration Information” on page 22) provides details about system configuration and requires administrative credentials to access.

  To disable this page, remove or rename the file jsp/shared/configInfo.jsp in the deployed Workspace Web application.

- **Workspace test module**—cds.test contains test cases and debugging utilities and is accessible when Workspace is running in debug mode.

  To disable this module, rename or delete the directory /modules/com/hyperion/tools/cds/repository/bpm/test from the deployed Workspace Web application.

- **Troubleshooting code**—Workspace ships with uncrunched JavaScript code for troubleshooting purposes. To remove this code so that it cannot be accessed, backup the directory /js/com in the workspace_static directory. Then, for each subdirectory of /js/com remove each JS file except for the file <directory-name>.js.

  For example, remove all files in /js/com/hyperion/bpm/web/common except for /js/com/hyperion/bpm/web/common/Common.js.

- **Client-side debug mode**—The ClientDebugEnabled parameter (in /WEB-INF/conf/BpmServer.properties) enables administrators to disable debug mode, which filters out debug and log request parameters. This parameter is case-insensitive and defaults to true. If this parameter is set to anything other than true, or if it is missing from the file, debug mode is disabled.

Quick Guide to Common Administrative Tasks

Use this section to quickly locate instructions for some common administrative tasks. Table 1 lists tasks involved in initially configuring and populating your system, and the system component used for each task. Table 2 gives this information for tasks involved in maintaining a system. These tables do not include all tasks covered in this guide.

<table>
<thead>
<tr>
<th>Task</th>
<th>Component</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start or stop a server</td>
<td></td>
<td>“Starting and Stopping Services” on page 25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“Stopping Services” on page 29</td>
</tr>
<tr>
<td>Provision users, groups, and roles</td>
<td>User Management Console</td>
<td>Hyperion Security Guide</td>
</tr>
<tr>
<td>Configure generated Personal Page</td>
<td>Explore module</td>
<td>“Configuring the Generated Personal Page” on page 136</td>
</tr>
<tr>
<td>Configure Broadcast Messages</td>
<td>Explore module</td>
<td>“Understanding Broadcast Messages” on page 137</td>
</tr>
</tbody>
</table>

36 Administration Tools and Tasks
<table>
<thead>
<tr>
<th>Task</th>
<th>Component</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide optional Personal Page content</td>
<td>Explore module</td>
<td>“Providing Optional Personal Page Content to Users” on page 139</td>
</tr>
<tr>
<td>Provide graphics for bookmarks</td>
<td>Explore module</td>
<td>“Configuring Graphics for Bookmarks” on page 139</td>
</tr>
<tr>
<td>Create custom calendars for scheduling jobs</td>
<td>Calendar Manager</td>
<td>“Creating Calendars” on page 124</td>
</tr>
<tr>
<td>Create public job parameters</td>
<td>Schedule module</td>
<td>“Administering Public Job Parameters” on page 128</td>
</tr>
<tr>
<td>Create or modify printers or directories for job output</td>
<td>Administrate module</td>
<td>“Managing Physical Resources” on page 42</td>
</tr>
<tr>
<td>Define database servers</td>
<td>RSC</td>
<td>“Adding Database Servers” on page 155</td>
</tr>
<tr>
<td>Configure services</td>
<td>RSC, LSC</td>
<td>Chapter 7, “Configuring RSC Services”, Chapter 8, “Configuring LSC Services”</td>
</tr>
<tr>
<td>Set system properties</td>
<td>Administrate module</td>
<td>“Setting General Properties” on page 40</td>
</tr>
<tr>
<td>Configure servlets</td>
<td>Servlet Configurator</td>
<td>Chapter 9, “Configuring the Servlets”</td>
</tr>
</tbody>
</table>

**Table 2  System Maintenance Tasks**

<table>
<thead>
<tr>
<th>Task</th>
<th>Component</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change which services run in a server</td>
<td>RSC, LSC</td>
<td>“Starting Services and server.dat” on page 28</td>
</tr>
<tr>
<td>Modify services</td>
<td>RSC, LSC</td>
<td>Chapter 7, “Configuring RSC Services”, Chapter 8, “Configuring LSC Services”</td>
</tr>
<tr>
<td>Modify Job Service</td>
<td>RSC</td>
<td>“Managing Jobs” on page 159</td>
</tr>
<tr>
<td>Modify system properties</td>
<td>Administrate module</td>
<td>“Setting General Properties” on page 40</td>
</tr>
<tr>
<td>Delete services</td>
<td>RSC or the Installation program</td>
<td>Chapter 7, “Configuring RSC Services” or the Hyperion Reporting and Analysis – System 9 Installation Guides</td>
</tr>
<tr>
<td>Modify users, groups, or roles</td>
<td>User Management Console</td>
<td>Hyperion Security Guide</td>
</tr>
<tr>
<td>Inactivate obsolete users,</td>
<td>User Management Console</td>
<td>Hyperion Security Guide</td>
</tr>
<tr>
<td>Create MIME types</td>
<td>Administrate module</td>
<td>“Defining MIME Types” on page 44</td>
</tr>
<tr>
<td>Modify MIME types</td>
<td>Administrate module</td>
<td>“Modifying MIME Types” on page 45</td>
</tr>
<tr>
<td>Inactivate obsolete MIME types</td>
<td>Administrate module</td>
<td>“Inactivating or Re-activating MIME Types” on page 45</td>
</tr>
<tr>
<td>Add hosts</td>
<td>RSC</td>
<td>“Adding Hosts” on page 153</td>
</tr>
<tr>
<td>Add services</td>
<td>installation program</td>
<td>Hyperion Reporting and Analysis – System 9 Installation Guides</td>
</tr>
<tr>
<td>Configure common metadata services</td>
<td>Administrate module</td>
<td>“Host Shared Services Properties” on page 174</td>
</tr>
</tbody>
</table>
Overview

The Administer module, available from the Workspace Navigate menu, enables you to manage Workspace properties, performance, and user interaction. Toolbar buttons represent Administer menu items:

<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Navigate &gt; Administer Menu Item</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image]</td>
<td>General Properties</td>
<td>Define general system and user interface properties</td>
</tr>
<tr>
<td>![Image]</td>
<td>User Management</td>
<td>Provision users, groups, and roles</td>
</tr>
<tr>
<td>![Image]</td>
<td>Physical Resources</td>
<td>Specify printers and output directories for job output</td>
</tr>
<tr>
<td>![Image]</td>
<td>MIME Types</td>
<td>Create, modify, and delete Workspace MIME types</td>
</tr>
<tr>
<td>![Image]</td>
<td>Notifications</td>
<td>Define mail server properties and how end users receive e-mail notifications about jobs</td>
</tr>
<tr>
<td>![Image]</td>
<td>SmartCuts</td>
<td>Specify how to construct SmartCuts (shortcuts to imported documents in Workspace) for inclusion in e-mail notifications</td>
</tr>
</tbody>
</table>
### Setting General Properties

To set general and user interface properties:

1. Select **Navigate > Administer > General**.
2. Modify properties.
3. Click **Save Properties**.

#### General Properties

- **System Name**—Distinguishes the current installation from other Reporting and Analysis installations (*Installation* is defined as a system served by one GSM.)
- **Broadcast Messages**—Specifies the folder in which to store broadcast messages
- **Enable users to use Subscription and Notification**—Activates import event logging, which enables Event Service to identify subscription matches and notify users of changes in subscribed items (Effective Date—when logging begins)
- **Enable Priority Ratings**—Enables users to set priority ratings on items imported to the Explore module.
- **Enable Harvesting**—Activates Harvester Service, which enables users to use Impact Manager to extract and save Interactive Reporting metadata to relational data sources for use in other formats (see Chapter 4, “Using Impact Management Services”)
- **Default Open Format for Interactive Reporting**—Specifies whether Interactive Reporting documents open in Workspace (HTML) or Interactive Reporting Web Client by default

---

<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Navigate &gt; Administer Menu Item</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Row-level Security</td>
<td>Manage row-level security settings in data sources used by Interactive Reporting documents</td>
</tr>
<tr>
<td></td>
<td>Usage Tracking</td>
<td>Track system usage and define related properties</td>
</tr>
<tr>
<td></td>
<td>Event Tracking</td>
<td>Track events, such as document opens, documents closes for selected MIME types, and jobs run</td>
</tr>
</tbody>
</table>

**Note:**

You can use various methods to perform most Administer module tasks. For a complete list of all toolbars, menus, and shortcut menus, see the *Hyperion Workspace User's Guide*. 

---

40 Administer Module
**User Interface Properties**

- **Display all users/groups/roles in the system**—Lists all available users, groups, and roles when end users set access permissions on repository items. Selecting this option may impact system performance.

- **List up to \( nn \) users/groups/roles**—Number of users, groups, or roles displayed when end users set access permissions on repository items; default is 100. Specifying too low a number may prevent end users from seeing all users, groups, and roles to which they have access.

**Managing Users**

For information on managing users, groups, and roles, see the *Hyperion Security Guide*.

**Assigning Reporting and Analysis Default Preferences**

User Management Console enables users with Provisioning Manager and Explorer roles to set the default folder, new document folder, and start page application preferences for users and groups. Individual and group preferences have precedence over default preferences.

For default preferences to succeed, users and groups must have the roles and permissions necessary to access specified folders and interface elements.

*Provisioning Manager* is a Shared Services role that enables you to provisions users and groups to applications. It is also required for setting default user preferences for Reporting and Analysis. For more information, see the *Hyperion Security Guide*.

➤ To assign default preferences for Reporting and Analysis:

1. **Select Navigate > Administer > User Management.**
2. **Log on to User Management Console with a user name provisioned with Provisioning Manager and Explorer roles.**
3. **Expand the Projects node until a Reporting and Analysis application is displayed.**
4. **Right-click the application name and select Assign Preferences.**
   
   A three-step wizard is displayed in the Process bar.

5. **For step 1 of the Wizard, Select Users, select Available Users or Available Groups.**
6. **From the left panel, select user names or group names and click the right arrow.**
   
   To select consecutive names, select the first name, press and hold down Shift, and select the last name. To select names that are not consecutive, press and hold down Ctrl, and select each item. Use Add All to select all names.

7. **Repeat steps 5 and 6 to select a combination of users and groups.**
8. **When all user and group names are displayed in Selected Users and Groups, click Next.**
9. **For step 2 of the Wizard, Manage Preferences, specify these default preferences for the selected users and groups:**
- **Default Folder**—Repository location of the default folder.
- **New Document Folder**—Default folder in which the new document wizard searches for valid data sources, that is, Web Analysis database connection files and Interactive Reporting documents.
- **Start Page**—Reporting and Analysis interface displayed after logging on. Select None, Explore, Document, Favorite, or Scorecard.
  
  If you select Explore or Document for Start Page, you must specify a repository location.

10 **When all preferences are specified**, click Next.

11 **For step 3 of the Wizard, Finish**, choose between three tasks:
  
  - To configure options for another application, select one from the View pane.
  - To change preferences for currently selected users and groups, click Back.
  - To specify another set of users and groups and set their preferences, click Continue.

### Managing Physical Resources

Physical resources, such as printers and directories, are used as destinations for Interactive Reporting and SQR Production Reporting job output. Physical resources must be accessible to each server that is running Hyperion Interactive Reporting Service.

You should assign access permissions and notify end users about which physical resources to use. Users should see only the physical resources that they can use.

### Viewing Physical Resources

➤ **To view physical resources defined for Workspace:**

1 **Select Navigate > Administer > Physical Resources.**

2 From Display, select All, Only Printer, or Only Output Directory and click Update List.

➤ **To view properties settings for physical resources, click a resource name.**

### Access Permissions for Physical Resources

Unlike other Reporting and Analysis artifacts, which offer several access levels, physical resources have only three access levels: No Access, View, and Full Control.

You add roles, groups, or users and set their access permissions as you do for other artifacts. See the Hyperion Workspace User’s Guide for instructions on setting access permissions.
Adding Physical Resources

To add physical resources:

1. Select Navigate > Administer > Physical Resources.
2. In the Content pane, click Go next to Add Printer or Add Output Directory.
3. Specify required properties and optional properties.
   See “Printer Properties” on page 43 and “Output Directory Properties” on page 44.

Note:
Physical resources must be accessible to each server on which Hyperion Interactive Reporting Service is running.
4. Set access Permissions for this resource (see “Access Permissions for Physical Resources” on page 42).
5. Click Finish.

Modifying Physical Resources

To modify physical resources:

1. Select Navigate > Administer > Physical Resources.
2. Click Modify next to a resource name or select the resource name.
3. Make changes and click OK.
   See “Access Permissions for Physical Resources” on page 42, “Printer Properties” on page 43 and “Output Directory Properties” on page 44.

Deleting Physical Resources

To delete physical resources:

1. Select Navigate > Administer > Physical Resources.
2. Click Delete next to a resource name.
3. Confirm the deletion when prompted.

Printer Properties

Printers are used for Interactive Reporting job output:

- **Type**—Read-only property; set as *Printer*
- **Name**—Name for the printer; visible to end users
- **Description**—Helps administrators and end users identify the printer
- Printer Address—Network address of the printer (for example, `\f3prt\techpubs`); not visible to end users

**Output Directory Properties**

Output directories are used for Interactive Reporting and SQR Production Reporting job output. They can be located locally or on a network and can be FTP directories:

- **General properties:**
  - **Type**—Read-only property; set as `Output Directory`.
  - **Name**—Name for the output directory; visible to end users
  - **Description**—Helps administrators and end users identify the directory
  - **Path**—Directory's full network path (for example, `\apollo\Inventory_Reports`)

- **FTP properties:**
  - **Directory is on FTP Server**—Enable if the output directory is located on an FTP server, and set these options:
    - **FTP server address**—Address of the FTP server where the output directory is located (for example, `ftp2.hyperion.com`)
    - **FTP User Name**—User name to access the FTP output directory
    - **FTP Password**—Password for FTP user name
    - **Confirm Password**—Retype the password entered for FTP password

**Managing MIME Types**

Before you can import items into the repository, their MIME types must be defined in Workspace. Although Workspace has many built-in MIME types, you may need to define others.

You can associate a MIME type with multiple file extensions. For example, you can associate the extensions `.txt`, `.bat`, and `.dat` with the `text` MIME type.

Multiple MIME types can use one extension. For example, if your organization uses multiple versions of a program, you can define a MIME type for each version; however, file names of all versions use the same extension. When users opens files with extensions that belong to multiple MIME types, they are prompted to select a program executable.

In the MIME type list, traffic-light icons indicate active (green) or inactive (red); see “Inactivating or Re-activating MIME Types” on page 45.

**Defining MIME Types**

➤ To define MIME types:

1. Select Navigate > Administer > MIME Types.
2 At the bottom of the content pane, click Go (to the right of Add MIME Type).
3 Supply a name and description.

4 In the file extensions box, enter an extension and click .
When entering extensions, type only the extension letters. Do not include a period (.).

5 Optional: Under Advanced Options, click Change Icon, and select from the available images or browse to add an image to the repository, then click OK.

6 Click Finish.

Note:
Newly defined MIME types are active by default.

Modifying MIME Types

➤ To modify MIME types:
1 Select Navigate > Administer > MIME Types.
2 In the listing of MIME types, click Modify.
3 Change properties.

To remove a file extension, select it in the <Extensions> list and click .

4 Click OK.

Inactivating or Re-activating MIME Types

To prevent items from being imported to the repository, inactivate their MIME types. Although repository items with inactive MIME types are still accessible, end users must specify which programs to use when opening them.

You can re-activate an inactive MIME type at any time.

➤ To inactivate or re-activate MIME types:
1 Select Navigate > Administer > MIME Types.
2 In the MIME type list, click Modify Properties.
3 Change the Active setting:
   ● To inactivate a MIME type, clear Active and click OK. Its traffic-light icon changes to red.
   ● To re-activate a MIME type, select Active and click OK. Its traffic-light icon changes to green.
Deleting MIME Types

Unlike inactivating MIME types, deletion is permanent and affects associated items. You cannot import files that have extensions associated with a deleted MIME type.

For items associated with a deleted MIME type, the text “unknown file type” is displayed instead of MIME type icons. When users open these items, they are prompted to select a program executable.

You can delete MIME types that you define; however, you cannot delete built-in Workspace MIME types.

➤ To delete MIME types:

1 Select Navigate > Administer > MIME Types.
2 Click Delete next to a MIME type.

Managing Notifications

Notification properties control how users receive notifications about the jobs and documents to which they subscribe:

- “Understanding Subscriptions and Notifications” on page 46
- “Modifying Notification Properties” on page 48

Understanding Subscriptions and Notifications

Subscriptions and notifications are handled by Event Service.

Topics that discuss how Event Service handles subscriptions and notifications:

- “Subscription Types” on page 46
- “How Event Service Obtains Information” on page 47
- “Notification Mechanisms” on page 47

Subscription Types

Subscription types that users can subscribe to and receive notifications about:

- New or updated versions of items
- Changed content in folders
- Job completion
- Job exceptions

Independent of subscriptions, Event Service sends notifications to these users:

- Owners of scheduled jobs, when job execution finishes
Users who run background jobs, when job execution finishes

**How Event Service Obtains Information**

When users subscribe to items or folders, Workspace sends subscription information through LSM to Event Service, which adds the subscriptions to its subscriptions list.

Repository Service maintains a list of imported and updated artifacts, which includes all imported items, folders, and job output; modified item properties; updated versions; and artifact metadata.

Repository Service includes in its list both imported or modified items or folders, and the folders that contain them.

Every 60 seconds, Event Service obtains the Repository Service’s list of new and modified items, and compares them to the subscription list. Event Service then sends notifications to subscribed users.

Repository Service discards its list after giving it to Event Service, which, in turn, discards the list after it notifies subscribers of changes.

Other services notify Event Service when they complete actions that may trigger subscriptions, such as successful job execution. Event Service checks these events against the subscription list and sends notifications to subscribers.

**Notification Mechanisms**

Ways in which Event Service notifies users:

- Send e-mails with embedded SmartCuts to notify users about changes to items, folders, new report output, job completion, or exception occurrences
  
  Optionally, Event Service may send file attachments, based on how users chose to be notified on the Subscribe page.

- Display notifications of completed scheduled jobs or background jobs in the Schedule module

- Display notification of job completion after a job runs in the foreground

- Display a redlight icon in Exceptions Dashboard when `output.properties` indicates that exceptions occurred
  
  When exceptions occur, the importer of the file sets properties to indicate the presence of exceptions and to specify exception messages. The importer is usually Job Service, and the file is usually job output.

  Exceptions can be flagged by any of these methods:

  - SQR Production Reporting code
  - Manually by users who import files or job output
  - APIs that set exception properties on files or output
Hyperion Interactive Reporting Service does not support exceptions, but you can set exceptions on Interactive Reporting documents using the API or manual methods.

Users choose whether to include the Exceptions Dashboard on Personal Pages and which jobs to include on the Exceptions Dashboard.

**Modifying Notification Properties**

To modify Notification properties:

1. Select **Navigate > Administer > Notifications**.
2. Modify Notification properties and mail server options:

**Note:**

If you change the “Enable email attachment,” “Maximum attachment size,” “Mail server host name for sending email notifications,” or “Email account name for sending e-mail notifications” property, you must restart Core Services for the setting to take effect. For information on starting services, see “Starting Reporting and Analysis Services” on page 26.

- **Notifications**
  - **Enable email attachment**—Allows end users to send file attachments with their e-mail notifications. If jobs generate only one output file, that file is attached to the e-mail. If jobs generate multiple output files including PDF files, the PDF files are attached to e-mails; otherwise, no files are attached.
  - **Maximum attachment size**—Maximum allowed size for attachments, in bytes.
  - **Time to live for entries in the notification log**—Number of minutes after which events are removed from the notification log and are no longer displayed in the Explore module. Expiration times for scheduled jobs and background jobs.

- **Mail Server Options**
  - **Mail server host name for sending email notifications**
  - **Email account name for sending email notifications**

**Note:**

To send e-mails with embedded SmartCuts, you must also set SmartCut properties.

- **Require authentication**—Makes authentication (ASMT) mandatory. Enter user name and password when enabled. Default is disabled.

After specifying notification properties, you can click **Send Test E-mail** to view your mail server entries and enter a destination e-mail address.

3. Click **Save Properties**.
Managing SmartCuts

SmartCuts are shortcuts in URL form to imported documents in Workspace. SmartCut properties are used to construct SmartCuts that are included in e-mail notifications.

URLs for SmartCuts:
http://hostname:port/workspace/browse/get/Smartcut

For Example:
http://pasts402:19000/workspace/browse/get/Patty/Avalanche_CUI_Style_Guidelines.pdf/

(Alternatively, a SmartCut may start with https instead of http.)

➤ To modify SmartCut properties:

1 Select Navigate > Administer > SmartCuts.
2 Modify SmartCut properties:

Note:
If you change SmartCut properties, you must restart the Workspace server and Job Service for the settings to take effect.

- Name—Web component for the SmartCut
- Description—Workspace description
- Host—Host on which UI Services reside
- IP Port—Port number on which Workspace runs
- Root—Web application deployment name for Workspace, as set in your Web server software
  Typically, this is workspace/browse. The last segment (browse) must match the servlet name specified during installation.
- Protocol for SmartCuts generated in email notifications—HTTP or HTTPS
3 Click Save Properties.

Managing Row-Level Security

Row-level security enables users to view only those records that match their security profile, no matter what their search criteria. It enables administrators to tag data in the row level of a database, thus controlling who has read access to information. Row-level security is critical to applications that display sensitive data such as employee salaries, sales commissions, or customer details. Lack of row-level security could be a big detriment to organizations that want to distribute information to their user community using the Internet and intranets.
If you want to implement row-level security in Reporting and Analysis, keep these points in mind:

- At least one Hyperion Interactive Reporting Data Access Service instance must be configured to access the data source storing your row-level security information.
- The database client library should be installed on the computer where the Hyperion Interactive Reporting Data Access Service is running.
- The data source for the Reporting and Analysis repository that has the row-level security table information should be configured.
- For security reasons, the user name and password to access the data source should differ from that used for the Reporting and Analysis user account.

See the *Hyperion Interactive Reporting Studio User’s Guide* for information about implementing row-level security in Interactive Reporting documents.

Row-level security properties are stored in the repository; however, the *rules* about how to give access to the data are stored in the data source.

To modify row-level security properties:

1. Select **Navigate > Administer > Row Level Security**.
2. Modify these row-level-security properties:
   - **Enable Row Level Security**—Row-level security is disabled by default.
   - **Connectivity**—Database connectivity information for reports’ source data.
   - **Database Type**—Type of database that you are using. Database types available depend on connectivity selection.
   - **Data Source Name**—Host of the data source database.
   - **User Name**—Default database user name used by Job Service for running SQR Production Reporting jobs on this database server; used for jobs that were imported with no database user name and password specified.
   - **Password**—Valid password for *user name*.
3. Click **Save Properties**.

**Tracking System Usage**

Usage tracking records information about Workspace activities as they occur and provides a historical view of system usage. This information answers questions like:

- Who logged in yesterday?
- Which Workspace reports are accessed most frequently?

You can configure your system to track numerous activities. For example, you can track opening, closing, and processing Interactive Reporting documents or you can track only opening Interactive Reporting documents.
Activities are recorded as events in the repository database. Events are recorded with pertinent details and information that distinguishes them from each other. Event times are stored in GMT. Events are deleted from the database in a configurable time frame.

Usage Service must be running to track events set in the user interface. Usage Service can be replicated and all Usage Services access one database.

The user name and password to access the usage tracking information may differ from that used for Workspace. Hyperion recommends that Usage Tracking use its own schema in the repository database; however, an alternate schema is not required. For more information about configuring Usage Tracking schema, see the *Hyperion Reporting and Analysis – System 9 Installation Guides*.

Topics that provide detailed information about tracking usage and events:

- “Managing Usage Tracking” on page 51
- “Tracking Events and Documents” on page 51
- “Sample Usage Tracking Reports” on page 52

### Managing Usage Tracking

Usage tracking is managed through the Administer module and LSC. All configurable properties, except run type, are managed in the Administer module. To modify the run type, see “Common LSC Properties” on page 167.

To manage usage tracking:

1. **Select Navigate > Administer > Usage Tracking.**
2. **Change these properties:**
   - **General preferences**
     - **Usage Tracking Active**—Select to turn on usage tracking.
     - **Mark records ready for deletion after_days**—Number of days after which usage tracking events should be marked for deletion by the garbage collection utility. Default is 30.
     - **Delete records every_days**—Number of days after which the garbage collection utility should be run. Default is 7.
   - **Connectivity preferences**—User name and password are populated from the usage tracking database and should only be changed if the database is moved.
3. **Select Apply.**

### Tracking Events and Documents

Usage Service keeps records about logon instances, document opens, documents closes for select MIME types, jobs run, job output views, and queries processed by Reporting and Analysis. Usage Service must be running to track events. By default, events are not tracked.
To track events:

1 Select Navigate > Administer > Event Tracking.

2 Select an event to track it:
   - System Logons
   - Database Logons
   - Timed Query Event
   - Open Interactive Reporting Document
   - Process Interactive Reporting Document
   - Close Interactive Reporting Document
   - Run Interactive Reporting Job
   - View Interactive Reporting Job Output
   - Run SQR Production Reporting Job
   - View SQR Production Reporting Job Output
   - Run Generic Job
   - View Generic Job Output

3 To track documents, move one or more available MIME types to the Selected MIME Types list.
   Tracking occurs each time a document of a selected MIME type is opened.

4 Click Apply.

Sample Usage Tracking Reports

Sample usage tracking reports provide immediate access to standard Workspace usage reports. You can modify standard reports or create your own reports.

The Interactive Reporting document, `sample_usage_tracking.bqy`, which generates usage tracking reports, is in the \Root\Administration folder in Explore.

To view the Administration folder, from Explore, select View > Show Hidden.

Caution!

The sample reports could contain sensitive company information when used with your data. Use access permissions when importing the reports so only the intended audience has access.
# Using Impact Management Services

## Introduction

Impact Management Services, introduced with the Impact Manager module, enable you to collect and report on metadata, and to update the data models that imported documents use. Impact Management Assessment Services and Impact Management Update Services perform these tasks. Task results are displayed in Show Task Status and Show Impact of Change, and managed in Manage Task List.

## In This Chapter

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>53</td>
</tr>
<tr>
<td>About Impact Management Services</td>
<td>54</td>
</tr>
<tr>
<td>Impact Management Assessment Services</td>
<td>54</td>
</tr>
<tr>
<td>About Impact Management Metadata</td>
<td>54</td>
</tr>
<tr>
<td>Impact Management Update Services</td>
<td>55</td>
</tr>
<tr>
<td>Impact Manager Module Size and Time Guidelines</td>
<td>57</td>
</tr>
<tr>
<td>Running the Update Services</td>
<td>58</td>
</tr>
<tr>
<td>Script Logging Levels</td>
<td>58</td>
</tr>
<tr>
<td>Access to Impact Management Services</td>
<td>59</td>
</tr>
<tr>
<td>Synchronize Metadata Feature</td>
<td>59</td>
</tr>
<tr>
<td>Update Data Models Feature</td>
<td>61</td>
</tr>
<tr>
<td>Data Model Updates and Jobs</td>
<td>65</td>
</tr>
<tr>
<td>JavaScript Update Feature</td>
<td>65</td>
</tr>
<tr>
<td>Custom Update Feature</td>
<td>69</td>
</tr>
<tr>
<td>Using the SortDataModelTopics Script</td>
<td>69</td>
</tr>
<tr>
<td>Advanced Scripting</td>
<td>73</td>
</tr>
<tr>
<td>Connecting Interactive Reports</td>
<td>105</td>
</tr>
<tr>
<td>Using Show Task Status List</td>
<td>107</td>
</tr>
<tr>
<td>Managing Tasks</td>
<td>109</td>
</tr>
<tr>
<td>Using Show Impact of Change Interactive Report</td>
<td>113</td>
</tr>
<tr>
<td>Creating New Data Models</td>
<td>115</td>
</tr>
<tr>
<td>Changing Column Data Types</td>
<td>120</td>
</tr>
<tr>
<td>Compatible Replacement Data Model and Document Character Sets</td>
<td>121</td>
</tr>
<tr>
<td>Changing User Names and Passwords for Interactive Reporting Documents</td>
<td>121</td>
</tr>
<tr>
<td>Service Configuration Parameters</td>
<td>122</td>
</tr>
</tbody>
</table>
The Impact Management Services facility aims to eliminate the breakage of queries and their reports caused by changes to the database structures that they rely on. Release 9.3 and later enables technical users to develop an impact improvement mechanism to respond to arbitrary changes; for example, re-branding of dashboards and reports with new logos, changes to formulae used in common calculations, changing business vocabulary used in dashboards and reports, as well as restructuring and optimizing documents.

Rhino, the Impact Management Services scripting engine, transforms Interactive Reporting documents in custom ways. Transforming documents minimizes the impact of external change on existing dashboards and reports.

### About Impact Management Services

Impact Management Services is a collection of features that help reduce the work required to modify documents that are built and in production use. When a change happens, you identify what that means to you (one or more database tables require change, business rules or some logic needs to be modified, or user interface elements have to change).

If the change affects only one document, an Interactive Reporting Studio application is the most efficient tool to use to modify that document. If the change impacts multiple documents, or you do not know the impending impact, use Impact Management Services.

The Impact Manager module consists of two services: Impact Management Assessment Services and Impact Management Update Services. It also provides two lists: Show Task Status and Manage Tasks, and an interactive report called Show Impact of Change.

Impact Management Services are fault tolerant. For example, they detect and finish tasks that were left incomplete following an unplanned system shutdown.

Deployment of Impact Management Services is flexible. For example, the feature can be installed on computers in a Reporting and Analysis cluster, or on a computer recently added to the cluster.

Impact Management Services are scalable. For example, they can be run on one computer or on multiple computers to accommodate escalating performance requirements.

### Impact Management Assessment Services

Impact Management Assessment Services parse imported documents to extract and store metadata. Metadata includes sections that are in the document; tables and columns that are used by each data model, query, and results section; and section dependencies (for example, Results A depends on Query B depends on Data Model C).

### About Impact Management Metadata

Extracted metadata is stored in metadata tables, which share a database with the repository tables. Impact Management Assessment Services can be invoked automatically when a new document is imported, or when a new version of a document is imported. Configure Impact Management
Assessment Services by selecting Enable Harvesting in Manage General Properties of the Administer module, or configure a synchronization at a specific time. See “Synchronize Metadata Feature” on page 59.

If harvesting is enabled, Impact Management Assessment Services examine the task queue at set intervals, which by default is every 30 seconds.

Impact Management Assessment Services can also synchronize metadata tables with repository tables. The operation harvests new documents found that have not been harvested or that have changed since the documents were last harvested. Documents that have been opened, modified, and saved in Interactive Reporting Studio, are also caught by a synchronization operation. However, automatic harvesting does not catch these documents, therefore selecting Enable Harvesting in Manage General Properties of the Administer module alone does not harvest the documents.

Impact Management Update Services

Impact Management Update Services are responsible for updating imported documents according to pre-written instructions, referred to as transformations.

Update Data Models Transformation

The Update Data Models transformation is used to replace one or more data model sections with another data model. The transformation is most useful when a database changes, causing the documents that use the database to break. Transformation can also reduce the number of distinct data models in use to accommodate future upgrades.

Link Between Data Models and Queries

Data model sections are only referred to by query sections. Therefore, as long as the new data model can be attached to the query sections correctly, the rest of the document continues to work as expected.

The coupling between a query section and a data model section is through symbolic references, based on the names for the tables and columns exposed by the data model section. A small number of more complex dependencies exist regarding filters, however coupling basically relies on names. If two data models expose the names required by a query; for example, those names used in the Request and Filter lines, then either data model can support that query. If data model A exposes equivalent or more names than data model B, A is a valid replacement for B.

The concept of logical names is vital. Data model sections translate database table and column names into logical names. The generated names are the database names with spaces replacing underscores. The first letter of each word is capitalized by default, however, the logical names can be changed by users. If a table is used twice in a data model, the logical name generated for the second instance appends a number. Therefore, if the Dept table is displayed twice in a data model, the second instance is called Dept2. The names referenced in a query are always the logical names of the data model.
The Update Data Models transformation leverages symbolic coupling, by using the logical names that are independent of the database names to perform tasks.

**Update Data Models Workflow**

1. Original documents are imported.
2. Documents are harvested as part of import or through a synchronize operation.
3. Documents are used to perform daily tasks until the database requires change.
4. Use an Impact of Change report to identify the documents impacted by proposed changes.
5. Create data models to update the impacted imported documents.
6. Documents with replacement data models are harvested as part of import or through a synchronize operation.
7. From Impact Management Services, launch the Update Data Models feature.
8. Transformation parameters are specified.
   a. Select an example document with a data model impacted by the database change.
   b. Select a replacement data model.
   c. The Impact Manager module displays all Interactive Reporting documents with a data model equivalent to the selection made in step 8a. These are the candidate documents for transformation.
   d. A subset of the documents is selected and composed into a task that is queued for transformation.
9. Transformation is applied to all selected documents of the Impact Manager task.
   a. Documents are retrieved from the repository.
   b. Transformation is performed.
   c. Transformed documents are reimported as new versions of the original documents.
10. Documents are available for use against the new database definition.

**JavaScript Update Workflow**

1. Create the documents to be updated.
2. Import the documents into the repository, including the newsections.bqy and the configuration file (if these two files are later versions than the defaults provided).
3. From Impact Management Services, launch the JavaScript Update feature.
4. Select the imported documents, including the newsections.bqy and the configuration file, to be updated.
5. Update the documents. The documents are updated to create new versions of the original documents and these are saved in the original folder.
Custom Update Workflow

Custom updates enable you to customize the update process, therefore the workflow may vary depending on the selected scripts and parameters.

1. Import documents and customized scripts, if applicable.
2. From Impact Management Services, launch the Custom Update feature.
3. Specify a script. The remainder of the Custom Update is determined by the parameters required by the script.
4. Select and define parameters.
5. Monitor the execution of the scripts with associated parameters.
6. Documents are updated. A version or new document is added for each successfully updated document.

Impact Manager Module Size and Time Guidelines

The Impact Manager module can only perform data model updates if it is allowed to harvest metadata information out of the Interactive Reporting documents that are imported into the common repository. Information can be harvested as documents are imported, or harvesting can take place when the administrator requests a synch operation. Harvested metadata is placed in a set of 28 tables that start with “V8_H_”.

A synch operation causes the Synchronize Metadata Feature to look at all imported documents. If metadata for a document does not exist in the V8_H_ tables, or if the document is more recent than its metadata in the V8_H_ tables, then it needs to be harvested.

The time required to harvest an Interactive Reporting document and the amount of space required to store the information, depend on the complexity of the structures in the Interactive Reporting document. As harvesting does not look at data, a very large but simple Interactive Reporting document (say with just a query and a result section) might only require a fraction of the time and space required for a small Interactive Reporting document that contains dozens of queries, results, pivots, charts, reports, and dashboard sections.

The following guidelines are indicative of what has been observed when harvesting a variety of documents.

For a random set of around 800 documents:
- Average time to harvest a document—15 seconds
- Average space for a document—30 KB

For a complex set of documents:
- Average time to harvest a document—60 seconds
- Average space for a document—150 KB

To ensure there are no space shortage problems, it is recommended that 100K per imported document be allocated. Each version is a document, so it is important not to forget these when allocating size.
A synch of a repository with 1,000 documents is likely to take over 4 hours to complete.

**Running the Update Services**

The user interface submits tasks to the Impact Management Services. On receipt tasks are queued in a central location and then dispatched according to the priority of the running instances of the service. Undoing tasks has a higher priority than all others and they are guaranteed to be executed first. See “Using the Undo Feature” on page 110.

On task submission, the Impact Management Services acknowledge reception in a confirmation dialog displaying an allocated task number.

Dispatch and execution takes place in background and can be monitored using Show Task Status.

A task with many actions (many documents to be transformed) is always executed in parallel, with the individual actions dispatched to all available instances of the service.

Impact Management Services stores logging data for each document that is transformed. The logs are available in the server log files and in the User Interface (by double-clicking an entry in Show Task Status). You can change the level of logging to be more or less verbose. The number of messages in the logs is controlled by the Log4J configuration file (server logs) and during task submission (user logs). See Script Logging Levels.

**Note:**

For the server logs for Harvester and Transformer services, (files server_messages_HarvesterService.log and server_messages_TransformerService.log), the Log4J configuration file controls the number and formatting of messages. Whereas, for the user logs (accessed by double-clicking in Workspace), the formatting is fixed. However, the user can change the number of messages.

**Script Logging Levels**


The higher the level set, the more messages are displayed in the logs. The levels are explained in Table 3.

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug</td>
<td>Determines what is happening during script development or to track down problems</td>
</tr>
</tbody>
</table>
### Level Description

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Warn</td>
<td>Warns of recoverable problems that require correcting</td>
</tr>
<tr>
<td>Error</td>
<td>Indicates the inability to correctly perform the requested processing</td>
</tr>
<tr>
<td>Fatal</td>
<td>Indicates the script cannot continue</td>
</tr>
<tr>
<td>Always</td>
<td>Messages that are always displayed</td>
</tr>
</tbody>
</table>

There are env methods available to log messages at each of these levels. For example, `env.logInfo()`, `env.logDebug()`, and so on. See “ScriptEnvironment Object” on page 82.

There is also a default logging level associated with the script execution. The `env.log()` method logs messages at that level. The default level is initially error, but can be changed by using `env.setLogLevel()`.

The `env.logClassName()` method provides information on the type of an object, because values returned by methods are a combination of JavaScript and Java objects.

---

### Access to Impact Management Services

Only users who are assigned the Reporting and Analysis Administrator role and that hold appropriate licensing can access Impact Management Services.

After initially logging on to Workspace, select Navigate > Impact Manager, and select an option.

---

### Synchronize Metadata Feature

The Synchronize Metadata feature ensures that metadata kept for Interactive Reporting documents in the repository is up-to-date. Only documents with discrepancies are re-harvested. The synchronize action is not required if Enable Harvesting is selected in Manage General Properties of the Administer module. Select Navigate > Administer > General, and select Enable Harvesting.

The only option available for the feature is when to perform a synchronization. Request a synchronization to **run now or schedule** the operation to occur later. See “Using the Run Now Option” on page 59 and “Using the Schedule Option” on page 60.

**Note:**

Scheduling the synchronize operation with a date and time in the past, is equivalent to requesting the operation to run now.

---

### Using the Run Now Option

This option enables you to process the synchronization at the time of the request.
To run Synchronize Metadata immediately:

1 In Workspace, perform an action:
   - Import files into Workspace to harvest the metadata
   - Use Interactive Reporting Studio to change a file

2 Select Navigate > Impact Manager > Synchronize Metadata.
   Synchronize Metadata is displayed.

3 Select Run now.

4 Click Submit.
   A confirmation dialog box is displayed with a task request number.

5 Optional: To view the scheduled date and time, on the confirmation dialog box, click Details.

6 Click OK to close the dialog box.

7 Select Navigate > Impact Manager > Show Task Status.
   Ensure that synchronization is successful, by checking the status of the task request number.

8 Optional: To verify the status of the task request number, set a filter:
   a. Select Task Number, enter the task request number, and select Harvester.
   b. Click Process.
      Show Task Status displays the task.
      i. If Waiting is displayed, click Refresh.
         Updating the task from Waiting may take time.
      ii. If Success is displayed, select File > Close > Current.
      iii. If Fail is displayed, double-click the task to check the log.

Tip:

In Show Task Status, to view the most recent task, click to display the last task status page.
The latest entry is appended to the end of the list.

Using the Schedule Option

This option enables you to process the synchronization at a specified date and time in the future.

To schedule the synchronization of the metadata:

1 In Workspace, perform an action:
   - Import files into Workspace to harvest the metadata
   - Use Interactive Reporting Studio to change a file

2 Select Navigate > Impact Manager > Synchronize Metadata.
Synchronize Metadata is displayed.

3 Select Schedule.

The date and time drop-down lists are displayed.

4 Select a date.

A calendar control is displayed.

5 Select a time.

6 Click Submit.

A confirmation dialog box is displayed with a task request number.

7 Optional: To view the scheduled date and time, on the confirmation dialog box, click Details.

8 Click OK to close the dialog box.

9 To monitor the status of the task request, select Navigate > Impact Manager > Show Task Status.

10 Optional: To refine the monitoring process, set a filter:

a. Select Task Number, enter the task request number, and select Harvester.

b. Click Process, to monitor the status of the synchronize operation.

Whether the metadata is run immediately or scheduled for the future, clicking Submit causes Impact Management Assessment Services to receive the request and return a numeric request identifier. The identifier is used to filter the Impact Management Assessment Services task log. See “Using Show Task Status List” on page 107.

When Impact Management Assessment Services synchronize the metadata, only documents that have changed since the last synchronization or have no metadata are harvested.

---

**Update Data Models Feature**

The Update Data Models transformation feature enables data models in documents to be updated to reflect changes in underlying databases. You must select which data models are to be updated and supply new data models to replace the original. There are two ways in which you can update data models.

The first option, match the full data model, is restrictive and requires that the replacement data model contain all or more tables than the original data model. This is a complete data model match.

The second option, match only the query request, is more flexible. The replacement data model is only required to satisfy the queries that are associated with the original data model. For this, the replacement data model must only contain the tables and columns referenced in the query request line, query filters and sort columns. If the original data model is a master data model, the replacement must satisfy the request line, filters and sort columns of all the queries associated with the master data model. This is a less stringent requirement than matching the full data model.
Specifying Data Models

This procedure enables you to specify original and replacement document data models, interactively (using the Impact Manager module screens) or from a pre-generated list.

Note:
The documents that contain both data models must have been harvested. If a selected document has not been harvested, an error is displayed. See “Synchronize Metadata Feature” on page 59.

To specify the documents and data models:

1. **Select Navigate > Impact Manager > Update Data Models.**
   Specify Data Models is displayed.

2. **Enter a Task Description, to identify the task in Show Task Status.**
   A default description is provided.

3. **Perform an action:**
   a. Select Specify documents and data models interactively, and select an option.
      i. Select **Match the full data model.**
         - Click Browse (next to Select file containing original data model), and select an imported and synchronized Interactive Reporting document
         - Select the original data model from drop-down list (Data model sections are created when a query section is created. Because the data model section is not visible as a unique section, users may not be aware that data models are in separate sections under default names. Use Promote to Master Data Model to make a data model section visible, and enable the data model to be reused in new queries. To assist with specifying which data model is to be updated, query names are displayed after the data model in the drop-down list. See “Link Between Data Models and Queries” on page 55)
         - Click Browse (next to Select file containing replacement data model), and select an imported and synchronized Interactive Reporting document
         - Select the replacement data model from the list (For example, select Superset DataModel (Superset Query), if the selected replacement data model option requires all columns from the original data model to be present)
         - Click Next to move to Candidates
      ii. Select **Match only the query request.**
         - Click Browse (next to Select file containing replacement data model), and select an imported and synchronized Interactive Reporting document
         - Select the replacement data model from the list (For example, select Subset DataModel (Subset Query), if the selected data model includes a subset of the tables in the original data model, or select Superset DataModel (Superset Query),...
if the selected replacement data model option requires all columns from the
original data model to be present)

● Click Next to move to Candidates

b. Select Use a pre-generated list of documents and data models.
   i. Click Browse (next to Select a Task Definition File).

   The Task Definition File is a text file, that contains one line per document to
transform. Therefore, to update 10 documents, the file would contain 10 lines. Each
line contains the raw parameters required to run the script separated by commas;
that is, parameter1="value1", parameter2="value2". The values are the UUIDs and
data models for the transformation.

   ii. Click Next to move to Schedule.

Viewing Candidates to Update

This procedure displays the documents and data models that are candidates for the
transformation. This is either data models equivalent to the specified original data model (full
data model match) or data models supporting queries that are compatible with the replacement
data model (query request match).

Note:

For convenience, the Update Data Models transformation service searches for all compatible
data models. Any or all of them can be updated simultaneously.

➤ To use the candidate list to select data models for update:

1 Select a document in the list to be updated.

Other options for selecting data models for update are:

● Click Select All to update all candidates
● Use Ctrl+click or Shift+click to highlight and select individual or all documents in the list
At least one data model must be selected before clicking Next.

2 Optional: To activate the sort feature, in the candidate list table, click a column header.

For example, click Document to sort candidates by document title. The sort feature reorders the
selected candidates to be updated.

3 Click Next to move to Schedule.

4 Optional: Click Finish to complete the update.
Using Scheduling Options

This procedure enables you to select when the data model update occurs. Request the update to run now or schedule the operation to occur later. The scheduling process operates in a similar manner to synchronizing metadata.

➤ To update immediately:

1. Select Run now.
2. Select the Script logging level.
   For example, select Debug, to receive all script messages. See “Script Logging Levels” on page 58.
3. Click Finish.
   A confirmation dialog box is displayed with a task request number.
4. Optional: To view the scheduled date and time, on the confirmation dialog box, click Details.
5. Click OK to close the dialog box.
   Ensure that the update is successful, by checking the status of the task request number.
   a. Select Task Number, enter the task request number, and select Transformer.
   b. Click Process.
      Show Task Status displays the task.
      i. If Waiting is displayed, click Refresh.
         Updating the task from Waiting may take time.
      ii. If Success is displayed, select File > Close > Current.
      iii. If Fail is displayed, double-click the task to check the log.

   Tip:

   In Show Task Status, to view the most recent task, click to display the last task status page. The latest entry is appended to the end of the list.

➤ To schedule the update:

1. Select Schedule.
   The date and time drop-down lists are displayed.
2. Select a date.
   A calendar control is displayed.
3. Select a time.
4 Select the **Script logging level**.
   
   For example, select Debug, to receive all script messages. See “Script Logging Levels” on page 58.

5 **Click Finish**.
   
   A confirmation dialog box is displayed with a task request number.

6 **Optional**: To view the scheduled date and time, on the confirmation dialog box, click **Details**.

7 **Click OK** to close the dialog box.

8 **To monitor the status of the task request, select** Navigate > Impact Manager > Show Task Status.

9 **Optional**: To refine the monitoring process, set a filter:
   
   a. Select **Task Number**, enter the task request number, and select **Transformer**.
   
   b. **Click Process**, to monitor the status of the update.

**Reviewing the Confirmation Dialog Box**

The dialog box provides a numeric request identifier or a task request number that can be used to filter the Impact Management Update Services task log. See “Using Show Task Status List” on page 107.

To close the confirmation dialog box, click OK.

**Optional**: To view the scheduled date and time, on the confirmation dialog box, click **Details**.

**Data Model Updates and Jobs**

When a Data Model Update is run, a set of log entries is created and while these exist, it is possible to undo an Impact Management Services task by re-instating the prior versions. In the case where the undo operation finds a job, it marks the undo task as failed in Show Task Status (see “Using Show Task Status List” on page 107 and “Using the Undo Feature” on page 110). The Data Model Update feature operates on Interactive Reporting documents imported as jobs. However, the undo operation cannot be used on jobs, as they have no versions.

**JavaScript Update Feature**

The Impact Manager module JavaScript Update feature is equivalent to the Dashboard Development Services Update Utility on the client. JavaScript Update enables users to use the latest dashboard features without having to re-create documents from scratch.

The Impact Manager module includes a configuration file that determines the sections and controls; such as, text labels, drop-down lists and list values, to be updated, added, removed, or transferred to the target file.

An Interactive Reporting document called *newsections.bqy* is available that contains the latest version of the dashboard infrastructure. The JavaScript Update feature opens the nominated
target documents and performs a comparison of section names, by examining if sections exist in newsections.bqy and the Interactive Reporting document to be updated. If sections exist in both, the sections in the target document are removed and replaced by sections from newsections.bqy.

**Note:**

In Release 9.3 and later, the JavaScript Update feature can also update jobs.

## Using JavaScript Update

There are two options to run JavaScript Update.

1. Specify the parameters interactively.
2. Use a pre-generated definition file.

➤ To perform a JavaScript update:

1. **Select Navigate > Impact Manager > JavaScript Update.**
   Control Mode is displayed.
2. **Enter a Task Description,** to identify the task in Show Task Status.
   A default description is provided.
3. **Select an option:**
   a. **Select Specify parameters interactively,** and click **Next** to move to Parameters.
   b. **Select Use a definition file.**
      i. **Click** Browse (next to Select a Task Definition File).
      The Task Definition File is a text file, that contains one line per document to transform. Therefore, to update 10 documents, the file would contain 10 lines. Each line contains the raw parameters required to run the script separated by commas; that is, parameter1="value1", parameter2="value2".
      ii. **Click** Next to move to Schedule.

## Selecting Parameters

If Specify parameters interactively was chosen in Control Mode, Parameters enables you to select the parameters to use in JavaScript Update.

➤ To select parameters:

1. **In Parameters,** perform an action:
   a. **If the JavaScript configuration file and New sections file parameters are pre-filled:**
      i. **Click** Browse (next to Documents to update).
ii. From the multi-file picker, select the documents to update, click to move selected items, and click OK.

b. If no parameters are pre-filled or to change the parameters:
   i. Click Browse (next to JavaScript configuration file), navigate to and select JavaScriptUpdateConfig_dds.js.
      The script is located under /Administration/Impact Manager/Script Repository.
      Optional: To show hidden script files, in Select, click .
   ii. Click Browse (next to New sections file), navigate to and select newsections.bqy. The file is located under /Administration/Impact Manager/Script Repository.
   iii. Click Browse (next to Documents to update).
   iv. From the multi-file picker, select the documents to update, click to move selected items and click OK.

2 Click Next to move to Schedule.

Using Scheduling Options

This procedure enables you to select when the JavaScript update occurs. Request the update to run now or schedule the operation to occur later. The scheduling process operates in a similar manner to synchronizing metadata.

➤ To update immediately:

1 Select Run now.
2 Select the Script logging level.
   For example, select Debug, to receive all script messages. See “Script Logging Levels” on page 58.
3 Click Finish.
   A confirmation dialog box is displayed with a task request number.
4 Optional: To view the scheduled date and time, on the confirmation dialog box, click Details.
5 Click OK to close the dialog box.
6 Select Navigate > Impact Manager > Show Task Status.
   Ensure that the update is successful, by checking the status of the task request number.
   a. Select Task Number, enter the task request number, and select Transformer.
   b. Click Process.
      Show Task Status displays the task.
   i. If Waiting is displayed, click Refresh.
      Updating the task from Waiting may take time.
ii. If Success is displayed, select File > Close > Current.

iii. If Fail is displayed, double-click the task to check the log.

**Tip:**

In Show Task Status, to view the most recent task, click \( \rightarrow \) to display the last task status page. The latest entry is appended to the end of the list.

➤ To schedule the update:

1. **Select Schedule.**
   
The date and time drop-down lists are displayed.

2. **Select a date.**
   
   A calendar control is displayed.

3. **Select a time.**

4. **Select the Script logging level.**
   
   For example, select Debug, to receive all script messages. See “Script Logging Levels” on page 58.

5. **Click Finish.**
   
   A confirmation dialog box is displayed with a task request number.

6. **Optional:** To view the scheduled date and time, on the confirmation dialog box, click Details.

7. **Click OK to close the dialog box.**

8. **To monitor the status of the task request, select Navigate > Impact Manager > Show Task Status.**

9. **Optional:** To refine the monitoring process, set a filter:
   
   a. Select Task Number, enter the task request number, and select Transformer.
   
   b. Click Process, to monitor the status of the update.

### Verifying Updated JavaScript

When the status of the JavaScript update is Success, verify that the procedure is complete.

➤ **To verify the update is complete:**

1. **If a single document or a batch of documents were updated, open one document.**
   
   For example, from the Explore module, double-click an updated Interactive Reporting document.

2. **From the top panel, click \( \text{?} \).**
   
   The document release number is updated to reflect the release specified in the new sections file.
3 Close the document without saving.

Custom Update Feature

The Custom Update feature enables users to perform generic transformations depending on the update script provided. The parameters vary depending on the requirements of the script. SortDataModelTopics.js is included as a sample. It can be found in the repository under /Administration/Impact Manager/Script Repository with the two standard Impact Management Services scripts: UpdateDataModels.js and JavaScriptUpdate.js.

Note:

In Release 9.2 (Service Pack 2) and later, the Custom Update feature can also update jobs.

Using the SortDataModelTopics Script

The SortDataModelTopics script enables documents to be updated so the topics in any data models are displayed in Workspace in a user-defined order or alphabetically.

When an Interactive Reporting document is opened in Workspace and a query is visible, a list of topics is displayed in the catalog pane under Tables. The topics are displayed in the order in which they were added to the Interactive Reporting document, which makes locating topics difficult if there are many in the list.

The SortDataModelTopics script enables the user to specify the order in which the topics are displayed in these lists, using three parameters.

1. Use the first parameter to select a file containing a list of topic names, in the order preferred by the user.
2. Use the second parameter (true or false) to specify whether topics that are not included in the ordered file should be sorted alphabetically.

Topics that are not mentioned in the ordered file are placed after topics that are mentioned, and are sorted according to the second parameter. Therefore, if you provide an empty file and the second parameter is true, all topics will be ordered alphabetically, making it easy to locate a topic in the list.

Note:

The empty file should contain a blank line.

The third parameter enables selection from a set of files to be updated, through a multi-file picker.

A version is added for each successfully updated file. Therefore, double-clicking a file in Workspace displays the updated content.
Performing a Custom Update

To use the Custom Update feature to sort documents in a user-defined order or alphabetically:

1. Select Navigate > Impact Manager > Custom Update.
   
   Choose Script is displayed.

2. Click Browse (next to Choose Script), and select SortDataModelTopics.js.
   
   The script file must be selected from /Administration/Impact Manager/Script Repository/ or a sub-folder.

3. Optional: To reveal hidden files in the Script Repository, click [view hidden files].

4. Enter a Task Description, to identify the task in Show Task Status.
   
   A default description is provided.

5. Select an option:
   a. Select Specify parameters interactively, and click Next to move to Parameters.
      
      In Release 9.3.1 and later, if the selected script contains no parameters, clicking Next directs you to Schedule.
   
   b. Select Use a definition file.
      i. Click Browse (next to Select a Task Definition File).
      
      The Task Definition File is a text file. Each line must contain three parameter values, separated by commas: orderings (The UUID of the file containing the sort order listing), sortUnknownTopics (true or false), and document (The UUID of a document to be transformed).
      
      For example:
      
      orderings="/order.txt",sortUnknownTopics="true",document="/some.bqy".
      
      Users wanting to perform batch updates compile a list of documents using an Interactive Reporting document to generate the parameter file.
      
      ii. Click Next to move to Schedule.

   Note:

   In Release 9.3.1 and later, Impact Management Services scripts can be imported as specialized JS type files. If you double-click a script in the repository, the Custom Update feature is launched with the script file pre-filled.

Selecting Parameters

If Specify parameters interactively was chosen in Choose Script, Parameters enables you to select the parameters to use in the update.
To select parameters:

1. In Parameters, enter parameter values for the script.
   a. Click Browse (next to Select the documents whose topics should be re-ordered).
      For example, select the parameters file. See “Creating a Parameter Text File” on page 71.
      A multi-file picker is displayed. Select the documents to update, click ➡️ to move the selected item, and click OK.
   b. Click Browse (next to [Optional] The file containing the list of topics in the desired order).
      For example, select an Interactive Reporting document.
   c. From Sort other topic names into alphabetical order, select true or false.

2. Click Next to move to Schedule.

Creating a Parameter Text File

To create a parameter text file in Notepad:

1. In Notepad, create the file.
   For example, if you are using the SortDataModelTopics script, list the data model names in the order by which to sort. This list need only specify those names that you prefer to view first.

2. Save the text file.

3. Import the file into Workspace.

Using Scheduling Options

This procedure enables you to select when a custom update occurs. Request the update to run now or schedule the operation to occur later. The scheduling process operates in a similar manner to synchronizing metadata.

To update immediately:

1. Select Run now.

2. Select the Script logging level.
   For example, select Debug, to receive all script messages. See “Script Logging Levels” on page 58.

3. Click Finish.
   A confirmation dialog box is displayed with a task request number.

4. Optional: To view the scheduled date and time, on the confirmation dialog box, click Details.

5. Click OK to close the dialog box.
6 Select Navigate > Impact Manager > Show Task Status.

Ensure that the update is successful, by checking the status of the task request number.

a. Select Task Number, enter the task request number, and select Transformer.
b. Click Process.

Show Task Status displays the task.

i. If Waiting is displayed, click Refresh.

Updating the task from Waiting may take time.

ii. If Success is displayed, select File > Close > Current.

iii. If Fail is displayed, double-click the task to check the log.

Tip:

In Show Task Status, to view the most recent task, click to display the last task status page. The latest entry is appended to the end of the list.

➤ To schedule the update:

1 Select Schedule.

The date and time drop-down lists are displayed.

2 Select a date.

A calendar control is displayed.

3 Select a time.

4 Select the Script logging level.

For example, select Debug, to receive all script messages. See “Script Logging Levels” on page 58.

5 Click Finish.

A confirmation dialog box is displayed with a task request number.

6 Optional: To view the scheduled date and time, on the confirmation dialog box, click Details.

7 Click OK to close the dialog box.

8 To monitor the status of the task request, select Navigate > Impact Manager > Show Task Status.

9 Optional: To refine the monitoring process, set a filter:

a. Select Task Number, enter the task request number, and select Transformer.
b. Click Process, to monitor the status of the update.
Verifying the Custom Update

When the status of the update is Success, verify that the procedure is complete.

Verify the sort order has been modified, by navigating to the location of your files, and clicking . If a single document was updated, a duplicate version of the Interactive Reporting document is displayed with information appended. For example, Custom_Updated_File(topics reordered).bqy. Double-click to open the document, and select a query. From the catalog pane, expand Tables, and check the order of the topics. Close the document without saving.

If a batch of documents was updated, duplicate versions of the Interactive Reporting documents are displayed. Double-click to open a document, and select a query. From the catalog pane, view the order of the topics. Close the document without saving.

Advanced Scripting

This topic discusses customizing scripts to update documents in Workspace or on the desktop in Interactive Reporting Studio. For more details on Interactive Reporting Studio scripting, see “Updating Documents with Advanced Scripting” in the Hyperion Interactive Reporting – System 9 Object Model and Dashboard Development Services Developer’s Guide, Volume 5: Dashboard Studio.

Workspace Custom Scripting Environment

The custom scripting environment of the Impact Management Services provides a mechanism for manipulating the content and structure of an Interactive Reporting document through a Document Object Model (DOM) and JavaScript. Although this environment is similar to the scripting environment in Interactive Reporting Studio, there are differences. For example, the custom scripting environment of the Impact Management Services:

- Does not work in the context of an active Interactive Reporting document
- Provides access to all properties in the document
- Does not perform logical system-level integrity checks
- Is not contained inside the Interactive Reporting document
- Executes a script over multiple documents

The custom scripting environment performs arbitrary, common transformations on one or more documents. This mechanism is used to implement the Update Data Models and Update JavaScript features of the Impact Management Services.

Scripts can be imported into Workspace and then run using the Custom Update feature of the Impact Management Services to make changes to other imported documents. These scripts can also be run on a desktop by the Dashboard Development Services Update Utility. From the desktop, changes can be made only to files on disks visible from that desktop. The desktop is typically a development and test environment.
Scripts in Workspace run under the control of the Impact Management Services and consequently can use the Undo feature. If a change made through scripts is unwanted, the task that used the script can be undone and the documents are returned to the pre-script state.

**Calling Scripts in Workspace**

Within Workspace, the Custom Update feature of the Impact Management Services is used. The feature presents three steps to execute scripts:

1. Browse for and select a script.
2. Enter parameters required by the script. The Impact Management Services builds a parameter form that is specific to that script. Or you can specify sets of parameter values by using a batch input file.
3. **Optional:** Schedule when to execute the script.

See “Custom Update Feature” on page 69.

**Monitoring Script Execution**

Monitor script execution in the Show Task Status list. While awaiting execution, and during the running of a script, the status is displayed as Waiting (gray). Upon completion, the status changes to Success (green) or Fail (red). In Release 9.3.1, the log file of a Waiting script displays the date and time that it is scheduled to run.

When a task is complete, double-clicking the entry in the Show Task Status list displays generated log messages. Use logs to debug errant scripts. See “Using Show Task Status List” on page 107 and “Script Logging Levels” on page 58.

**Custom Scripts**

These scripts are available to update documents in Workspace.

- **JavaScriptUpdate.js**—The JavaScriptUpdate script enables users to take advantage of the latest dashboard features without having to re-create documents from scratch. See “JavaScript Update Feature” on page 65
- **UpdateDataModels.js**—The UpdateDataModels script enables data models in documents to be updated to reflect changes in underlying databases. See “Update Data Models Feature” on page 61
- **SortDataModelTopics.js**—Part of the Custom Update Feature, see “Using the SortDataModelTopics Script” on page 69

**Script Parameters**

The parameters required by a script are specified using comments in the header. These are similar in format to the JavaDoc comments used to document Java.
In Release 9.3 and earlier, these parameters are only available in Workspace.

The minimum that can be specified to define a parameter is the name; for example, @param sourceLanguage.

This assumes that the input is a simple string and displays an (initially empty) text box on the UI.

**Optional:** An @inputType line enables more specific data input methods:

- text—Text
- password—Text displayed as asterisks (*)
- file_picker_single_value—Select one file from the repository
- file_picker_multi_values—Select multiple files from the repository, all of which constitute one value
- file_picker_multi_values_parallel_execution—Select multiple files from the repository, all of which can be processed in parallel by separate instances of the script
- dropdown—Select from a predefined set of fixed values

Input types can be given a default value using @defaultValue. The @defaultValue of file_picker type is the fully qualified path and name; for example, /Administration/Impact Manager/Script Repository/SortDataModelTopics.js.

**Note:**

If this is not unique or the file does not exist, then a warning dialog is displayed and the parameter default value is not set. It has the same effect as not specifying the default value.

Drop-down lists require a separate @comboValues line that specifies possible choices, separated by commas.

**Note:**

For custom scripts, parameter values are validated only when the script is executed, not at submission time. For example, if an unacceptable value is specified for a script, the user is not informed at the time of task submission. If a script cannot recover from invalid data, it logs a message and throws an exception, causing the status to display as Fail (red) in Show Task Status, alerting the user to the problem.

---

### Document Object Model Tree Structure

The Document Object Model (DOM) is a tree structure of nodes and properties; each node is made up of more nodes and properties. The DOM and JavaScript engine provide the ability to interrogate and update the state. In Impact Management Services, it is not necessary to expand the whole Interactive Reporting document, only those nodes with properties of interest. For
example, when doing a data model update, only query and data model sections need to be expanded. However, this procedure requires no expansion, as the information is available at the top level of the DOM.

Expanding part of an Interactive Reporting document speeds up document loading, consumes less memory, and is only available in Release 9.3. In later releases it is an ignored parameter. The document loading routines expand only what is required as it is requested. Any scripts that make use of this optimization continue to work; the Document Conversion Strategy parameter is ignored.

**Note:**

In Release 9.3.1 and later, bqReadWriteDom and bqReadOnlyDom scripts can be included, however their values are ignored.

Each document manipulated by a script is stored in the form of a DOM, represented by a tree of nodes, each of which contains a set of associated properties.

The DOM for a document is acquired by retrieving the file and loading the content; for example,

```javascript
var uuid = env.getParameterValue("document");

var file = repository.retrieveFile(uuid);
var dom = env.getBqyDocument(file, bqReadWriteDom, bqDashboardReportStrategy);
```

The first line retrieves the parameter that contains the document UUID. The second line is used to copy the file from the repository to a temporary file which is deleted when the script ends. The third line loads the content of the file, providing a BqyDocument object that represents the DOM.

**Note:**

The second parameter, bqReadWriteDom, specifies that the document is to be rewritten. If it is not to be rewritten, specify bqReadOnlyDom to reduce the amount of memory required for the DOM. The third parameter is the document conversion strategy, bqDashboardReportStrategy. It determines how much of the underlying document structure is accessible to the script.

Using different strategies, the amount of memory required by a script can be reduced, as can the time spent loading the document.

**Document Conversion and Loading Strategies**

When loading documents in Release 9.3, save memory by loading only those portions of the DOM that are required by a given script; for example, JavaScript Update uses only dashboard sections. If you want to log a list of section names in a document, you do not need to load the entire tree of nodes that lie beneath the sections. An example of the required syntax:
env.getBqyDocument(documentFile, bqReadWriteDom, bqJavascriptUpdateStrategy);
env.getBqyDocument(documentFile, bqReadOnlyDom, bqDashboardReportStrategy)
env.getBqyDocument(documentFile, bqReadOnlyDom, bqDashboardReportStrategy)

These strategies are provided for loading sections:

- bqDashboardReportStrategy—Only dashboards and reports
- bqDatamodelUpgradeStrategy—All data models and queries
- bqJavascriptUpdateStrategy—Only dashboards
- bqTopLevelSectionsStrategy—All sections and section level properties (a minimal DOM is created)
- null—The whole document

Note:
In Release 9.3.1 and later, a just-in-time approach to DOM building is implemented which makes document loading strategies redundant. Any strategy parameter provided is simply ignored. Also in Release 9.3.1 and later, the bqReadWriteDom script is ignored.

**Traversing the Document Object Model**

To manipulate the content of a node in the DOM, you must locate the node.

The top-level nodes that represent the sections of a document can be accessed directly by using the Sections collection. The shapes within a dashboard are accessible through its Shapes collection. However, there is no collection for the children of a node.

Methods are provided to access the children of a node:

- getChildren()—Returns a complete list of children of a node
- getChildrenOfType()—Returns a list of children of a node that have a specific type
- addChild()—Adds a new child to the end of a list of children of a node
- removeChild()—Removes the specified node from a list of children of a node
- setChildren()—Replaces a list of children of a node with another list
- dump()—Dumps the DOM tree, starting at the given node, for debugging

To iterate over all subnodes of a given node, use getChildren() to retrieve a list that contains them. Use getChildrenOfType() to limit this to the children of a particular type. For example, the Root.MyDocument node contains a Rpt.DocComp node for each section in the document, which can be located using this line:

```javascript
var sections = root.getChildrenOfType("Rpt.DocComp");
```

A node is added as a child of another by using addChild(). Use this to copy a node from one part of the DOM (or the DOM of another document) to another location.

To remove a child node, use removeChild().
Note:

The list of children returned by getChildren() and getChildrenOfType() is read-only. If you update the list by assigning a new value to an entry, this does not affect the node. However, the current list of nodes can be replaced using setChildren().

The content of a sub-tree of the document can be written to the log using dump(). By default, this dumps the tree to standard output, but by supplying parameters, it can be written to any print stream.

XPath-Style Searching

While obtaining lists of child nodes enables access to the entire DOM, you can search for nodes that satisfy a set of criteria.

For example, this code can be used to log the names of all shapes within a document:

```javascript
for (var i = 0; i < dom.Sections.length; i++) {
  var section = dom.Sections[i];

  if (section.Type == bqDashboard) {
    env.log("Dashboard " + section.AnnotName + " has shapes");

    var shapes = section.Shapes;

    for (var j = 0; j < shapes.length; j++)
      env.log(shapes[j].Name);
  }
}
```

The DOM provides user-friendly collection names for both the Sections inside a document and the Shapes inside a dashboard. However, a complex search example that looks for all DataThreshold.DataThreshold nodes inside all ThreshFmt.ThreshFmt nodes, inside all ColColl.Item nodes, inside all table sections, results in multiple nested loops.

The Impact Management Services scripting provides an alternative approach, through XPath-style searches. For example, this is the code to use for the complex search example:

```javascript
```

This single statement provides an array that contains the required nodes. Property matching requirements can be included to narrow down which nodes are to be returned.

For example, to limit the result to those nodes in the column named Drawn inside the table named Rankings:

```javascript
```
Searches need not begin at the root of the DOM. If a variable that contains a section node is searched, use a relative path to find other nodes beneath that section; for example:

```javascript

var items = table.findNodesByPattern("ColColl.Item[Label='Drawn']" + "/ThreshFmt.ThreshFmt/DataThreshold.Threshold");
```

Use getNodesByPattern() if there is a possibility that a node may not exist (that is, if documents to be processed using a script may not contain this node) or where there can be many of these nodes. In these cases, the length of the returned array is used to determine the situation.

However, if one node matching the pattern is guaranteed, use getNodeByPattern() which returns one object, rather than an array.

The search mechanism provides two wildcard facilities. An asterisk (*) in place of a node name, represents any type of node. A pair of slashes (//) represents any number of intervening nodes.

For example, to find all images in a document, in dashboards or reports (in the body, header, footer, section header, or section footer), use this example code:

```javascript
var pattern = "//Box.Item[RuntimeClassName='PictField']";
var pictures = dom.findNodesByPattern(pattern);
```

**Differences Between the Impact Management Services and Interactive Reporting Studio Document Object Models**

Impact Management Services provides the DOM to its scripts and Interactive Reporting Studio, including Interactive Reporting Web Client, provides the BQY object model (BOM) to scripts embedded within Interactive Reporting documents.

The DOMs available in the Impact Management Services scripting differ from those provided to event handlers in Interactive Reporting Studio scripting:

- All collection indices start at zero, rather than one
- The node names and properties match those stored in the underlying document, as displayed in the Dashboard Studio Inspector Utility
- The BOM provides user-friendly names to resemble the view through the Interactive Reporting Studio; whereas, the DOM provides fewer user-friendly names
- The BOM does not provide access to the majority of properties, however the DOM provides access to all properties
- Using the DOM, the BOM event handlers for sections and shapes cannot be called to effect changes to the document
- The BOM provides safety checks and restrictions, however the DOM provides only basic type checking
- Using the DOM, you can change and transform anything; for example, you can create files that are not recognized by other software
Investigating the Impact Management Services DOM Structure


The left pane displays the nodes contained within the document as a tree. The right pane displays the names of all properties of the selected node, and their current values and associated data types.

Use the Inspector Utility when writing scripts, to determine where in the DOM the data resides that must be manipulated to achieve the intended change, and to generate the path that provides programmatic access to data of interest.

Accessing Properties

Access properties in Impact Management Services, as you do in Interactive Reporting Studio. The only difference is the DOM exported by Interactive Reporting Studio provides more user-friendly names for frequently used properties.


For example, this code accesses the IsHidden property of a dashboard, making the section visible if it is hidden.

```javascript
var dashboard = dom.Sections["Dashboard"];

if (dashboard.IsHidden) {
    env.log("Making section " + dashboard.Name + " visible");
    dashboard.IsHidden = 0;
}
```

Collections

An important difference between the Interactive Reporting Studio scripting DOM and the Impact Management Services DOM is that all collections are zero-based, not one-based. For example, a loop that would have been coded as:

```javascript
for (var i = 1; i <= collection.Count; i++)
    // some processing on collection[i]
```

is now written as:

```javascript
for (var i = 0; i < collection.length; i++)
    // some processing on collection[i]
```
**Property Types**

Every property of a DOM node has one of these data types:

- Byte
- DWord
- Long
- String
- Structure
- Word

**Accessing the File System**

To access the underlying file system from within a script; for example, where a large amount of configuration information is needed that does not change from execution to execution, use these methods.

- `env.getFileSystem()` — Retrieve an object that provides access to the underlying file system
- `env.createTempFile()` — Create a temporary file that is cleaned up when the script completes
- `fs.getFile()` — Retrieve a Java File object that refers to the file with a given path within Workspace
- `fs.writeBytesToStream()` — Write the contents of a byte array to a file

**General Java Code in Scripts**

It can be necessary to construct Java objects as part of processing a script. For example, RevertImageResources creates a FileOutputStream using the call:

```java
var fos = new Packages.java.io.FileOutputStream(imageFile);
```

The call is of the form:

```java
var object = new Packages.java.some.package.ClassName(necessary, parameters);
```

**Using Batch Input Files**

All parameters for a transformation script can be entered interactively by using the user interface, or you can request the processing of many sets of parameters by providing them as a batch input file.

Each line of a batch input file contains a complete set of parameters, as a comma-separated list of name="value" specifications.
For example, to use the SortDataModelTopics script to transform the three documents “/some.bqy”, “/some/other.bqy” and “/yet/another/example.bqy”, using the topic orderings in “/order.txt”, and sorting unspecified topic names alphabetically, use this input file:

`orderings="/order.txt",sortUnknownTopics="true",document="/some.bqy"
orderings="/order.txt",sortUnknownTopics="true",document="/some/other.bqy"
orderings="/order.txt",sortUnknownTopics="true",document="/yet/another/example.bqy"`

Note:

Each parameter value is quoted and all of them must be included on each line, even where the value does not change.

In Workspace, the values of any parameters that represent files need to be UUIDs. The sample scripts are explicitly coded to enable batch files to specify file paths, by using code similar to this to convert them into UUIDs where necessary:

```javascript
var document = env.getParameterValue("document");
if (document.substring(0, 1) == "/")
    document = repository.getFileUuid(document);
```

To enable annotation of batch input files, blank lines and any lines beginning with # are ignored.

Note:

The code also works on the desktop, because there the UUID of a file is identical to the file system path.

**Scripting References**

This topic includes scripting references and examples of objects, methods, and properties that are available to use on the desktop and in Workspace.

**ScriptEnvironment Object**

Each script has a global variable called `env`, which provides access to the ScriptEnvironment object within which it runs and hosts the features that lead to granting access to documents and the document repository. A repository can be the Reporting and Analysis repository, if the script is running in Workspace, or the file system, if the script is running on the desktop.

**expandRequestAction()**

Actions are added to the list for the task. Generally `expandRequestAction()` is used to generate multiple additional tasks to handle a collection of input files. For example, a user requests that an action occurs for a folder. The folder is expanded into an array of files and the script runs for each file in the array.
Example using expandRequestAction():

```javascript
env.expandRequestAction(strParam, arrUuidValues)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strParam</td>
<td>Represents the file descriptor that is to be expanded</td>
</tr>
<tr>
<td>arrUuidValues</td>
<td>An array of the unique identifiers of the set of files on which to act. In Workspace this is a set of real UUID values, on the desktop it is the list of paths of the files as expanded</td>
</tr>
</tbody>
</table>

**getBqyDocument()**

Used to retrieve a document from the repository, create the DOM, and provide access to the nodes and properties of the Interactive Reporting document.

Example using getBqyDocument():

```javascript
var domBqy = env.getBqyDocument(filBqy, bqOpenMode, bqStrategy)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filBqy</td>
<td>Represents the Interactive Reporting document, generally retrieved using the retrieveFile method of the repository artifact</td>
</tr>
<tr>
<td>bqOpenMode</td>
<td>Defines the way the file is opened. For example, using bqReadOnlyDom the file is read-only, or using bqReadWriteDom the file has read/write properties. For scripts running in Release 9.3.1 or later this parameter is ignored because the Rhino engine loads nodes only if scripts attempt to reference them or the properties below them (see following table).</td>
</tr>
<tr>
<td>bqStrategy</td>
<td>Determines, as an efficiency measure, how much of the DOM is built. For scripts running in Release 9.3.1 or later this parameter is ignored because the Rhino engine loads nodes only if scripts attempt to reference them or the properties below them (see following table).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bqDashboardReportStrategy</td>
<td>Loads only dashboards and reports</td>
</tr>
<tr>
<td>bqDatamodelUpgradeStrategy</td>
<td>Loads only data models and queries</td>
</tr>
<tr>
<td>bqJavaScriptUpdateStrategy</td>
<td>Loads only dashboards</td>
</tr>
<tr>
<td>bqTopLevelSectionsStrategy</td>
<td>Loads all sections and the section level properties</td>
</tr>
<tr>
<td>null</td>
<td>Loads the whole document</td>
</tr>
</tbody>
</table>

**getFileLines()**

Used to retrieve file content from the repository as an array of strings, given the UUID.

Example using getFileLines():

```javascript
var arrLines = env.getFileLines(filToRead)
```
**getParameterValue()**

Values are obtained for a single-valued parameter, based on the name. If the named parameter value does not exist, return null.

The parameter value can be entered on the command line or through the Custom Update parameter gathering screen in Workspace.

Example using getParameterValue():

```javascript
var strVal = env.getParameterValue(strName)
```

**getParameterValues()**

All values are obtained for a potentially multi-valued parameter as an array, based on the name. If the named parameter value does not exist, return null.

Example using getParameterValues():

```javascript
var arrValues = env.getParameterValues(strName)
```

**setParameterValues()**

Used to set the value of named parameter. For example, if the same values are set up over and over, a separate script can be written that lists only a subset of the parameters, which are the only ones displayed in the parameter screen, setParameterValues() is used to set the others, and loadScript() is used to read in the original.

Example using setParameterValues():

```javascript
setParameterValue(strName, strValue)
```

**getRepository()**

Used to retrieve the repository artifact in whose context the script is running. If the script is running on the desktop, this is the file system.

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>filToRead</td>
<td>The text file from the repository to expand into the constituent lines. A file object consists of information about the file, but is not the content of the file.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strName</td>
<td>The name of the parameter as supplied in the command line. For example, <code>script.js -param:document=c:\docs\myBqy -param:type=Query</code> In this case, <code>strName</code> is either document or type.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strName</td>
<td>The name of the parameter as supplied in the command line. For example, <code>script.js -param:document=&quot;[c:\docs\file1.bqy&quot;, &quot;d:\docs\file2.bqy&quot;]</code> In this case, <code>strName</code> is either document or type.</td>
</tr>
</tbody>
</table>
Example using `getRepository()`:
```javascript
var repLocal = env.getRepository();
```

**getScriptUuid()**
The repository UUID of this script is retrieved.
Example using `getScriptUuid()`:
```javascript
var uuiScript = env.getScriptUuid();
```
In Release 9.3.1 and later, use this in Workspace and on the desktop. In Release 9.3.0 and earlier, `getScriptUuid()` is only available in Workspace.

**isDesktopMode()**
Returns true if the script is running on the desktop.
Example using `isDesktopMode()`:
```javascript
var blnDesktop = env.isDesktopMode();
```

**isServerMode()**
Returns true if the script is running in Workspace.
Example using `isServerMode()`:
```javascript
var blnWorkspace = env.isServerMode();
```

**loadScript()**
The JavaScript file is loaded and merged with the main script.
Release 9.3.1 or later example using `loadScript()`:
```javascript
env.loadScript(strPath, strDesc, uuiScript);
```
Release 9.3.0 or later example using `loadScript()`:
```javascript
env.loadScript(filScript, strDesc, uuiScript);
```

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>strPath</strong></td>
</tr>
<tr>
<td><strong>filScript</strong></td>
</tr>
<tr>
<td><strong>strDesc</strong></td>
</tr>
<tr>
<td><strong>uuiScript</strong></td>
</tr>
</tbody>
</table>

**Example 1:** In Release 9.3.1 or later, use the string to search for the file in the same location as the script. If it fails to locate the file it searches the root of the script repository folder: in
Workspace the root is /Administration/Impact Manager/Script Repository, and on the desktop it is C:\Hyperion\BIPlus\DDS\scripts.

```javascript
env.loadScript("lib_hysl_core.js");
```

**Example 2:** In Release 9.3.0 or later, use the file object (`env.loadScript(filScript, strDesc, uuiScript);`) to implement a similar mechanism to Example 1.

```javascript
function _loadScript(in_strScript){
    var uuid, fil
    var folServer = "/Administration/Impact Manager/Script Repository/lib/"
    var folDesktop = "C:\Hyperion\BIPlus\DDS\scripts\"
    if (env.isServerMode()){
        uuid = cn_repLocal.getFileUuid(folServer + in_strScript)
    } else {
        uuid = folDesktop + in_strScript
    }
    fil = cn_repLocal.retrieveFile(uuid)
    env.loadScript(fil, in_strScript, uuid)
}
_loadScript("lib_hysl_core.js");
```

**writeBqyDom()**

A document is written to disk that is ready to import into the repository as a new version of an existing document or as a new document.

Example using `writeBqyDom()`:

```javascript
var filBqy = env.writeBqyDom(domBqy);
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>domBqy</td>
<td>The DOM that contains all the document nodes</td>
</tr>
</tbody>
</table>

**Reporting and Analysis Repository: Repository Artifact**

The repository artifact provides access to the features of the container that holds the documents. If the script is run in Workspace it uses the Reporting and Analysis repository. Publications that are stored here include; documents, connection information, and the folder hierarchy. If the script is run on the desktop, the file system represents the repository artifact with reduced features.

A repository artifact is created by calling `env.getRepository()` , and the object has these methods.

**addVersion Method**

A new version of the file is added to the repository. This method applies only in Workspace. If the number and names of sections are not changed by your script, then the Interactive Reporting database connection (oce) information is not required.

Example 1 using `addVersion`:
```javascript
var intV = objRep.addVersion(strUuid, objFile, strDesc);
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strUuid</td>
<td>The document UUID</td>
</tr>
<tr>
<td>objFile</td>
<td>The file that contains the new content, created by calling <code>env.writeBqyDom()</code> method</td>
</tr>
<tr>
<td>strDesc</td>
<td>The description to add to the repository</td>
</tr>
</tbody>
</table>

**Example 2 using addVersion:**

```javascript
var intV = objRep.addVersion(strUuid, objFile, strDesc, objOce);
```

Use this method format if the Interactive Reporting database connection files associated with the sections require change, or if you have modified the query or data model. This is used by Data Model Update; see “Update Data Models Feature” on page 61.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strUuid</td>
<td>The document UUID</td>
</tr>
<tr>
<td>objFile</td>
<td>The file that contains the new content, created by calling the <code>env.writeBqyDom()</code> method</td>
</tr>
<tr>
<td>strDesc</td>
<td>The description to add to the repository</td>
</tr>
<tr>
<td>objOce</td>
<td>An object that represents section information for the Interactive Reporting document, including the Interactive Reporting database connection information that is associated with each query and data model.</td>
</tr>
</tbody>
</table>

**Example 3 using addVersion:**

```javascript
var intV = objRep.addVersion(strUuid, objFile, strDesc, objOceOld, objOceNew);
```

This form of the method is tailored to a specific situation, where the updated document has the same query and data model sections as the original (one for one mapping, the same number of each as the original, with identical names to the original) and you want to retain the settings of the previous version.

The oce details are copied from `objOceOld` (the previous version in repository) to `objOceNew`, as is.

JavaScript Update uses this form of the method, because it retains the oce settings of the previous version. See “JavaScript Update Feature” on page 65.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strUuid</td>
<td>The document UUID</td>
</tr>
<tr>
<td>objFile</td>
<td>The file that contains the new content, created by calling the <code>env.writeBqyDom()</code> method</td>
</tr>
<tr>
<td>strDesc</td>
<td>The description to add to the repository</td>
</tr>
</tbody>
</table>
### convertBqyFileToUnicode Method

The Interactive Reporting document file is converted from a code page based format to Unicode format. To convert the format the desktop calls to Interactive Reporting Studio through its COM interface. This operation requires both `brioqry.exe` and `brioqry.tlb`.

Use this method if you are trying to convert an Interactive Reporting document to the latest format. For Release 9.3 and later, all images are updated to the Resource Manager format as well as converting to Unicode.

Example using `convertBqyFileToUnicode`:

```javascript
var objFile = objRep.convertBqyFileToUnicode(objFileOld, intCodePage)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objFileOld</td>
<td>The original file in the earlier format</td>
</tr>
<tr>
<td>inCodePage</td>
<td>The code page of the original document which is accessible from the DOM of the original file; that is, the attribute StdCodePage2</td>
</tr>
</tbody>
</table>

### findFiles Method

A list of the files is retrieved that are contained within the folder represented by a UUID. The UUIDs of the files are returned in an array. The call can return the files in the folder, or all the files in a hierarchy of folders under that folder.

**Example 1** using `findFiles`:

```javascript
var clcFiles = objRep.findFiles(uuiFolder, uuiMimeType, blnRecursive)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiFolder</td>
<td>The UUID of the folder. On the desktop, this is the path name.</td>
</tr>
<tr>
<td>uuiMimeType</td>
<td>The file type to search for; for example, an Interactive Reporting document</td>
</tr>
<tr>
<td>blnRecursive</td>
<td>False: examine just the folder or True: expand all sub-folders</td>
</tr>
</tbody>
</table>

**Example 2** using `findFiles`:

```javascript
var repLocal = env.getRepository()
var uuiFolder = env.getParameterValue("document")
if (repLocal.isFolder(uuiFolder)){
    var uuiMime = env.getMimeTypeUuid("application/x-brioquery")
    var clsUuuid = objRep.findFiles(uuiFolder, uuiMime, true)
    var a = 1
    for (var it = clsUuuid.iterator(); it.hasNext(); a++) {
```
folderExists Method
Returns true if the specified folder path exists within the repository.
In Release 9.3, this method is only available in Workspace and does not apply to the desktop.
Example using folderExists:
```
var blnExists = objRep.folderExists(strPath)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strPath</td>
<td>The path that represents the folder. In Workspace, a folder is represented by a UUID, and on the desktop, the UUID is the same as a path.</td>
</tr>
</tbody>
</table>

getCurrentFolder Method
Returns the string that represents the current working folder.
Example using getCurrentFolder:
```
var uuiPath = objRep.getCurrentFolder()
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiPath</td>
<td>The UUID of the folder path that is the current working folder</td>
</tr>
</tbody>
</table>

getFileUuid Method
The UUID that corresponds to the given absolute path of a file is returned. On the desktop, complete paths and the UUID are identical, but when writing scripts that are intended for the desktop or Workspace, treat UUID values and paths as if they are different and make the extra calls that convert paths to UUID values.
In Release 9.3, this method is only available in Workspace and does not apply to the desktop.
Example using getFileUuid:
```
var uuiFile = objRep.getFileUuid(strPath)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strPath</td>
<td>The complete path that represents the file</td>
</tr>
</tbody>
</table>
**getFolderContentsFor Method**

A list of the names of the files are retrieved that are contained within the folder represented by the given path. The call returns the names in just the folder, or in the entire hierarchy of folders under the folder.

In Release 9.3, this method is only available in Workspace and does not apply to the desktop.

Example using `getFolderContentsFor`:

```javascript
var arrNames = objRep.getFolderContentsFor(strPath, blnRecursive)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strPath</td>
<td>The complete path that represents the folder</td>
</tr>
<tr>
<td>blnRecursive</td>
<td>False: examine just the folder or True: expand all sub-folders</td>
</tr>
</tbody>
</table>

**getFolderUuid Method**

The UUID that corresponds to the given absolute path of a folder is returned. On the desktop, complete paths and the UUID are identical, but when writing scripts that are intended for the desktop or Workspace, treat UUID values and paths as if they are different and make the extra calls that convert paths to UUID values.

Example using `getFolderUuid`:

```javascript
var uuiFolder = repLocal.getFolderUuid(strPath)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strPath</td>
<td>The complete path that represents the file</td>
</tr>
</tbody>
</table>

**getNameForUuid Method**

The name that represents the leaf node of the path for the file referenced by the UUID is retrieved. In Release 9.3, this method is only available in Workspace and does not apply to the desktop.

Example using `getNameForUuid`:

```javascript
var strName = objRep.getNameForUuid(uuiPath)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiPath</td>
<td>The UUID of the files whose name is required</td>
</tr>
</tbody>
</table>

**getPathForUuid Method**

The path of the file referenced by the UUID is retrieved. In Release 9.3, this method is only available in Workspace and does not apply to the desktop.

Example using `getPathForUuid`:
var strPath = objRep.getPathForUuid (uuiPath)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiPath</td>
<td>The UUID of the file whose path is required</td>
</tr>
</tbody>
</table>

**getSubfolderPathsFor Method**

A list of sub-folders for a folder is retrieved.

In Release 9.3, this method is only available in Workspace and does not apply to the desktop.

Example using getSubfolderPathsFor:

```javascript
var arrPaths = objRep.getSubfolderPathsFor(uuiPath, blnRecursive)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiPath</td>
<td>The UUID of the files whose name is required</td>
</tr>
<tr>
<td>blnRecursive</td>
<td>False: examine just the folder or True: expand all sub-folders</td>
</tr>
</tbody>
</table>

**isFile Method**

Returns true if the UUID is a file.

Example using isFile:

```javascript
var bln = objRep.isFile(uuiPath)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiPath</td>
<td>The UUID of the object type that is being tested</td>
</tr>
</tbody>
</table>

**isFolder Method**

Returns true if the UUID is a folder.

Example using isFolder:

```javascript
var bln = objRep.isFolder(uuiPath)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiPath</td>
<td>The UUID of the object type that is being tested</td>
</tr>
</tbody>
</table>

**makeFolder Method**

One or more folders are created.

Optional: Creates a hierarchy of folders if they do not exist.

In Release 9.3, this method is only available in Workspace and does not apply to the desktop.
Example using makeFolder:

```
objRep.makeFolder(strPath, strDesc, blnRecursive)
```

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>strPath</td>
<td>The path that describes the folder to be created</td>
</tr>
<tr>
<td>strDesc</td>
<td>Optional: The description to be added to the repository for the path</td>
</tr>
<tr>
<td>blnRecursive</td>
<td>Optional: If this is set to true and the full parent sub-folders do not exist above the lowest node in the path, then folder creation starts at the first missing node and continues until all sub-folders in the path are created</td>
</tr>
</tbody>
</table>

**publishBqyFile Method**

An Interactive Reporting document is published or imported into the repository, configures the Interactive Reporting database connection mappings, and identifies how the Workspace server treats sections. This function performs the work also done by the publishing wizard when an Interactive Reporting document is imported into Workspace.

Example using publishBqyFile:

```
```

<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>objF</td>
<td>The Interactive Reporting document that is being published</td>
</tr>
<tr>
<td>strN</td>
<td>The file name being published</td>
</tr>
<tr>
<td>strD</td>
<td>The description associated with the file being published</td>
</tr>
<tr>
<td>uuiF</td>
<td>The folder UUID under which this is to be published</td>
</tr>
<tr>
<td>blnD</td>
<td>True indicates that the Interactive Reporting document contains dashboard sections</td>
</tr>
<tr>
<td>strH</td>
<td>The section name that is displayed when the Interactive Reporting document is activated on the thin client and when (Home) is clicked</td>
</tr>
<tr>
<td>oceP</td>
<td>An object that represents section information for the Interactive Reporting document, including the Interactive Reporting database connection information is associated with each query and data model</td>
</tr>
</tbody>
</table>

The example illustrates publishing a copy of an Interactive Reporting document. For example, if the selected file is called *sales analysis*, it is published with a name provided by the user, or if no name is provided as *Copy of sales analysis*, into the same folder as the source document. The Interactive Reporting database connection mappings from the source file are also copied to the new file so it can be processed in the same way as the source file. The script works on the desktop and in Workspace.

**Example:** Publishing a copy of an Interactive Reporting document.

```
/** *
 */
```
* @param document Select the source to copy.
* @inputType file_picker_single_value
* * @param target Provide a name to call the copied file
* */

var uuiSrc = env.getParameterValue("document");
var repLocal = env.getRepository();
var filSrc = repLocal.retrieveFile(uuiSrc);
var vrsSrc = repLocal.retrieveVersionedDocument(uuiSrc);
var strSrc = vrsSrc.getName();
var strTrg = env.getParameterValue("target");
if (strTrg == null){
    strTrg = "Copy of " + strSrc;
}
var uuiFolder = vrsSrc.getParentIdentity();
var domSrc = env.getDocument(filSrc, bqReadWriteDom, null);
var oceMapOld = vrsSrc.getSectionOCEMapping();
var oceMapNew = domSrc.sectionOCEPairInfos(uuiFolder);
for (var a = 0; a < oceMapOld.length; a++) {
    if (oceMapOld[a].isOCEEnabled()) {
        oceMapNew[a].setOCEDocument(oceMapOld[a].getOCEDocument());
        oceMapNew[a].setOCEEnabled(true);
    }
}
var strDesc = "this file was copied by a Rhino script from " + strSrc
var blnD = domSrc.isDashboard()
var strH = domSrc.getInitialTCSection()
repLocal.publish(filSrc, strTrg, strDesc, uuiFolder, blnD, strH, oceMapNew);

**retrieveFile Method**

The latest or a specific version of a file is retrieved from the repository.

Example using retrieveFile:

```javascript
var filBqy = objRep.retrieveFile(uuiBqy, intVersion)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuidBqy</td>
<td>The UUID of the specified file. On the desktop the UUID is identical to the full path of the file.</td>
</tr>
<tr>
<td>intVersion</td>
<td><strong>Optional</strong>: If omitted, then the latest version is obtained</td>
</tr>
</tbody>
</table>

**retrieveVersionedDocument Method**

The latest version of a versioned document object identified by UUID is retrieved. This method provides access to the document description, the keywords, the display name of the published file, the Interactive Reporting database connection file, and how the connection maps to the document.
Note:
retrieveVersionedDocument is not for use on the desktop.

Example using retrieveVersionedDocument:

```javascript
var vrsBqy = objRep.retrieveVersionedDocument(uuiBqy)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiBqy</td>
<td>The UUID of the specified document</td>
</tr>
</tbody>
</table>

The Node Object
An Interactive Reporting document is composed of hierarchically arranged node sets. Most nodes have commonalities and share methods and properties that apply to most Interactive Reporting document node types.

**addChild()**
An existing node is added as a child of another node.

**Tip:**
Useful to copy a node from one location to another.

Example using addChild():

```javascript
nodTrg = nodMyNode.addChild(nodSrc)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodSrc</td>
<td>References the source node to be replicated</td>
</tr>
</tbody>
</table>

**addProperty()**
An existing property is added to another node.

**Tip:**
Useful to copy a property from one node to another.

Example using addProperty():

```javascript
prpRef = nodMyNode.addProperty(prpSrc)
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prpSrc</td>
<td>References the source property to be replicated</td>
</tr>
</tbody>
</table>

**cloneNode()**

The entire node and its subordinates are cloned.

Example using cloneNode():

```javascript
var bsndNew = nodMyNode.bsndSrc.cloneNode()
```

**findNodeByPattern()**

A single node, if any, that matches the specified pattern is retrieved.

**Note:**

If a node is not found, an exception is thrown.

Example using findNodeByPattern():

```javascript
nodFound = nodMyNode.findNodeByPattern(strPattern)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strPattern</td>
<td>The search pattern</td>
</tr>
</tbody>
</table>

**findNodesByPattern()**

Nodes that match the specified pattern are retrieved.

**Note:**

No exception is thrown if none are found.

Example using findNodesByPattern():

```javascript
arrNodes = nodMyNode.findNodesByPattern(strPattern)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strPattern</td>
<td>The search pattern</td>
</tr>
</tbody>
</table>

**getChildren()**

Returns an array of nodes that are directly under this node.

Example using getChildren():
arrNodes = nodMyNode.getChildren()

**getNodeType()**

Returns the type of the node; for example, a string.

Example using getNodeType():

```
strType = nodMyNode.getNodeType()
```

**getPathWithContext()**

Returns the path, as a slash-separated list of node names, with name attribute values that remove ambiguity as to the node identity.

Example using getPathWithContext():

```
strPath = nodMyNode.getPathWithContext()
```

**getProperty()**

The object that represents a given property is retrieved.

**Tip:**

Useful for getting values of out arrays.

Example using getProperty():

```
prpResult = nodMyNode.getProperty(strName)
arrValues = prpResult.getValues()
```

<table>
<thead>
<tr>
<th>Property Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prpResult</td>
<td>An array of values that require reading or modification.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strName</td>
<td>The property name</td>
</tr>
</tbody>
</table>

getProperty() may be accompanied by a getValues() call; for example:

```
arrScripts = nodDocScripts.getProperty("EScript").getValues()
var startUp = arrScripts[0]
var shutDown = arrScripts[1]
var preProc = arrScripts[2]
var postProc = arrScripts[3]
```

In Release 9.3.1 and later, you can use the EScript property to access multi-valued properties that correspond to simple arrays; for example:

```
var someScript = docAnnotation.EScript[i]
```
In you are using a release earlier than 9.3.1, use this code:

```javascript
var eScripts = docAnnotation.EScript.getValues()
var someScript = eScripts[i]
```

**hasProperty()**

Returns true if the named property exists, otherwise returns false.

Example using hasProperty():

```javascript
blnResult = nodMyNode.hasProperty(strName)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strName</td>
<td>The property name</td>
</tr>
</tbody>
</table>

Use hasProperty() rather than performing a Boolean test on the property name, as this returns false if the property is false or zero.

```javascript
// this is not safe because if Offset is 0 it will return false
if (node.Offset){
  // do whatever is needed if the node has an Offset property
}

// this is safe
if (node.hasProperty("Offset"){
  // do whatever is needed if the node has an Offset property
}
```

**removeChild()**

The nominated child node is removed.

Example using removeChild():

```javascript
nodMyNode.removeChild(nodChild)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodChild</td>
<td>The child node to remove</td>
</tr>
</tbody>
</table>

**removeProperties()**

The properties identified by the array of names are removed.

Example using removeProperties():

```javascript
nodMyNode.removeProperties(arrNames)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrNames</td>
<td>The array of property names to delete</td>
</tr>
</tbody>
</table>
Note:
removeProperties() is useful to downgrade the format of an Interactive Reporting document to one that is generated by an earlier format or Interactive Reporting Studio Release. However, you are not required to do this because any properties or nodes not understood by Interactive Reporting Studio are ignored when the document is loaded, and therefore lost when the document is saved.

**replaceChildNode()**

The old child node is replaced with a new node.

Example using replaceChildNode():

```javascript
nodRef = nodMyNode.replaceChildNode(nodChild, nodNew)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodChild</td>
<td>A node that exists as a child</td>
</tr>
<tr>
<td>nodNew</td>
<td>The new node to replace the child</td>
</tr>
</tbody>
</table>

**The BqyDocument Object**

The BQY Document Object Model (DOM) provides access to the features of Interactive Reporting documents. Use this object to modify the internal properties of documents.

Example: Retrieving a DOM for a document.

```javascript
var uuiBqySrc = env.getParameterValue("document");
var objRep = env.getRepository();
var bqySrc = objRep.retrieveFile(uuiBqySrc);
var domSrc = env.getBqyDocument(bqySrc, bqReadWriteDom, null);
```

Note:
A DOM is a collection of BQYNode objects arranged in a hierarchy or tree structure.

**close()**

The document is closed.

Example using close():

```javascript
domSrc.close()
```

Note:
It is important to use this if a single script processes many documents as it saves resources.
**compressBqy()**

The Interactive Reporting document is compressed into the specified file.

**Note:**

Returns true if the file is compressed, or false if the file does not require compression.

Example using compressBqy():

```
var bln = domSrc.compressBqy(strNameOld, strNameNew, intInsHdrLen)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>strNameOld</td>
<td>The Interactive Reporting document name to be compressed</td>
</tr>
<tr>
<td>strNameNew</td>
<td>The name of the compressed Interactive Reporting document. If it is identical to strNameOld, then the old uncompressed file is removed</td>
</tr>
<tr>
<td>intInsHdrLen</td>
<td>The number of bytes of Interactive Reporting Web Client (Insight) Header to skip to get to the main header</td>
</tr>
</tbody>
</table>

**copy()**

A copy is created on the specified section, and the copy is added to the DOM as a section node (Rpt.DocComp).

Example using copy():

```
var rdcTarget = domSrc.copy(rdcSource)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdcSource</td>
<td>The section to be copied, the Rpt.DocComp object</td>
</tr>
</tbody>
</table>

**getInitialTCSection()**

A string is returned that identifies the Home section of an Interactive Reporting document for publishing to the thin client.

Example using getInitialTCSection():

```
strName = domSrc.getInitialTCSection()
```

**isBQYProcessable()**

Determines whether the Interactive Reporting document contains at least one section to be processed.

Example using isBQYProcessable():

```
blnResult = domSrc.isBQYProcessable()
```
**isCompressed()**
Determines whether the document, from which the DOM derives, is compressed.

**Note:**
isCompressed() is useful if a requirement for the script is to change the compression status of an Interactive Reporting document.

Example using isCompressed():
```javascript
blnResult = domSrc.isCompressed()
```

**isDashboard()**
Determines whether the Interactive Reporting document contains at least one dashboard section.

Example using isDashboard():
```javascript
blnResult = domSrc.isDashboard()
```

**sectionOCEPairInfos()**
An array of Interactive Reporting database connection mappings is provided for the DOM.

**Note:**
These arrays are not the published Interactive Reporting database connection files associated with the document. However, you can associate the array of mappings with each query published Interactive Reporting database connection, and enable the document to access a data source defined in Workspace.

Example using sectionOCEPairInfos():
```javascript
oceMap = domSrc.sectionOCEPairInfos(uuiParentFolder)
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuiParentFolder</td>
<td>The folder UUID where the document is published</td>
</tr>
</tbody>
</table>

**Example 1:** Copying the Interactive Reporting database connection mappings from one DOM to another, and republishing the DOM as a new publication or a new version.

```javascript
function copyBqy(in_repSrc, in_bqySrc, in_bqyTrg){
    var uuiFold = in_repTrg.getFolderUuid(in_bqyTrg.strFolder)
    var oceMapO = in_bqySrc.vrs.getSectionOCEMapping();
    var oceMapN = in_bqySrc.dom.sectionOCEPairInfos(uuiFold);
    for (var a = 0; a < oceMapO.length; a++) {
        if (oceMapO[a].isOCEEnabled()) {
            oceMapN[a].setOCEDocument(oceMapO[a].getOCEDocument());
            oceMapN[a].setOCEEnabled(true);
        }
    }
}
```
Example 2: Publishing a new document and assigning a specific Interactive Reporting database connection to the queries of the new document.

```javascript
var uuiFold = rep.getFolderUuid("/sales/monthly")
var oceMap = bqySrc.dom.sectionOCEPairInfos(uuiFold);
var uuiOCE = rep.getFileUuid("/OCE/salesInfo.oce")
for (var a = 0; a < oceMap.length; a++) {
    if (oceMap[a].isOCEEnabled()){
        oceMap[a].setOCEDocument(uuiOCE);
    }
}
var strD = "my description"
var blnD = bqySrc.dom.isDashboard()
var strH = bqySrc.dom.getInitialTCSection()
var filBqy = bqySrc.file
var strN = bqySrc.strName
in_repSrc.publishBqy(filBqy, strN, strD, uuiFold, blnD, strH, oceMap);
```

**Method and Properties References**

This topic includes reference tables for methods and properties.

**Reference for env Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createTempFile()</td>
<td>Create a temporary file that is cleaned up when the script completes</td>
</tr>
<tr>
<td>expandRequestAction()</td>
<td>Add a new sub-task for each set of values</td>
</tr>
<tr>
<td>getBqyDocument()</td>
<td>Construct a DOM from the content of an Interactive Reporting document</td>
</tr>
<tr>
<td>getDescription()</td>
<td>Retrieve the description associated with the script</td>
</tr>
<tr>
<td>getFileLines()</td>
<td>Read the lines of a file and construct an array that contains one string per line</td>
</tr>
<tr>
<td>getLogLevel()</td>
<td>Retrieve the current default log level that is used when calling log()</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>getMimeTypeUuid()</td>
<td>Retrieve the UUID of the specified MIME type</td>
</tr>
<tr>
<td>getNullUuid()</td>
<td>Retrieve a null UUID constant</td>
</tr>
<tr>
<td>getParameterValue()</td>
<td>Retrieve the value of the specified script parameter</td>
</tr>
<tr>
<td>getParameterValues()</td>
<td>Retrieve all of the values assigned to a multi-value script parameter</td>
</tr>
<tr>
<td>getRepository()</td>
<td>Retrieve an object that can be used to access the content of the repository</td>
</tr>
<tr>
<td>isDesktopMode()</td>
<td>Determine whether the script is being run on the desktop</td>
</tr>
<tr>
<td>isServerMode()</td>
<td>Determine whether the script is being run in Workspace</td>
</tr>
<tr>
<td>loadScript()</td>
<td>Load the content of another script into this script environment</td>
</tr>
<tr>
<td>log()</td>
<td>Post a message at the current default logging level</td>
</tr>
<tr>
<td>logAlways()</td>
<td>Post a message that is always written to the log</td>
</tr>
<tr>
<td>logClassName()</td>
<td>Post a message that contains the specified Java class name of the object</td>
</tr>
<tr>
<td>logDebug()</td>
<td>Post a message for debugging</td>
</tr>
<tr>
<td>logError()</td>
<td>Post a message associated with a detected error condition</td>
</tr>
<tr>
<td>logFatal()</td>
<td>Post a message associated with a detected error condition</td>
</tr>
<tr>
<td>logInfo()</td>
<td>Post an informational message</td>
</tr>
<tr>
<td>logWarn()</td>
<td>Post a warning message</td>
</tr>
<tr>
<td>md5Hash()</td>
<td>Generate an MD5 hash from the specified string</td>
</tr>
<tr>
<td>setLogLevel()</td>
<td>Set the default level at which logging is to be performed</td>
</tr>
<tr>
<td>setProgress()</td>
<td>Update the progress of the script</td>
</tr>
<tr>
<td>updateDescription()</td>
<td>Set a new description for this script invocation</td>
</tr>
<tr>
<td>writeBqyDom()</td>
<td>Write the specified DOM out to a file</td>
</tr>
</tbody>
</table>

## Reference for Repository Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addVersion()</td>
<td>Add a version of a document</td>
</tr>
<tr>
<td>convertBqyFileToUnicode()</td>
<td>Convert the specified document from code page to Unicode</td>
</tr>
<tr>
<td>findFiles()</td>
<td>Find all files in a folder</td>
</tr>
<tr>
<td>getFileUuid()</td>
<td>Retrieve the UUID of the file with a specified path</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>getFolderUuid()</td>
<td>Retrieve the UUID of the folder with a specified path</td>
</tr>
<tr>
<td>isFile()</td>
<td>Determine whether the specified UUID represents a file</td>
</tr>
<tr>
<td>isFolder()</td>
<td>Determine whether the specified UUID represents a folder</td>
</tr>
<tr>
<td>publishBqyFile()</td>
<td>Import a file into the repository with the specified content</td>
</tr>
<tr>
<td>remapOCEs()</td>
<td>Remap the OCEs of the specified document to the provided set</td>
</tr>
<tr>
<td>retrieveFile()</td>
<td>Retrieve the document with the specified UUID as a temporary file</td>
</tr>
<tr>
<td>retrieveVersionedDocument()</td>
<td>Retrieve the versioned document associated with the specified UUID</td>
</tr>
</tbody>
</table>

**Workspace-Specific Repository Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>changeToFolder()</td>
<td>Change the logical position within Workspace to the specified folder path</td>
</tr>
<tr>
<td>folderExists()</td>
<td>Determine whether a folder with the specified path exists in Workspace</td>
</tr>
<tr>
<td>getCurrentFolder()</td>
<td>Retrieve the path to the current folder where this script is located in Workspace</td>
</tr>
<tr>
<td>getFolderContentsFor()</td>
<td>Retrieve the UUIDs of all files in the folder</td>
</tr>
<tr>
<td>getPathForUuid()</td>
<td>Get the path in Workspace represented by the specified UUID</td>
</tr>
<tr>
<td>getSubfolderPathsFor()</td>
<td>Retrieve the UUIDs of all subfolders of the folder</td>
</tr>
<tr>
<td>makeFolder()</td>
<td>Create a subfolder with the specified name</td>
</tr>
</tbody>
</table>

**Reference for Node Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addChild()</td>
<td>Add a child under this node</td>
</tr>
<tr>
<td>addProperty()</td>
<td>Add the specified property to this node</td>
</tr>
<tr>
<td>dump()</td>
<td>Dump the content of the node and the children of the node to standard output</td>
</tr>
<tr>
<td>findNodeByPattern()</td>
<td>Find one node that matches the specified pattern</td>
</tr>
<tr>
<td>findNodesByPattern()</td>
<td>Find all nodes that match the specified pattern</td>
</tr>
<tr>
<td>getChildren()</td>
<td>Retrieve a list of all the children of this node</td>
</tr>
<tr>
<td>getChildrenOfType()</td>
<td>Retrieve a list of all the children of this node with the specified node type</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getContextualName()</code></td>
<td>Retrieve the logical name of this node</td>
</tr>
<tr>
<td><code>getNodeType()</code></td>
<td>Retrieve the type of this node</td>
</tr>
<tr>
<td><code>getPathWithContext()</code></td>
<td>Retrieve a string that represents the location of this node in the document, including contextual information to make the path unique</td>
</tr>
<tr>
<td><code>getProperties()</code></td>
<td>Retrieve a list of properties for this node</td>
</tr>
<tr>
<td><code>getProperty()</code></td>
<td>Retrieve the property of this node with the specified name</td>
</tr>
<tr>
<td><code>getRoot()</code></td>
<td>Retrieve the root node of the DOM in which this node is stored</td>
</tr>
<tr>
<td><code>hasProperty()</code></td>
<td>Determine whether this node has a property with the specified name</td>
</tr>
<tr>
<td><code>newNode()</code></td>
<td>Construct a node</td>
</tr>
<tr>
<td><code>removeChild()</code></td>
<td>Remove the specified child node</td>
</tr>
<tr>
<td><code>removeProperties()</code></td>
<td>Remove the specified list of properties from this node</td>
</tr>
<tr>
<td><code>replaceChildNode()</code></td>
<td>Replace the specified child node with the node provided</td>
</tr>
<tr>
<td><code>setChildren()</code></td>
<td>Replace the list of children of this node with the provided list</td>
</tr>
</tbody>
</table>

### Reference for document

A document retrieved by using `env.getBqyDocument()` contains these properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DesignPassword</td>
<td>Password required to enter Design mode</td>
</tr>
<tr>
<td>DocumentPassword</td>
<td>Password required to open the document</td>
</tr>
<tr>
<td>EncryptedScripts</td>
<td>Determines whether scripts in the document are encrypted</td>
</tr>
<tr>
<td>EventScripts</td>
<td>Document-level scripts</td>
</tr>
<tr>
<td>Name</td>
<td>Document name</td>
</tr>
<tr>
<td>Path</td>
<td>Path to the document</td>
</tr>
<tr>
<td>Root_MyResources</td>
<td>Root.MyResources node (or null if the document does not include Resource Manager data)</td>
</tr>
<tr>
<td>Sections</td>
<td>All sections contained in the document</td>
</tr>
<tr>
<td>Type</td>
<td>Retrieve the runtime class name</td>
</tr>
<tr>
<td>Unicode</td>
<td>Determines whether the document string content is in Unicode or code page format</td>
</tr>
</tbody>
</table>
The same document also contains these methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy()</td>
<td>Copy the specified section to the document, rename it, if necessary, to avoid duplicates</td>
</tr>
<tr>
<td>getChartSections()</td>
<td>Retrieve a list of all chart sections</td>
</tr>
<tr>
<td>getChildrenWithRuntimeClass()</td>
<td>Retrieve all child nodes with a specified RuntimeClassName</td>
</tr>
<tr>
<td>getCodePage()</td>
<td>Retrieve the code page used by the document</td>
</tr>
<tr>
<td>getDashboardSections()</td>
<td>Retrieve a list of all the dashboard sections</td>
</tr>
<tr>
<td>getInitialTCSection()</td>
<td>Retrieve the home section identifier</td>
</tr>
<tr>
<td>getPivotSections()</td>
<td>Retrieve a list of all pivot sections</td>
</tr>
<tr>
<td>getQuerySections()</td>
<td>Retrieve a list of all query sections</td>
</tr>
<tr>
<td>getResultsSections()</td>
<td>Retrieve a list of all results sections</td>
</tr>
<tr>
<td>getSource()</td>
<td>Get the path to the Interactive Reporting document from which this document was loaded</td>
</tr>
<tr>
<td>getTableSections()</td>
<td>Retrieve a list of all table sections</td>
</tr>
<tr>
<td>isBQYPasswordProtected()</td>
<td>Determine whether the document has a password</td>
</tr>
<tr>
<td>isBQYProcessable()</td>
<td>Determine whether the document has at least one processable section</td>
</tr>
<tr>
<td>load()</td>
<td>Load a document from an Interactive Reporting document on disk</td>
</tr>
<tr>
<td>optimizeImages()</td>
<td>Optimize all of the Resource Manager images to remove duplicates</td>
</tr>
<tr>
<td>save()</td>
<td>Save the document to an Interactive Reporting document on disk</td>
</tr>
<tr>
<td>sectionOCEPairInfos()</td>
<td>Retrieve a list of all the document OCE mappings</td>
</tr>
<tr>
<td>setCodePage()</td>
<td>Set the document code page</td>
</tr>
<tr>
<td>setEndianness()</td>
<td>Set whether the document should be stored as big- or small-endian</td>
</tr>
<tr>
<td>setHeader()</td>
<td>Set the document header</td>
</tr>
<tr>
<td>setSource()</td>
<td>Set the path to the source from which this document was loaded</td>
</tr>
</tbody>
</table>

**Connecting Interactive Reports**

The Impact Management Services includes a prebuilt interactive report dashboard to report the impact of change. The dashboard queries the Reporting and Analysis platform database tables and must be configured correctly before it can report on the repository.
Step 1—Configuring Hyperion Interactive Reporting Data Access Service

Use LSC to configure Hyperion Interactive Reporting Data Access Service so the service references the database system that contains the Reporting and Analysis repository tables. The configuration must match the way that business reporting data sources are configured.

For example, if the repository is implemented using MS SQL Server, the Hyperion Interactive Reporting Data Access Service configuration is displayed in the Add Data Source to DAS screen with the parameters:

Connectivity Type: ODBC
Database Type: MS SQL Server
Select the name of the data source: metadata

Note:
The data source name is metadata, as created in the ODBC configuration, and references the database instance in MS SQL Server.

Step 2—Creating Interactive Reporting Database Connections

Use Interactive Reporting Studio to create an Interactive Reporting database connection (OCE extension) that references repository tables using a matching data source name (for example, metadata) as selected in the Hyperion Interactive Reporting Data Access Service configuration.

Step 3—Importing Interactive Reporting Database Connections into Workspace

Import the Interactive Reporting database connection created in Step 2.

To import an Interactive Reporting database connection into Workspace:

1. Log in to Workspace.
2. In the Explore module view the Root folder.
3. Optional: Select View > Show Hidden to display the Administration folder.
4. Expand Administration.
5. Expand Impact Manager.
6. Import the Interactive Reporting database connection created in Step 2—Creating Interactive Reporting Database Connections procedure.
7. Name the imported file.
   For example, name the file metadata.oce.
Specify a default user identification and name to connect reports to the repository tables. The user identification requires select access to the repository tables.

**Step 4—Associating Interactive Reporting Database Connections with Interactive Reports**

Associate the Interactive Reporting database connection imported in Step 3.

To associate the Interactive Reporting database connection:

1. From the Explore module, select Administration > Impact Manager, and select the document named Show Impact of Change.
2. Right-click and select Properties. Properties is displayed.
3. Select .
4. From Connection, for each Query/DataModel Name, select metadata.oce.
5. From Options, select Use Default User Name & Password.
6. Click OK.
7. Optional: If you are using Release 9.2 and earlier, repeat steps 1–6 for the document named Show Task Status.

The Interactive Reporting documents are ready to deliver output.

**Using Show Task Status List**

Show Task Status is an Impact Manager module option that displays the status of tasks performed by Impact Management Services, and is based on the logging tables. Logging tables provide a list of logs of Impact Management Assessment Services and Impact Management Update Services tasks that have been processed or are currently processing within Impact Management Services.

Show Task Status includes paging buttons at the bottom of the screen. >>> navigates to the last page of tasks, where Refresh displays the latest entry in the list, providing that the list is sorted by ascending time (the default). Therefore, the most recent status changes can be easily monitored.

To use Show Task Status:

1. Select Navigate > Impact Manager > Show Task Status. Show Task Status is displayed.
2. In Status of Tasks, from Time Zone, select an option:
   - Browser—Displays the user browser time zone
UTC—Displays the time zone set as Coordinated Universal Time (In Release 9.2 and earlier, the UTC Offset option was a potential issue. However, the code in Release 9.3 and later assumes that the server and browser machines are set to the correct time within their time zones. Therefore, the UTC Offset option is no longer necessary.)

3 To apply filters, perform an action:
   a. Select From Time to set a start time filter.
      Date and time drop-down lists are displayed.
      i. Select a date from the calendar control.
      ii. Select a time.
   b. Select To Time to set an end time filter.
      Date and time drop-down lists are displayed.
      i. Select a date from the calendar control.
      ii. Select a time.
   c. Select Owner, and select the requester name.
   d. Select Task Number, to filter by the task request number displayed in the confirmation dialog box.
      i. Enter the task number.
      ii. Select an option:
         ● Transformer—Update task requests
         ● Harvester—Synchronization task requests
   e. Select Status, and select a status value from the drop-down list.
      For example, select Success to display only the successful tasks.

4 Click Process.

5 Optional: After processing, click Refresh to update the filtered data in the Status of Tasks table.

6 Optional: To view the most recent task in Show Task Status, click to display the last task status page.

   The latest entry is appended to the end of the list.

7 Optional: Select a task, and double-click or right-click to display the task log.

   Tasks are displayed in a table.

Table 4  Show Task Status Column Descriptions

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Color-coded status:</td>
</tr>
<tr>
<td></td>
<td>Green—Success or Undo Success</td>
</tr>
<tr>
<td></td>
<td>Gray—Waiting</td>
</tr>
<tr>
<td></td>
<td>Red—Fail or Undo Fail</td>
</tr>
</tbody>
</table>
## Managing Tasks

The Manage Task List is an Impact Manager module option that enables you to manage task requests. The list is based on the logging tables, and includes a function to undo a request that has run.

To use the Manage Task List:

1. **Select Navigate > Impact Manager > Manage Task List.**

   Manage Task List is displayed.

2. **In Task List, from Time Zone, select an option:**
   - **Browser**—Displays the user browser time zone
   - **UTC**—Displays the time zone set as Coordinated Universal Time (In Release 9.2 and earlier, the UTC Offset option was a potential issue. However, the code in Release 9.3 and later assumes that the server and browser machines are set to the correct time within their time zones. Therefore, the UTC Offset option is no longer necessary.)

3. **To apply filters, perform an action:**
   - a. **Select From Time** to set a start time filter.
      - Date and time drop-down lists are displayed.
      - i. Select a date from the calendar control.
      - ii. Select a time.
   - b. **Select To Time** to set an end time filter.
      - Date and time drop-down lists are displayed.
      - i. Select a date from the calendar control.
ii. Select a time.

c. Select Owner, and select the requester name.

4 Click Process.

5 Optional: After processing, click Refresh to update the filtered data in the Task List table.

<p>| Table 5  Manage Task List Column Descriptions |
|-----------------|------------------|</p>
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submit Time</td>
<td>Local submit date and time for the task</td>
</tr>
<tr>
<td>Description</td>
<td>Task description</td>
</tr>
<tr>
<td>Task</td>
<td>Task reference number</td>
</tr>
<tr>
<td>Owner</td>
<td>Requester name</td>
</tr>
<tr>
<td>Success</td>
<td>Successful tasks</td>
</tr>
<tr>
<td>Failure</td>
<td>Failed tasks</td>
</tr>
<tr>
<td>Waiting</td>
<td>Pending tasks</td>
</tr>
<tr>
<td>Cancelled</td>
<td>Cancelled requests</td>
</tr>
</tbody>
</table>

Using the Undo Feature

The Undo feature is a part of Manage Task List that enables the user to retrieve a request made to run a script. As the script executes, all documents that are modified are saved as new versions and this is tracked in a task log. If a user requests to undo a task, the log is used to identify all documents changed by the task and the latest version of those documents is removed, revealing the original documents.

Undo only reverts to versions of files associated with the task being undone. If the Undo feature discovers that the latest version of a file is not the same as the version it created, then that part of the undo is skipped, and a message is displayed. The undo process continues with other documents in the log.

See “Data Model Updates and Jobs” on page 65.

To use the Undo feature, Show Actions for Task must be displayed.

➤ To Show Actions for Task:

1 In Manage Task List, from Task List, select a task.

2 Double-click to view the task actions.

Show Actions of a Task is displayed.
Using Show Actions of a Task

Show Actions of a Task provides information about the selected task from Manage Task List. The table at the top displays a summation of the task. Selecting that task and clicking Undo performs a roll back process. The other table provides options that are applicable to the task. From this table, you can undo a task request or view an action log for the task.

An example of a task to undo is to revert Interactive Reporting documents that the JavaScript Update transformation has updated. For example, JavaScript Update replaces earlier sections in a set of Interactive Reporting documents that contain JavaScript with later (new) sections. The property settings from the earlier sections are transferred to the later sections, so the new code can work with old properties. The JavaScriptUpdate script enables users to use the latest dashboard features without having to re-create documents from scratch.

The task information is presented in Show Actions of a Task. If As Specified is selected, information about the folder that contains the original Interactive Reporting documents to be updated is displayed. These documents are displayed if Folder Expansion is selected. The resulting output is displayed in Files Changed, which when selected displays the updated documents that contain the later sections. When Undo is clicked, the feature removes the updated documents to reveal the initial documents.

The table columns change depending on the selected action.

<p>| Table 6 | As Specified and Folder Expansion Columns |</p>
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Color-coded status:</td>
</tr>
<tr>
<td></td>
<td>Green—Success</td>
</tr>
<tr>
<td></td>
<td>Gray—Waiting</td>
</tr>
<tr>
<td></td>
<td>Red—Fail</td>
</tr>
<tr>
<td></td>
<td>Pink—Cancel</td>
</tr>
<tr>
<td></td>
<td>Orange—Partially Successful</td>
</tr>
<tr>
<td>Submit Time</td>
<td>Local submit date and time for the task</td>
</tr>
<tr>
<td>Description</td>
<td>Task description</td>
</tr>
<tr>
<td>Action</td>
<td>Numeric order of process</td>
</tr>
<tr>
<td>Completed Time</td>
<td>Local complete date and time for the task</td>
</tr>
<tr>
<td>Duration</td>
<td>Performance time in seconds to complete the request</td>
</tr>
<tr>
<td>Service Id</td>
<td>Processor name</td>
</tr>
</tbody>
</table>

<p>| Table 7 | Files Changed Columns |</p>
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undo Status</td>
<td>Color-coded status:</td>
</tr>
<tr>
<td>Column Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>Gray—Undo Undefined</td>
</tr>
<tr>
<td></td>
<td>Green—Undo Success</td>
</tr>
<tr>
<td></td>
<td>Red—Undo Fail</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Task description</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Numeric order of process</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Explanation of the task (For example, add version)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>File, (Version)</th>
<th>File used in the task and version number</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Undo Description</th>
<th>Explanation of the undo process (For example, Version was successfully removed)</th>
</tr>
</thead>
</table>

**Note:**

Right-click a selected task in the table to Show Log For Action. The log file is displayed.

➤ To use the Undo feature:

1. **Optional:** From Time Zone, select an option:
   - Browser—Displays the user browser time zone
   - UTC—Displays the time zone set as Coordinated Universal Time (In Release 9.2 and earlier, the UTC Offset option was a potential issue. However, the code in Release 9.3 and later assumes that the server and browser machines are set to the correct time within their time zones. Therefore, the UTC Offset option is no longer necessary.)

2. From Actions, select an option:
   - As Specified—Provides an overview of the task input (For example, displays a folder)
   - Folder Expansion—Provides a view of the contents of the As Specified input (For example, files within a folder)
   - Files Changed—Displays the output of the task request (For example, a list of the updated files resulting from a JavaScript Update)

3. Select the task, and click Undo.

   A dialog box is displayed to verify that Task number nn and associated actions are to be undone.

4. Click OK.

   A confirmation dialog box is displayed with a task request number.

5. **Optional:** To view the scheduled date and time, on the confirmation dialog box, click Details.

6. Click OK to close the dialog box.

7. Navigate to Show Task Status to monitor the undo process.

8. In Show Task Status, select Task Number.

9. Enter the task request number, and select Transformer.
10 Click Process.

Show Task Status displays the task. The task Description is Undo task nn. The task request number is appended.

Note:

If a task is selected that has undergone the undo process, a dialog box is displayed that states there is no action to undo for task number nn. The task is not submitted.

Tip:

To assist with the undo process of a task, the properties of a document contains a list of version and update information. From the Explore module, right-click a document, and select Properties. In Properties, click (Versions), and scroll through the list to view updates and the original document version. Versions can be used to check any updated document.

Using Show Impact of Change Interactive Report

Show Impact of Change is an Impact Manager module option that displays the tables, columns, and joins that are used in documents on the Query Panel. Selected values display the effects on the documents of changes to selected items. As a starting point for creating a change, the Impact of Change interactive report enables you to nominate the tables or columns that have changed, and displays the reports that use those impacted structures.

Note:

The example uses Microsoft Access database software and the sample database. The Interactive Reporting database connection must be configured, see “Connecting Interactive Reports” on page 105.

To use Show Impact of Change:

1 Select Navigate > Impact Manager > Show Impact of Change.

The Query Panel of the Show Impact of Change interactive report is displayed.

2 From the lists, select items, and click to apply the selections.

For example, if MSAccess and the sample database are used, from Rdbms Table Name, select products and regions. Selections are displayed in Currently Selected Query Limits.

3 Click to process the query.

The items selected in the Query Panel are displayed. For example, the table tabs in Document with RDBMS tables selected display products and regions.
The Show Impact of Change interactive report contains information that helps identify what to do in anticipation of a change to a schema:

- **Documents with RDBMS tables selected**—Impacted documents that use the selected tables and columns
- **RDBMS/Topic column mappings**—Interactive Reporting document topics or items mapped to RDBMS tables or columns
- **Topic/RDBMS column mappings**—Reverse map of RDBMS tables or columns to Interactive Reporting document topics or items
- **Data Models with topics in common**—Common data models where impacted tables or columns are used (For example, how many Interactive Reporting documents are updated with one replacement data model)
- **RDBMS table usage details**—Documents and sections in which tables and columns are used
- **Custom request items**—Custom SQL in request items that Update Data Models, JavaScript Update, or a Custom Update may impact
- **Custom query limits**—Custom SQL in filter items that Update Data Models, JavaScript Update, or a Custom Update may impact

## Generating Parameter Files

A parameter file for updating data models can be created by any convenient mechanism; for example, in the Show Impact of Change interactive report. To generate a parameter file, click ![download](download) from the top panel of the dashboard.

There are two scenarios to create a parameter file, dependent upon the environment in which the Show Impact of Change interactive report is running.

➤ To generate a parameter file in Workspace:

1. **In Show Impact of Change, click ![download](download) to ensure that queries are processed.**
   The results are displayed.

2. **From the top panel, click ![download](download).**
   Workspace Export is displayed.

3. **From Section Names, select IM_DMU_PARAM_FILE.**

4. **From File format, select Text (tab-delimited).**

5. **Click Export.**
   Another Web browser window is activated displaying the tab separated parameter file.

6. **Select File > Save As or File > Save Page As.**
   Save As is displayed.
7 Name the parameter file IM_DMU_PARAM_FILE.txt, and click Save.

➤ To generate a parameter file on the desktop or in a Web plug-in:

1 In Show Impact of Change, from the top panel, click  

A drop-down list is displayed.

2 Select Export as Impact Manager parameter file.

Export Section is displayed.

3 In File name, enter IM_DMU_PARAM_FILE.txt.

4 From Save as type:, select Text (tab-delimited) (*.txt), and click Save.

5 Click again, to hide the drop-down list.

The generated parameter file contains these three entries per row that you must edit, (there are six altogether). Replace the entries to suit your requirements.

- replacementDmName='$RDN$'—Replace $RDN$ with the name of the new data model section that replaces the old data model
- replacementDocUuid='$RDU$'—Replace $RDU$ with the UUID of the document that contains the new data model section
- replacementDocVersion='$RDV$'—Replace $RDV$ with the version of the document that contains the new data model section

Creating New Data Models

You must create a new data model and ensure that it exposes all internal table and column names which are exposed by the replaced data model.

Renaming Tables or Columns

To build a new data model, you re-create or synchronize the existing data model against the new database, and change the name of the tables or columns in the new data model to match those in the existing data model.

For example, a column orders.orddate is renamed orders.order_date (physical name). The original data model exposed this column as Orders.Orddate (display name). The new data model gives the column a default name of Orders.Order Date (display name). To replace the original data model with the new one, edit the properties of the column and change the display name to Orders.Orddate. An example, changing physical and display names is provided in Figure 1.
Example: Renaming Tables and Columns

Microsoft Access database software and Interactive Reporting Studio are used as examples in these procedural steps.

➤ To copy a table and make initial changes to the column names:

1 In a database; for example, MSAccess, open the sample database.

On the desktop, if the installation was not customized, the sample database is located under C: \Hyperion\BIPlus\docs\samples. In Workspace, the sample database is located in the Explore module, under Sample Content, in Interactive Reporting Documents.

2 Right-click a table, and select Copy.

For example, select PCW_CUSTOMERS.

3 Right-click again, and select Paste.

4 In Paste Table As, enter a Table Name.

For example, enter Outlets. Ensure that Structure and Data is selected.

5 Click OK.

A copy of PCW_CUSTOMERS called Outlets is created.
6 Right-click **Outlets**, and select **Design View**.

   The table is opened in design mode.

7 **Overwrite Field Name** to change the column names.

   For example, overwrite **STORE_ID** with **outlet_id**, **STORE** with **outlet**, and **STORE_TYPE** with **outlet_type**.

8 **Close the Outlets table, and click Yes to save changes.**

---

To change the physical name of a table:

1 **Open Interactive Reporting Studio**, select **Sample.oce**, and click **OK**.

2 On the **Sample.oce Host User and Host Password dialog box**, click **OK**, without entering any text.

3 From the catalog pane, expand **Tables**, and drag a topic onto the content area.

   For example, select **PCW_CUSTOMERS**.

4 **Right-click the topic header**, and select **Properties**.

   Topic Properties is displayed.

5 **Enter a new Physical Name**.

   For example, enter **outlets** to replace **PCW_CUSTOMERS**.

6 **Click OK**.

---

To synchronize the data model with the database:

1 In **Interactive Reporting Studio**, select the topic with the new physical name; for example, **Pcw Customers**, and select **DataModel > Sync with Database**.

   Data Model Synchronization is displayed. If **Show Detail Information** is selected, this dialog box provides information on changes that were made with the synchronization.

2 **Click OK**.

---

To change the display names of columns:

1 In **Interactive Reporting Studio**, using the topic in the content area, right-click a column name, and select **Properties**.

   Topic Item Properties is displayed.

2 **Change the column name**, and **click OK**.

   For example, change **Outlet Id** to **Store Id**.

3 **Repeat steps 1–2 to change the other column names**.

   For example, change **Outlet** to **Store** and **Outlet Type** to **Store Type**.

   The display names of the columns are renamed.
Optional: Alternatively, to achieve an equivalent end result of changing the display names, perform these actions:

a. Drag a topic onto the Interactive Reporting Studio content area.

b. Rename the display names of the renamed columns and the topic.

For example, a data model is created that can replace another data model that uses only the Pcw Customers topic. The edited topic now exposes names matching the original topic and is a valid replacement.

Using Normalized and Denormalized Data Models

If a data model requires change because tables are being consolidated or divided, the creation of the new data model involves additional steps. To create a data model that is a superset of the original table structure use metatopics. You must give metatopics and the columns correct names, so the new data model is a true superset of the original data model. When names are correct, use the new data model in place of the original.

See “Metatopics” in the Hyperion Interactive Reporting Studio User’s Guide.

Deleting Columns

Deleted columns are replaced by a computed item with a constant value. For example, string columns may return n/a, and numeric columns may return 0. These replacements enable reports to continue working and display the constant value (for example, n/a) for the deleted columns.

Note:

If an entire table is deleted, it is treated as if the table has all columns deleted.

These procedures describe creating a computed item to mask the deletion of columns. Before creating the computed item, a series of processes; such as, copying tables, changing names, and synchronizing data models, must be performed.

➤ To copy a table and make initial changes to the column names:

1 In a database; for example, MSAccess, open the sample database.

   On the desktop, if the installation was not customized, the sample database is located under C:\Hyperion\BIPlus\docs\samples. In Workspace, the sample database is located in the Explore module, under Sample Content, in Interactive Reporting Documents.

2 Right-click a table, and select Copy.

   For example, select PCW_Items.

3 Right-click again, and select Paste.

4 In Paste Table As, enter a Table Name.

   For example, enter Goods. Ensure that Structure and Data is selected.
5 Click OK.
    A copy of PCW_Items called Goods is created.

6 Right-click Goods, and select Design View.
    The table is opened in design mode.

7 Select a row; for example, Dealer Price, and delete it.

8 Save and close the database.

➤ To change the physical name of a table:

1 Open Interactive Reporting Studio, select Sample.oce, and click OK.

2 In the Sample.oce Host User and Host Password dialog box, click OK, without entering any text.

3 From the catalog pane, expand Tables, and drag a topic onto the content area.
    For example, select and drag PCW_ITEMS.

4 Right-click the topic header, and select Properties.
    Topic Properties is displayed.

5 Enter a new Physical Name.
    For example, enter Goods as the physical name.

6 Click OK.

➤ To synchronize the data model with the database:

1 In Interactive Reporting Studio, select a topic; for example, Pcw Items, and select DataModel > Sync with Database, to perform a synchronization.
    Data Model Synchronization is displayed. If Show Detail Information is selected, the dialog box provides information on synchronization changes. For example, Dealer Price was deleted from the Goods topic.

2 Click OK.

➤ To use a computed item to mask deletion of columns:

1 In Interactive Reporting Studio, right-click a topic header; for example, Pcw Items, and select Promote to Meta Topic.
    Another topic is added to the content area. For example, the new topic is called Meta Pcw Items.

2 Right-click the original topic header; for example, Pcw Items, and select Properties.
    Topic Properties is displayed.

3 Change the topic name, and click OK.
    For example, change the name to Pcw Items topic. Two topics are now displayed. For example, the topics are Pcw Items topic and Meta Pcw Items.
4 Right-click a topic header; for example, Meta Pcw Items, and select Properties.

   Topic Properties is displayed.

5 Remove Meta from Topic Name, and click OK.

6 Select the topic from step 5; for example, Pcw Items, and select DataModel > Add Meta Topic Item > Server.

   Modify Item is displayed.

7 Enter the Name of the row that was deleted in the database, and enter a definition.

   For example, enter Dealer Price in Name, and enter 0 as the Definition.

8 Click OK.

   The computed item is added to the topic. For example, Dealer Price is added to Pcw Items.

9 Select the topic with the computed item added; for example, Pcw Items, and select DataModel > Data Model View > Meta.

   The selected topic is displayed in Meta View, and other topics are removed.

### Changing Column Data Types

Changes to a database schema may result in changes to the data types of columns. For example, strings become integers, or conversely integers become strings. When this occurs additional actions may be required to complete the migration of an Interactive Reporting document to the new schema.

If the type change affects a filter, the data type of the data model column is applied to the filter in the Interactive Reporting document. The filter type in an Interactive Reporting document is copied from the data model when it is created and cannot be accessed by developers or users.

Some data type changes require no action and are unaffected. Those changes are marked as OK in Table 8. The changes marked as Warn require attention because values cached in the Interactive Reporting document may not be migrated correctly.

<table>
<thead>
<tr>
<th>From/To</th>
<th>string</th>
<th>int</th>
<th>real</th>
<th>date</th>
<th>time</th>
<th>timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>OK</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
</tr>
<tr>
<td>int</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
</tr>
<tr>
<td>real</td>
<td>OK</td>
<td>Warn</td>
<td>OK</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
</tr>
<tr>
<td>date</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>OK</td>
<td>Warn</td>
<td>OK</td>
</tr>
<tr>
<td>time</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>OK</td>
<td>Warn</td>
</tr>
<tr>
<td>timestamp</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>Warn</td>
<td>OK</td>
</tr>
</tbody>
</table>

If the type change affects a Request line item, no action is taken because request item data types are accessed by clicking Option in Item Properties. If the Impact Manager module changes the
data types, unforeseen effects in results, tables, charts, pivots, or reports may occur—especially if computations are applied to the column that is returned.

## Compatible Replacement Data Model and Document Character Sets

When updating a data model, both Interactive Reporting documents defined in the update process must contain compatible character sets. For example, a data model in a Latin 1 document can be used to update a data model in a Latin 1 or Unicode Interactive Reporting document, but not a Latin 2 document. A data model in a Unicode document can only replace a data model in another Unicode document as Unicode documents can contain multiple character sets.

For example, in Table 9, searching for bqy_a and Dm_1 to be replaced with bqy_b and Dm-33, provides these possibilities.

### Table 9  Replacement Data Models and Interactive Reporting Documents

<table>
<thead>
<tr>
<th>Case</th>
<th>Search For</th>
<th>Replace With</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Unicode (U)</td>
<td>Unicode (U)</td>
<td>Only Unicode interactive reports are listed for selection</td>
</tr>
<tr>
<td>2</td>
<td>Unicode (U)</td>
<td>Code Page 1 (CP1)</td>
<td>Both Unicode and CP1 interactive reports are listed for selection</td>
</tr>
<tr>
<td>3</td>
<td>Code Page 1 (CP1)</td>
<td>Code Page 1 (CP1)</td>
<td>Both Unicode and CP1 interactive reports are listed for selection</td>
</tr>
<tr>
<td>4</td>
<td>Code Page 2 (CP2)</td>
<td>Code Page 1 (CP1)</td>
<td>Error Message: The documents and the replacement data model must contain compatible character sets</td>
</tr>
<tr>
<td>5</td>
<td>Code Page 1 (CP1)</td>
<td>Code Page 2 (CP2)</td>
<td>Error Message: The documents and the replacement data model must contain compatible character sets</td>
</tr>
<tr>
<td>6</td>
<td>Code Page 1 (CP1)</td>
<td>Unicode (U)</td>
<td>Error Message: The documents and the replacement data model must contain compatible character sets</td>
</tr>
</tbody>
</table>

## Changing User Names and Passwords for Interactive Reporting Documents

An Interactive Reporting document can be imported to obtain credentials for the queries used to connect to the data source in a variety of ways:

- Credentials are specified for the Interactive Reporting document
- Credentials are obtained from the Interactive Reporting database connection
- A prompt for credentials is displayed

No action is required where credentials from queries are obtained from the Interactive Reporting database connection or where a prompt occurs for the credentials. The queries that are replaced continue to prompt or to reference the Interactive Reporting database connection for the credentials.
Explicitly configured credentials may require changes as these credentials may stop working against the new data source. By changing the way the queries are imported in the replacement Interactive Reporting document, you can alter how credentials are handled in the updated Interactive Reporting document. Table 10 illustrates what happens to an Interactive Reporting document that was originally imported to connect to a data source with some explicit credentials; for example, user name=scott and password=tiger.

**Table 10  Interactive Reporting Document Before And After Update**

<table>
<thead>
<tr>
<th>Imported Replacement Interactive Reporting Document</th>
<th>Interactive Reporting Document After Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit Credentials</td>
<td>Connects the query to the data source using new credentials, user name=sa and password=secret, and processes without asking the user for values and without regard to the contents of the Interactive Reporting database connection.</td>
</tr>
<tr>
<td>Prompt User</td>
<td>Displays a log on dialog box and the user supplies a user name and password to connect.</td>
</tr>
<tr>
<td>Use Interactive Reporting database connection Default</td>
<td>Connects the query to the data source using the definition in the Interactive Reporting database connection at the time the connection is attempted.</td>
</tr>
</tbody>
</table>

### Service Configuration Parameters

Oracle's Hyperion® Impact Management Services contain several service configuration parameters that can be changed by running the Local Service Configurator (LSC). See “Assessment and Update Services Properties” in Chapter 8, “Configuring LSC Services.”

Two properties visibly affect the behavior of the Impact Manager module.

**Request Queue Polling Interval**

Determines the frequency at which the service scans the requests queue. The user experiences this interval as the greatest possible delay between initiating an action and the service starting the action.

**Clear Log Entries After**

Can affect the Task Status report of the Impact Manager module functions. Actions older than the value of the parameter may be purged from the repository. The Undo feature can only operate using information found in the log. Actions with purged log entries cannot be undone. See “Using the Undo Feature” on page 110.
Managing Calendars

You can create, modify, and delete custom calendars using Calendar Manager. You can create calendars to schedule jobs based on fiscal or other internal or organizational calendars. Jobs scheduled with custom calendars resolve dates and variable date limits against quarterly and monthly dates specified in the custom calendars, rather than the default calendar.

Topics that provide information on Calendar Manager:
- “Viewing Calendar Manager” on page 123
- “Creating Calendars” on page 124
- “Deleting Calendars” on page 124
- “Modifying Calendars” on page 124
- “Calendar Manager Properties” on page 125
- “Viewing the Job Log” on page 125

Viewing Calendar Manager

You launch Calendar Manager by clicking from the Job Utilities toolbar. At the Job Utilities Logon screen, supply a user name, password, Reporting and Analysis host, and port number.

After the system confirms these values, Calendar Manager is displayed. It lists the default calendar and custom calendars with the years for each calendar in subfolders. You choose a calendar name or year to modify or delete it.
Selecting Calendars displays a blank Calendar Property tab. Selecting a calendar name displays the Calendar Properties tab with the selected calendar record. Selecting a year displays the calendar Periods and Years tab with the selected calendar and year.

You need the Job Manager role (see the Hyperion Security Guide) to create, modify, or delete calendars.

Creating Calendars

Calendar Manager uses the standard Gregorian calendar, which cannot be modified except for holiday designations and start week day, by default.

➤ To create a calendar:
1. In Calendar Manager, select Calendars from the left navigation pane.
2. Enter a name for the calendar.
3. Enter information as requested; click Save on each dialog box.

   You must select New Year and enter a year before you can save the calendar. For field information, see “Calendar Manager Properties” on page 125.

Deleting Calendars

➤ To delete whole calendars or individual years:
1. In Calendar Manager, navigate to a calendar or year.
2. Click \.
3. When prompted, verify deletion of the calendar or year.

   You cannot delete the calendar last year. To delete the calendar last year, you must delete the entire calendar.

Modifying Calendars

➤ To modify or add years to calendars:
1. In Calendar Manager, navigate to a calendar.
   - Select a calendar name to view calendar properties.
   - Select a year to modify periods or years and non-working days.
     - When modifying periods or years be sure the dates for weeks or periods are consecutive.
     - For field information, see “Calendar Manager Properties” on page 125.
2. Select New Year to add a year to this calendar, and modify properties.
3 Click **Save**.

---

**Calendar Manager Properties**

Topics that present properties available in Calendar Manager:

- “Calendar Properties” on page 125
- “Custom Calendar Periods and Years Properties” on page 125
- “Custom Calendar Non-Working Days Properties” on page 125

**Calendar Properties**

- **Calendar Name**—Name cannot be changed after it is saved.
- **User Defined Weeks**—Enables selection of week start day. The default week contains seven days and is not associated with other time periods. User-defined weeks can be associated with periods, quarters, or months, but cannot span multiple periods. Start and end dates cannot overlap and must be sequential.
- **Week Start**—If using user-defined weeks, select a starting day for the week.

**Custom Calendar Periods and Years Properties**

- **New Year**—Any year is valid if no other years are defined. If this is not the first year defined, the year entered must be sequential.
- **Quarter/Period/Week**—The system automatically assigns sequential numbers to quarters. All calendars contain 12 periods.
- **Start and End**—Enter initial Start and End dates. The system automatically populates the remaining periods and start and end dates, and assigns quarters logically. After the fields are populated, you can edit start and end dates, which cannot overlap and must be sequential.

**Custom Calendar Non-Working Days Properties**

- **Days of the week**—Selecting days of a week populates the calendar automatically.
  
  You can select non-working days by **day** or by **day of the week**.
- **Calendar**—The calendar reflects the day starting the week as previously selected. Clicking the arrows moves the calendar forward or back one month. You indicate working and non-working days on a day-by-day basis by selecting and deselecting days.

**Viewing the Job Log**

Calendar Manager keeps a job log in text format that contains information about schedule execution including job names, start and stop times, names of users who executed jobs, reasons why jobs were executed, whether output was viewed, and directories where output is located. Jobs that are not complete have no stop time value.
A scheduled Interactive Reporting job generates an HTML log file rather than a text file.

To view the Job Log:

1. Click **(View Job Execution Log Entries).**
   The Job Log Retrieval Criteria dialog box is displayed.
2. Optional: Specify start and end dates and user information.
   You can choose to view all log entries or only those for specific dates or users. See “Job Log Retrieval Criteria” on page 126.
3. Click OK to retrieve the log (see “Job Log Entries” on page 126).

**Job Log Retrieval Criteria**

To limit Job Log entries:

1. Select Start Date or End Date.
   A calendar is displayed from which you can select a date.
   If you omit a start date, Calendar Manager retrieves those entries with the defined end date, and vice versa.
2. Select All users or select User and enter a user name.
3. Click OK.

**Job Log Entries**

The Job Log Entries window contains information about the execution of schedules, including schedule name, job name, start time, and user name who executed the job.

Users can view only those log portions that pertain to their schedules. Administrators can view all log entries, but can limit their log view by requesting to view only those entries related to specific users.

Log entries are initially sorted in ascending order by schedule name. You can sort by columns (Schedule Name, Job Name, Start Time, User, Mark for Deletion) by selecting a column heading. To sort a column in descending order, press Shift+click on a column heading.

To change the column display order, select a column heading and drag it to the desired location.

**Job Log Entry Details**

To view Job Log entry details, select a log entry and click **Detail.**

Information displayed includes schedule name, job name, output artifact, start time, stop time, user, and times executed.
Deleting Job Log Entries

➤ To delete job log entries:

1 In Job Log Entries, select a log entry, and select Mark for Deletion.
   To select multiple log entries, use the Shift or Ctrl key.

2 Click Yes when prompted to confirm the deletion.
   Entries marked for deletion are not deleted until the next Workspace server cycle, which is a recurring event where Reporting and Analysis performs routine maintenance tasks.

Managing Time Events

Public recurring time events and externally triggered events, both of which can be viewed and accessed by users, are managed only by users with the Schedule Manager role (see the Hyperion Security Guide).

Topics that provide information on managing time events:

- “Managing Public Recurring Time Events” on page 127
- “Creating Externally Triggered Events” on page 127
- “Triggering Externally Triggered Events” on page 128

Managing Public Recurring Time Events

To create, modify, and delete public recurring time events, follow the procedures for personal recurring time events described in the Hyperion Workspace User’s Guide.

Creating Externally Triggered Events

Externally triggered events are non-time events against which jobs are scheduled. Jobs scheduled against externally triggered events run after the event is triggered.

➤ To create an externally triggered event:

1 Select Navigate > Schedule > Manage Events.

2 Select Externally Triggered Event from Add Another and click Go.

3 From Create Externally Triggered Event, perform these tasks:
   a. Enter a unique name and description for the event.
   b. Make sure Active is selected.
      If the Active option is not selected, the externally triggered event does not work.
   c. Optional: Select an Effective starting date and Schedule Time At.
      The default is the current date and time.
d. **Optional:** Select **Inactive after** and select a date and **Schedule Time At**.

4. Set access permissions (see the *Hyperion Workspace User’s Guide*) to enable roles, users, or groups to view and use the public recurring time event.

5. Click **Finish**.

**Triggering Externally Triggered Events**

Externally triggered events are non-time events that are triggered manually through the Schedule module. You can trigger an external event programmatically using the `triggerETE()` method specified for Externally Triggered Event interface of Interactive Reporting SDK (see the *Hyperion Workspace Developer’s Guide*).

➤ To trigger externally triggered events:

1. Select **Navigate > Schedule > Manage Events**.

2. Select **Modify** next to an event, and click **Trigger Now**.

   A message is displayed verifying that the event triggered.

3. Click **OK** to close the verification message.

4. Click **OK**.

**Administering Public Job Parameters**

Public job parameters are managed by users with the Schedule Manager role (see the *Hyperion Security Guide*).

To create, modify, and delete public job parameters, follow the procedures for personal job parameters described in the *Hyperion Workspace User’s Guide*, except save the parameters as public instead of personal. Apply access permissions to allow roles, groups, and users to use public job parameters.

**Managing Interactive Reporting Database Connections**

Interactive Reporting documents use Interactive Reporting database connections to connect to databases. Separate Interactive Reporting database connections can be specified for each query in Interactive Reporting documents. If no Interactive Reporting database connection is specified for a query when a document is imported, users cannot process that query unless it uses only local results. It is, therefore, important that you import and allow access to Interactive Reporting database connections by users who import Interactive Reporting documents.

To process Interactive Reporting documents in Workspace, no explicit access to the Interactive Reporting database connections is required when the SC_ENABLE flag is set to `true` (this is the default). When the SC_ENABLE flag is set to `false`, only users given explicit access by the importer to the Interactive Reporting database connection associated with the Interactive Reporting document have access. Use the ConfigFileAdmin utility to toggle the flag and to set the
Managing Pass-Through for Jobs and Interactive Reporting Documents

Pass-through enables users to log on once to Workspace and access their reports’ databases without additional authentication.

As the administrator, you can provide transparent access to databases for foreground jobs and for Interactive Reporting documents by enabling pass-through globally. When pass-through is enabled globally, item owners can enable or disable pass-through for jobs and Interactive Reporting documents.

You can configure pass-through with user logon credentials or credentials set in Preferences, or you can leave the credential choice up to item owners (see “Host Authentication Properties” on page 175).

Managing Job Queuing

Job queueing occurs when no Job Service is available to process a job. Administrators can control Job Service availability using the Job Limit and Hold properties (see “Job Service Dynamic Properties” on page 149)

Topics that explain how job queueing works in specific job types:

- “Scheduled Jobs” on page 129
- “Background Jobs” on page 130
- “Foreground Jobs” on page 130
- “Interactive Reporting Services Availability and Jobs” on page 130

Scheduled Jobs

Scheduled jobs are queued when all Job Services are processing the maximum concurrent jobs defined. The queue is maintained by Event Service. Schedules in the queue are sorted based on priority and by the order in which they are triggered.

When a schedule is ready for processing, Event Service builds the job and submits it to Service Broker. Service Broker gets a list of all Job Services that can process the job and checks availability based on the number of concurrent jobs that each Job Service is processing. This information is obtained dynamically from each Job Service.

If Service Broker cannot find a Job Service to process a job, it gives a Job Limit Reached exception, which enables queuing in Event Service. The schedule is added to the queue and job data (including job application and executable information) for selecting a Job Service is cached.
When the next schedule is ready for processing, Event Service builds the job and determines if that job type is in the queue (based on cached job data). If job type matches, the job is added to the queue. If not, the job is submitted to Service Broker for processing.

When Event Service queuing is enabled, a Job Service polling thread is initialized that checks for available Job Services. If one is available, then Job Service processes the first schedule it can, based on job data cached in Event Service. Scheduled job data is removed from cache after the schedule is submitted to Job Service. Job properties that are modified are used only if the changes were made after the schedule is activated and added to the queue.

Scheduled jobs are managed through Schedule module (see the Hyperion Workspace User’s Guide).

## Background Jobs

If a Job Service is not available to process a background job (which means job limits are reached), a command is issued to Event Service to create a schedule with a custom event that runs at that time. This command persists schedule information in the database. The schedule uses job parameters associated with the background job, and Event Service processes the job as it does other scheduled jobs.

## Foreground Jobs

If Job Service is not available to process a foreground job, an exception occurs notifying the user that Job Service is busy. The user is given the option to queue the job for processing by the next available Job Service. If the user decides to queue the job, a schedule is created with a custom event that runs at that time, and Event Service processes the job as it does other scheduled jobs. The schedule and event are deleted after the job is submitted to Job Service.

Users can select to run SQR Production Reporting foreground jobs without persisting output in the repository. After the job runs, the output is kept in a client-side temporary space, and then is removed after the user is done viewing the output.

## Interactive Reporting Services Availability and Jobs

As long as the number of job requests being concurrently processed is less than a specified maximum limit, Hyperion Interactive Reporting Service accepts additional requests to process jobs from a Job Service client. If the maximum limit is reached, Hyperion Interactive Reporting Service does not accept further requests to process jobs until at least one of the jobs in progress completes. Some of the job requests are blocked until the number of concurrent job requests being processed drops below the configured maximum limit. It is unspecified which jobs are blocked and when they are allowed to begin processing.
Running Jobs in Parallel

To enable job cycles to run in parallel, use RSC to change the value of PARALLEL_CYCLE_LIMIT, a Job Service dynamic property.

By default, PARALLEL_CYCLE_LIMIT is set to 1, and job cycles are executed sequentially, one after another.

To take advantage of the performance improvements in execution time, you must run multiple Hyperion Interactive Reporting Services such that Job Service can acquire two or more Hyperion Interactive Reporting Services as required by the PARALLEL_CYCLE_LIMIT dynamic parameter. The number of Hyperion Interactive Reporting Services for processing only Interactive Reporting jobs (type JOB_ONLY_BI_SERVICE) is specified in BIPlus/common/config/irmconfig.xml.

Set the PARALLEL_CYCLE_LIMIT value to be less than or equal to the value of MAX_NUMBER_OF_MANAGED_SERVICES for IRJob Service types; The default for IRJob service type MAX_NUMBER_OF_MANAGED_SERVICES = 1.

Set the MAX_NUMBER_OF_MANAGED_SERVICES value for IRJob Service type to be less than the value for BI Service type. The default for BI service type MAX_NUMBER_OF_MANAGED_SERVICES = 2.
Customizing the User Interface

- “Changing the Default Language Selection” on page 133
- “Replacing the Logon Panel Image” on page 133
- “Removing Access to Java Client Installers” on page 134
- “Preventing Users from Changing Passwords” on page 134
- “Redirecting URLs after Users Log Off” on page 135

Changing the Default Language Selection

Reporting and Analysis reads locale information from a user’s Web browser and selects the language to use based on browser locale settings. In certain cases, users may want to view content in a language other than that determined by the browser locale. End-users can override locale-based language selection by adding languages to their Web browser language options and moving the language they want the browser to use to the top of the list.

Reporting and Analysis automatically selects the supported language in order of priority. Reporting and Analysis attempts an exact match between language and locale code. If a close match is not found, Reporting and Analysis attempts to match based only on language code and ignores the country code.

Replacing the Logon Panel Image

➤ To replace the image that is displayed on the logon panel:

1. Create an image file named `logon_panel.gif` that has a pixel width of 298 and height of 423.
2. Rename the installed version of `logon_panel.gif` to `logon_panel.gif.original`.
If you are using a Web server, the filepath to `logon_panel.gif` is similar to this:
\BIPlus\AppServer\InstalledApps\workspace_static\themes\theme_tadpole\images_global\n
If you are not using a Web server, the filepath to `logon_panel.gif` is similar to this:
\BIPlus\AppServer\InstalledApps\WebLogic\8.1\Workspace\applications\workspace\themes\theme_tadpole\images_global\n
3 Move the `logon_panel.gif` file that you created in step 1 to one of the directories identified in step 2, depending on whether you are using a Web server.

4 Restart your application and Web servers.

**Removing Access to Java Client Installers**

Administrators can hide the Workspace Tools menu option that enables end-users to install Interactive Reporting Web Client and Oracle's Hyperion® Smart View for Office.

➤ To remove the Tools menu Install option:

1 Open the `\conf\cdsconfig.jsp` file that is located in, for example:
\BIPlus\AppServer\InstalledApps\WebLogic\8.1\Workspace\applications\workspace\n
2 Change the value of the `IncludeJavaClientItems` parameter to false; for example:

```xml
<module
    ...... 
    ...... 
    <param name="IncludeJavaClientItems" value="true" />
</module>
```

**Preventing Users from Changing Passwords**

Administrators may prevent users—including native CSS users—from being able to change their passwords by setting a flag that removes the Authentication tab in Preferences. This flag overrides the role set for password change.

➤ To prevent users from being able to change their passwords:

1 Open `cdsconfig.jsp` in `workspace\conf`.

2 Change the value of the `EnableNativeUserPasswordChange` parameter to false; for example:

```xml
<module
    id="cds.mode"
    ...... 
    ...... 
    <param name="EnableNativeUserPasswordChange" value="true" />
</module>
```
Redirecting URLs after Users Log Off

To facilitate integration with custom portals, administrators can redirect Web browsers to an arbitrary static URL after users log off Workspace.

To enable the logon module to redirect Web browsers after users log off:

1. Open `cdsconfig.jsp` in `workspace\conf`.
2. Change the value for the `postLogoffURL` parameter to the URL that you want to redirect users to after they log off; for example:

   ```xml
   <module
       id="cds.mode"
       
       
       
       postLogoffURL="http://www.google.com"
   </module>
   ```

Organizing Items and Folders

For efficient Workspace functioning, structure folders so users can access items quickly and easily. Within the folder hierarchy, balance folder size against hierarchy depth. Do not let folders contain huge numbers of items, nor let the number of levels in the folder hierarchy become excessive.

**Note:**
If you frequently import content into Workspace, run a virus scan regularly on the root folder.

A hidden folder named `System` is designed for administrator use. It is visible only to administrators, and only when hidden items are revealed. Use `System` to store files you do not want users to see, such as icon files for MIME types. You cannot rename, delete, or move the `System` folder.

To view the `System` folder, in Explore, select **View > Show Hidden**.

The `System` folder is now displayed in the folder list.

The import function enables you to import Interactive Reporting, SQR Production Reporting, and generic files to the repository directly from Workspace. To import Oracle’s Hyperion® Financial Reporting – System 9 files and Oracle’s Hyperion® Web Analysis – System 9 files into the repository, you must use Oracle’s Hyperion® Financial Reporting Studio and Oracle’s Hyperion® Web Analysis Studio.

Administrating Pushed Content

You can push content to add it to users Favorites.
For example, Chris, the marketing manager, wants everyone in marketing to access the marketing schedule document easily. Chris imports the schedule and pushes this item to the marketing group. Now members of the marketing group can view the schedule from Favorites rather than having to navigate through Explore to view the document.

For instructions on how to push items, see the Hyperion Workspace User’s Guide.

Administering Personal Pages

Administrators configure the generated Personal Page and content for users’ Personal Pages. For information about using the Personal Pages, see the Hyperion Workspace User’s Guide. For details about the configuration properties of the Personal Pages servlet, see “Personal Pages Properties” on page 182.

Tasks involved with administering personal pages:

- “Configuring the Generated Personal Page” on page 136
- “Understanding Broadcast Messages” on page 137
- “Providing Optional Personal Page Content to Users” on page 139
- “Displaying HTML Files as File Content Windows” on page 139
- “Configuring Graphics for Bookmarks” on page 139
- “Configuring Exceptions” on page 140
- “Viewing Personal Pages” on page 140
- “Publishing Personal Pages” on page 140
- “Configuring Other Personal Pages Properties” on page 140

Configuring the Generated Personal Page

When users first log on to Workspace, a default generated Personal Page is listed under Favorites, which Workspace automatically creates and saves the page as part of the user’s Personal Pages. Changes the administrator makes do not affect users who previously logged on. Therefore, the exact content of a user’s generated Personal Page depends on when that user first logs on.

After logging on initially, users modify their own Personal Pages. They can also create additional Personal Pages.

Due to access permissions, the generated page may differ between users. By carefully setting access permissions on files used for the generated page, you can arrange, for example, for users in the Sales department to see different content on the generated page than users in the Production department.

Items included on the generated Personal Page by default:

- One Broadcast Messages content window with links to all items in /Broadcast Messages
• One Broadcast Messages file content window for each displayable item in Broadcast Messages
• One content window for each of the first two preconfigured folders
• The first (as sorted) displayable HTML item in any preconfigured folder
• My Bookmarks content window
• Exceptions Dashboard content window

You can customize items included by default by setting Generated Personal Page properties in Servlet Configurator (see “Personal Pages: Generated Properties” on page 183).

➤ To configure the generated Personal Page, do any or all of these tasks:
• Set Generated Personal Page properties in Servlet Configurator.
• Populate Broadcast Messages with combinations of nondisplayable items for which links display on the generated Personal Page, and displayable HTML files or external links, whose content displays there.

All these items appear as links and constitute one content window under the Broadcast Messages heading. Some displayable items may be displayed as file content windows, depending on configuration settings in Generated Personal Page properties.

• In Broadcast Messages, create preconfigured subfolders that are displayed when users first log on. Populate these folders with displayable HTML items and nondisplayable items. Each preconfigured folder has a corresponding content window that contains links to all items in the folder. Each displayable item is displayed as a file content window.

Tip:

As with any content, only users with required access permissions can see items and folders in Broadcast Messages and other preconfigured folders. To tailor the generated page for groups, put folders and items intended for those groups in Broadcast Messages and preconfigured folders, and assign access permissions to the target groups. For example, if each group accesses different subsets of preconfigured folders, then users in each group see different content windows when they first log on.

Understanding Broadcast Messages

The Broadcast Messages folder disseminates messages to all system users, except as restricted by access permissions granted on individual content items. Put announcements and documents for wide distribution in this folder.

The terms Broadcast Messages content windows and Broadcast Messages refer only to the content of the Broadcast Messages folder itself, excluding the content of its subfolders (the preconfigured folders).

Broadcast Messages include:
• One content window that displays links to all items in Broadcast Messages
File content windows for each displayable item in Broadcast Messages

Unlike other content window types, Broadcast Messages cannot be deleted from users Personal Pages.

If users makes another page their default Personal Page, Broadcast Messages remain on the originally generated Personal Page. User can delete the generated page only if they added the Broadcast Messages folder to another Personal Page. (A user can acquire multiple pages containing the Broadcast Messages by copying pushed Personal Pages.)

**Configuring Content for Broadcast Messages**

Broadcast Messages is your vehicle for customizing what users see according to enterprise or administration needs. By including content for various groups and setting access permissions on each item or folder to ensure that only its intended group has access, you push content to users' browsers.

**Configuring Preconfigured Folders**

To configure preconfigured folders for Broadcast Messages, add them to the Broadcast Messages folder.

➢ To add folders to Broadcast Messages:

1. In Explore, select Broadcast Messages.

   **Tip:**
   To view the Broadcast Messages folder, select View > Show Hidden.

2. Select File > New Folder.

3. Enter a folder name and click OK.
   The folder you created is displayed in Broadcast Messages in Explore.

**Configuring Folder Items**

➢ To configure folder items:

1. In Explore, select a folder in Broadcast Messages.

2. Select File > Import > Item.


4. Select My Personal Page, and click Content.

5. Move the Broadcast Message subfolder from Select Content to My Personal Page Content, and click Save Settings.

6. Select Favorites > My Personal Page to view the added content.
Follow the directions for adding content to Personal Pages in the *Hyperion Workspace User’s Guide*.

**Renaming Broadcast Messages Folders**

When you rename the Broadcast Messages folder, the changed folder name is displayed in the title bar of the Broadcast Messages content window in Explore and on users’ Personal Pages. The system property *Folder containing broadcast messages* automatically reflects the changed name. After renaming Broadcast Messages or its subfolder, PersonalPage Content, you must manually change another property, Location. The Location property is found in Servlet Configurator, in the Personal Pages/Publish section (see “Personal Pages: Publish Properties” on page 183).

**Providing Optional Personal Page Content to Users**

Beyond what you configure for the generated Personal Page, you can configure optional content for users to include on their Personal Pages.

All preconfigured folders are optional content for users and are displayed on the Content page for users to add to Personal Pages. A preconfigured folder is displayed on a Personal Page as a content window when added, with links to the items it contains.

Import all content to preconfigured folders using Explore (see the *Hyperion Workspace User’s Guide*).

**Displaying HTML Files as File Content Windows**

Workspace allows users to display HTML files on their Personal Pages as file content windows. This means that, rather than having links to HTML files, the file contents are displayed on Personal Pages.

By default, the first displayable item in a preconfigured folder automatically displays as a file content window on each user’s generated Personal Page. As an administrator, you ensure that users with the required access permissions see the content HTML items by subscribing to them.

See the *Hyperion Workspace User’s Guide* for information on displaying HTML files as file content windows.

**Configuring Graphics for Bookmarks**

To provide graphics that users can use for image bookmarks, place graphic files in `/wsmedia/personalize` in the servlets deployment directory.

You can add customized icon files for users upon request. Add these image files to `/wsmedia` or folders that are within the scope of the Context root (`/Hyperion`) and give the user a URL that points to that file; for example, `/wsmedia/sqr/vcr.gif`.

---

*Administering Personal Pages 139*
Note:
Icons do not display on Personal Pages if the file names or directory contains double-byte character set (DBCS) characters.

Configuring Exceptions
To enable exceptions to be added to the Exceptions Dashboard, select the Advanced Option Allow users to add this file to the Exceptions Dashboard when importing through Explore.

For information on how users can add exception-enabled jobs or items to their Exceptions Dashboard, see the Hyperion Workspace User’s Guide.

To give jobs exceptions capability, you must design jobs (usually, SQR Production Reporting programs or Interactive Reporting jobs) to write exceptions to the output.properties file. See the Hyperion Workspace User’s Guide.

For programmers’ information about supporting exceptions in jobs, see the Hyperion Workspace User's Guide.

Viewing Personal Pages
Content that you defined in Explore is displayed in the Personal Page generated by Workspace for first-time users.

Publishing Personal Pages
You can publish Personal Pages so that users can copy them to their own Personal Pages, and you can change the default Publish properties for publishing Personal Pages (see “Personal Pages: Publish Properties” on page 183).

When Personal Pages are published, they are added to the Personal Page Content folder in Broadcast Messages (default folder location is root/Broadcast Messages/Personal Page Content).

Users with modify access to /Personal Page Content can publish Personal Pages (see the Hyperion Workspace User's Guide).

Note:
Make sure that users understand that even though two users can copy a published page, they are not guaranteed identical results. Access permissions on items included on the published page determine what users see.

Configuring Other Personal Pages Properties
Use Servlet Configurator to set Personal Page configuration properties (see “Personal Pages Properties” on page 182); for example:
- Color schemes
- Maximum number of Personal Pages
- Visibility of content window headings (colored bars that resemble title bars)
About RSC

RSC is a utility that enables you to manage remote or RSC services.

RSC configures services on all hosts of a distributed Reporting and Analysis system. RSC modifies the config.dat file that resides on the target host. You can run RSC from all server hosts in the system.

In addition to modifying services, you can use RSC for these tasks:

- Adding, deleting, and modifying hosts
- Adding, deleting, and modifying database servers
- Changing the database password used by RSC services

To remove RSC services, use the ConfigFileAdmin utility (see “Using the ConfigFileAdmin Utility” on page 161).

Starting RSC

To start RSC:

1. Start Service Configurator.
   - Windows: Select Start > Programs > Hyperion > Reporting and Analysis > Utilities and Tools > Service Configurator.
   - UNIX: Run ServiceConfigurator.sh, installed in /BIPLUS/bin.

2. From the Service Configurator toolbar, select Module > Remote Service Configurator, or click .
Logging On to RSC

➤ To log on to RSC, enter the requested information:
  ● Administrative user name
  ● Password for user name
  ● Reporting and Analysis host of the services to configure
  ● Reporting and Analysis port number for the server host; default is 6800

Note:
The port number may differ if GSM is configured during installation to run on a port other than 6800.

Using RSC

When you first log on to RSC, the services that are installed on the host that you are logged on to, and basic properties of the highlighted service, are displayed. Toolbar buttons represent functions you perform using RSC.

<table>
<thead>
<tr>
<th>RSC Toolbar Button</th>
<th>Tooltip</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Exit Remote Service Configurator" /></td>
<td>Exit Remote Service Configurator</td>
<td>Closes RSC after user confirmation</td>
</tr>
<tr>
<td><img src="image" alt="Refresh item listing" /></td>
<td>Refresh item listing</td>
<td>Updates the list of services and basic properties of the selected service</td>
</tr>
<tr>
<td><img src="image" alt="Ping service" /></td>
<td>Ping service</td>
<td>Checks whether a service is “alive”</td>
</tr>
<tr>
<td><img src="image" alt="Show defined hosts" /></td>
<td>Show defined hosts</td>
<td>Displays the Defined Hosts window, where you define, delete, or modify hosts</td>
</tr>
<tr>
<td><img src="image" alt="Show defined database servers in the system" /></td>
<td>Show defined database servers in the system</td>
<td>Displays the Defined Database Servers window, where you add, delete, and modify database servers</td>
</tr>
<tr>
<td><img src="image" alt="Delete selected item" /></td>
<td>Delete selected item</td>
<td>Deletes a service after user confirmation</td>
</tr>
<tr>
<td><img src="image" alt="Show item properties" /></td>
<td>Show item properties</td>
<td>Displays properties of a service for editing</td>
</tr>
<tr>
<td><img src="image" alt="Show Help for Remote Service Configurator" /></td>
<td>Show Help for Remote Service Configurator</td>
<td>Displays online help for RSC</td>
</tr>
</tbody>
</table>

Managing Services

With RSC, you can modify properties or delete installations of these services:
Event Service (ES)
Job Service (JF)
Name Service (NS)
Repository Service (RM)
Service Broker (SB)

Topics that provide information about working with services:

- “Adding RSC Services” on page 145
- “Deleting RSC Services” on page 145
- “Pinging RSC Services” on page 146

See also “Modifying RSC Service Properties” on page 146.

# Adding RSC Services

To add RSC services to Reporting and Analysis:

1. Run the Reporting and Analysis installation program to install the service software on a host computer.
2. Configure the service during the installation process.

For information about adding Reporting and Analysis services, see the *Hyperion Reporting and Analysis – System 9 Installation Guides*.

After the service is installed, you can reconfigure it using RSC.

*Note:*

After adding RSC services, all Service Brokers in your system are notified and begin dispatching requests to the services.

# Deleting RSC Services

To delete RSC services:

1. Select a service from the Services list and click ![Delete]

*Note:*

You cannot delete Name Service, Repository Service, or Event Service.

2. When prompted, click Yes to delete the service.

When you delete services, a warning appears letting you know that an error message is displayed during uninstallation. You can safely ignore the message.
Pinging RSC Services

Errors sometime occur because services did not start correctly or stopped working properly. An easy way to test service availability is to ping it; that is, to send a message and see if it responds.

➢ To ping a service from RSC, select the service and click .

If the service is responsive, a message is displayed; for example:

Service NS1_stardust is alive.

If the service is not responsive, a message is displayed indicating that ping could not connect to the service; for example:

A Brio.Portal error occurred in Ping:
ConnectionException: Connection refused: connect

This indicates that the service is not running. If you receive this error, refer to the service log file to investigate why the error occurred.

Modifying RSC Service Properties

When you modify service properties, the service receives notification of changes and immediately updates its configuration. Most properties are used while the service is running and take effect immediately. Examples of such properties are Max Connections and Logging Levels.

Properties that are only used at start time, however, do not take effect until the next time the service starts. Such properties include Host, Directory, Log File, and IP Port.

Not every service has all property groups. The groups of properties that all or most RSC services have, and other properties of each service, are described in these topics:

- “Common RSC Properties” on page 147
- “Job Service Properties” on page 149

Note:

RSC services not mentioned explicitly in this section have only common properties.

➢ To view or modify service properties:

1 Double-click the service name or select the service name and click .

A properties page is displayed.

2 Select a tab to view or modify a group of properties.

For example, select the Storage tab to modify properties that define where persistent storage of business data is located.
Common RSC Properties

All RSC services have general and advanced properties and most have storage properties:

- “General RSC Properties” on page 147
- “Advanced RSC Properties” on page 147
- “RSC Storage Properties” on page 148
- “Repository Service Dynamic Properties” on page 149

General RSC Properties

General properties of all RSC services:

- **Description**—Brief description of the service.
- **Host**—Host on which the service resides. You can select or define a host. If you define a host, enter a name that makes the service easily identifiable within your organization. The maximum number of characters allowed is 64. See “Managing Hosts” on page 153.
- **IP Port**—Service IP port number. The wizard assigns a unique port to each service. Even if you install multiple services of one type (Job Service, for example) on one host, the wizard automatically enters a unique IP port number for each one.
- **Directory**—Location where the service resides. Adopt a convention for naming the directories where you store service information. For example, for Event Service named ES_apollo, the directory might be `\biplus\data\ES_apollo`.

**Note:**

Changes to Host, IP Port, and Directory properties do not take effect until the service is restarted.

Advanced RSC Properties

Advanced properties describe the service’s logging level and the maximum number of connections the service supports. All services have advanced properties.

- **Log Levels**—Level at which service errors are logged. See “Configuring Logging Levels” on page 204.

  A change to this property takes effect immediately. Therefore, when errors occur and you want more debugging information, you can change the logging level without restarting the service.

- **Max Connections**—Maximum number of connections allowed. Consider memory allocation for the connections you allow. You must increase the maximum number of file descriptors on some systems, such as UNIX.

  A change to this property takes effect immediately. Changing the Max Connections setting without restarting the service is useful to dynamically tune the service at run time.
RSC Storage Properties

Storage properties are used by a service to connect to the database where it stores its data. These services store data of their own:

- **Name Service**—General configuration information, such as lists of hosts, and database servers
- **Repository Service**—Reporting and Analysis content metadata

**Caution!**

When replicating Repository Service, the file system used for storing physical files must be shared between systems for it to work. The file system is located with the first Repository Service, and remote Repository Services need read/write access to it.

- **Event Service**—Schedules and subscriptions

Service Broker and Job Service do not have storage properties.

Data for all these services is stored in the repository database, for which storage properties define connectivity:

- **DB Driver**—Name of the driver used to access the database. This is database dependent and should only be changed by an experienced administrator.
  
  If you change DB Driver, you must change other files, properties, data in the database, and the Java classpath. See “Changing the Repository Database Driver or JDBC URL” on page 158.

- **JDBC URL**—URL for Java access to the database using the JDBC driver. The services use this URL to connect to the database server.
  
  If you change JDBC URL, you must change other files, properties, and data in the database. For details, see “Changing the Repository Database Driver or JDBC URL” on page 158.

- **User Name**—User name for the database account. All services should use one database account.

- **Password**—Password for the database account.

**Caution!**

Reporting and Analysis only supports configurations in which all services connect to one database. For this reason, change the settings on this tab only if you are an experienced Reporting and Analysis administrator; otherwise, request assistance from Hyperion Customer Support.

Storage property settings rarely should be changed. The circumstances that would require changes include, for example, assignment of host names on your network, changes to a database user account (name or password), or changes to database type (as from Oracle to Sybase). Such changes require extensive changes to external systems configuration as well.
Repository Service Dynamic Properties

Dynamic properties enable Repository Service to issue warnings when disk space is running low.

- **DISK_SPACE_USAGE_LIMIT**—Determines when Repository Service should issue warnings about file system storage disk space limit (% used); default is 90
- **DISK_SPACE_NOTIF_EMAIL**—E-mail address to which notifications are sent when the specified disk space usage limit is reached; separate multiple e-mail addresses with semicolons (;)

Job Service Properties

Topics that describe properties unique to Job Service and other tasks that start from the Job Service’s Properties dialog box.

- “Job Service Dynamic Properties” on page 149
- “Job Service Database Properties” on page 150
- “Job Service SQR Production Reporting Properties” on page 150
- “Job Service Application Properties” on page 151
- “Executable Job Service Properties” on page 153

When you modify properties of Job Service, the service receives change notifications and updates its configuration immediately. Properties used while the service is running take effect immediately. Such properties include Max Connections, Logging Levels, and all properties on the Database, SQR Production Reporting, Application, and Executable tabs.

Properties only used at start time, however, do not take effect until the next time Job Service starts. Such properties include Directory, Log File, and IP Port.

Job Service Dynamic Properties

Dynamic properties provide information about how Job Service processes jobs:

- **Acquire Only Job-Only BI Services**—Determines whether this Job Service runs as a dedicated job processor or as an interactive processor that processes both Interactive Reporting jobs and interactive reports
- **Support BQY**—Determines whether Job Services can process Interactive Reporting jobs
  
  If set to true, then Job Service can run Interactive Reporting jobs; otherwise, it cannot. If Job Service is installed with Interactive Reporting services, then this property is set to true by default.
- **Job Limit**—Maximum number of concurrent jobs to be run by Job Service
  
  If this value is 0 or -1, an unlimited number of concurrent jobs can be run. Job Limit cannot be modified at runtime. Changes made to Job Limit are picked up by Job Service dynamically.
- **Hold**—Determines whether Job Service can accept jobs for processing
When set to true, Job Service continues to process jobs that are running, but does not process any new jobs.

- **Parallel Cycle Limit**—Determines the number of job cycles that can be executed simultaneously using different Interactive Reporting services if the number of Interactive Reporting services is greater than one; default is 1

All properties can be changed without restarting Job Service.

**Job Service Database Properties**

Database properties provide information needed for Job Service to connect to the databases against which it runs jobs, including Server Name, Type, and Location of the host where the database server resides.

Adding connectivity to local database servers enables Job Service to run programs that connect directly to local databases. (See “Adding Database Servers” on page 155.)

To define connectivity between Job Service and an additional database:

1. **Click Add.**
   A list of Reporting and Analysis database servers is displayed.
2. **Select a database and define connection strings or environment variables.**

To delete a database’s connectivity from Job Service, click **Delete.**

To modify the connectivity properties of a database:

1. **Select a database from the list and click Modify.**
2. **Modify or create environment variables using Name and Value.**
   For example, `name=ORACLE_SID, value=PAYROLL`.

**Note:**

The Database Servers list combined with the SQR Production Reporting servers list is used to construct data sources for importing SQR Production Reporting documents.

**Job Service SQR Production Reporting Properties**

The SQR Production Reporting page lists SQR Production Reporting servers that are currently defined and available on the same host as Job Service. SQR Production Reporting properties define the SQR Production Reporting servers in the system that are used to run SQR Production Reporting jobs.

You can add and delete SQR Production Reporting servers, and modify the path of SQR Production Reporting servers by clicking the corresponding button.
Job Service Application Properties

Job Service properties describe the applications used by Job Service. You can configure Job Service to run combinations of three job types: SQR Production Reporting, Interactive Reporting, and generic.

- **Application**—Name of the application. Select an application or add one. All applications defined in Reporting and Analysis are listed. Applications can have multiple executables, each on a different Job Service to distribute the load.
- **Description**—Optional read-only description of the application. Click Modify to change the description.
- **Command String**—Read-only command string to pass to the application when it runs. Click Modify to change the command string.

You can add applications to Job Service, delete applications that have no associated executables, and modify application properties by clicking the corresponding button. The Add button is available only when you must define executables for applications (see “Adding Applications for Job Service” on page 151).

After you add applications, you must define their executable properties (see “Executable Job Service Properties” on page 153).

Adding Applications for Job Service

When adding applications, you must specify the application and an executable. An application may be installed on multiple hosts. Each installation of the application has a different executable, or program file and path, which you define on the Executable page.

For example, Oracle Reports (an application) might be installed on two hosts, apollo and zeus. The Job Services on apollo and zeus might have identical application properties, but their executables would differ, because each host has its own executable file. For more information about Executable properties, see “Executable Job Service Properties” on page 153.

To add applications:

1. Click to view Job Service application properties.
2. Click Add to open Application Properties.
3. Supply a name and description.
4. Enter a command string to pass to the application when it runs.
   Use one method:
   - Select a predefined template.
Enter a command string in the field provided.
Build a command string using command tokens.

5 Click OK, then click the Executable tab to define the executable properties for the application.
See “Executable Job Service Properties” on page 153.

Command Tokens

You can use command tokens to build command strings to pass to applications when they run:

- **$CMD**—Full path and name of the executable.
- **$PARAMS**—Parameters defined for the program. You can set prompt and default values for individual parameters in program properties.
- **$PROGRAM**—Program to run. Examples of programs include shell scripts, SQL scripts, or Oracle Reports.
- **$BPROGRAM**—Program name with the file extension removed. Use this in combination with hardcoded text to specify a name for an error file, a log file, or another such file. An example would be `log=$BPROGRAM.log`.
- **$FLAGS**—Flags associated with the program.
- **$EFLAGS**—Flags associated with the executable or an instance of it. All jobs associated with this executable use these flags.
- **$DBCONNECT**—Database connect string associated with the program. If set, end users cannot specify a connect string at runtime.
- **$DBUSERNAME**—Database user name associated with the program. If set, end users cannot specify a user name at runtime.
- **$DBPASSWORD**—Database password associated with the program. If set, end users cannot specify a password at runtime.
- **$BPUSERNAME**—User name. If the user name is required as an input parameter to the job, specifying this token instructs the system to include the user name in the command line automatically, rather than prompting the user.

Command String Examples

Example command string template that runs Oracle Reports:

```
$CMD userid=$DBUSERNAME/$DBPASSWORD@$DBCONNECT report=$PROGRAM
destype=file desname=$BPROGRAM.html batch=yes errfile=$BPROGRAM.err
desformat=html
```

When the tokens in the above command string for Oracle Reports are replaced with values, the command executed in Job Service looks like this:

```
r30run32 userid=scott/tiger@Brio8 report=inventory destype=file
desname=inventory.html batch=yes errfile=inventory.err desformat=html
```
Example command string template that runs shell scripts on a Job Service running on UNIX:

$CMD $PROGRAM $PARAMS

When the tokens in the above command string for running shell scripts are replaced with values, the command executed in Job Service looks like this:

sh runscript.sh p1 p2 p3

Example command string template that runs batch files on Job Service running on a Windows system:

$PROGRAM $PARAMS

When the tokens in the above command string for running batch files are replaced with values, the command executed in the Job Service looks like this:

Runbat.bat p1 p2 p3

Executable Job Service Properties

Executable properties provide information about running applications used by Job Service:

- **Executable**—Location of the executable program for the application (full path and executable name); must be co-located with Job Service.
- **Flags**—Value used in the command line for the token $EFLAGS, which represents the flags associated with the program.
- **Environment Variables**—Environment variables associated with the application, for example, $PATH, $ORACLE_HOME.

Only Job Service has executable properties.

Managing Hosts

The Defined Hosts dialog box lists the currently defined hosts in Reporting and Analysis and identifies the host name and platform.

Topics that describe how to add, modify, and delete hosts:

- “Adding Hosts” on page 153
- “Modifying Hosts” on page 154
- “Deleting Hosts” on page 154

Adding Hosts

After you install services on a computer, you must add the computer as a host in Workspace.
To add hosts:

1. Click |Add| and click Add.
2. Supply a host name and the platform used by the host.

**Caution!**
The host name cannot start with numerals. Hyperion Interactive Reporting Data Access Service and Hyperion Interactive Reporting Service do not work if host names start with numerals.

3. Click **OK**.

Reporting and Analysis pings the host to make sure it is on the network. If the ping fails, an error message is displayed.

After Reporting and Analysis successfully pings the host and validates the host name, Reporting and Analysis adds the host and lists it in the Defined Hosts dialog box.

4. Click **OK**.

**Note:**
If you change the host name, you must restart Reporting and Analysis services and Job Service in order for the host to take effect.

**Modifying Hosts**
You modify a host to change its platform designation.

To modify hosts:

1. Click |Modify|.
2. Select a host from the list, and click **Modify**.
3. Select a platform for the host, and click **OK**.

**Deleting Hosts**
You cannot delete a host if services are installed on it.

To delete hosts:

1. Click |Delete|.
2. Select a host from the list and click **Delete**.
3. When prompted, click **Yes** to delete the host, and click **OK**.
Managing Repository Databases

Reporting and Analysis uses repository databases to store and manage application metadata:

● “Defining Database Servers” on page 155
● “Changing the Services Repository Database Password” on page 157
● “Changing the Repository Database Driver or JDBC URL” on page 158

Defining Database Servers

The Defined Database Servers dialog box lists the currently defined Reporting and Analysis repository database servers, identifying the database server name, type, and location (host) of each.

Topics that describe how to manage database servers using RSC:

● “Database Server Properties” on page 155
● “Adding Database Servers” on page 155
● “Adding Job Service Database Connectivity” on page 156
● “Modifying Database Servers” on page 156
● “Deleting Database Servers” on page 157

Database Server Properties

Properties for all repository database servers are Database type, User name, and Password, where user name is the default user name used by Job Service for running SQR Production Reporting programs on the database server (used when a database user name and password are not supplied when storing jobs in the repository).

Adding Database Servers

➤ To add database servers:

1 Click

2 Click Add.

3 Supply this information:

   ● Name—Alphanumeric name for the database server you want to add that is at least five characters.
   ● Database type—Type of database server you are using.
   ● Host—Host where the database server resides.
● **User name**—Default user name used by the Job Service for running SQR Production Reporting programs on the database server. Used if the job owner does not supply a database user name and password when importing a given job.

● **Password**—Valid password for user name.

4 Click **OK**.

### Adding Job Service Database Connectivity

To facilitate database connectivity, after you add a database server, you must associate it with Job Service. Doing so enables Job Service to eliminate network traffic by running a program that connects directly to a local database.

Multiple Job Services can access the same database. For example, you can define three Job Services within one Reporting and Analysis domain, and each Job Service can point to a given XYZ database loaded on a large UNIX server.

When asked to run a given report that uses data on the XYZ database, Service Broker dispatches the job to one of three Job Services associated with the database. Should a computer hosting one Job Service go down, Service Broker automatically routes the job to another Job Service.

➤ To associate database servers with Job Service:

1 Select a Job Service to associate with the database server.

2 Click ![image]

3 Select the **Database** tab, and click **Add**.

4 Select the database server to associate with Job Service, and click **OK**.

5 Supply this information:

   ● **Connectivity information**—Information needed depends on the database type. For example, for an Oracle database, enter a connect string.

   ● **Environment variables**—Required only to execute SQR Production Reporting jobs against the database. Used to specify database information and shared library information that may be required by SQR Production Reporting.

      For example: name = ORACLE_SID, value = PAYROLL

6 Click **OK**.

### Modifying Database Servers

➤ To modify database servers:

1 Click ![image]

2 Select a database server from the list and click **Modify**.

3 Make changes as necessary (see “Database Server Properties” on page 155), and click **OK**.
Deleting Database Servers

➤ To delete database servers:

1 Click .
2 Select a database server from the list and click Delete.
3 When prompted, click Yes to verify database deletion, and click OK.

Changing the Services Repository Database Password

When you change the password that the services use to access the repository database, the order of steps is critical. Carefully read all instructions before performing them.

Caution!

Make sure to change the password in Workspace before changing it in the database. If you perform the steps in the wrong order, you may lose the ability to run Workspace.

➤ To change the repository database password:

1 From RSC, select Name Service, Repository Service, or Event Service.
2 Click Show item properties, and select the Storage tab.
3 Change the password and click OK.
4 Repeat step 1 through step 3 for all Name Services, Repository Services, and Event Services with the same database account, making certain to enter the same password for each one.
   If these services use different database accounts, perform this step only for those that use the account whose password you are changing.
5 Close RSC.
6 In LSC, click Show host properties, and select the Database tab.
7 Change the password and click OK.
   This password property (like the other properties on the Database tab) applies to all LSC services on the local host, all of which use one database account.
   For more information about LSC, see Chapter 8, “Configuring LSC Services.”
8 Repeat step 6 and step 7 on every host that contains LSC services, making certain to enter the password the same way each time.
9 If you are using the same database for row-level security, change the password for row-level security from the Administer module.
10 Stop the Reporting and Analysis services.
11 Change the password in the database, making certain it matches the password entered for Reporting and Analysis services.
12 Restart the services.

**Changing the Repository Database Driver or JDBC URL**

When you change the driver for the repository database or its URL, the order of steps is critical. Carefully read all instructions before performing them.

---

**Caution!**

If you perform steps in the wrong order, you may lose the ability to run Workspace.

---

If parts of the JDBC URL change, such as the database server name, port number, or SID, you must update the JDBC URL property. To do so, perform the JDBC URL portions of the instructions.

1. **To change the database driver:**
   1. Stop Reporting and Analysis services.
   2. Back up `config.dat` and `server.xml`, in `\BIPlus\common\config`.
   3. Start LSC.
   4. Click Show host properties, and select the Database tab.
   5. Update the database driver and JDBC URL properties.
   6. Start the ConfigFileAdmin utility.
      - See “Modifying config.dat” on page 162.
   7. Type 3 to select Get Name Server Data.
      - You can use this data listing to preserve all Name Service properties you do not wish to change.
   8. Type 4 to select Modify Name Server Data.
   9. As the program prompts you for each property, refer to the listing you just displayed, and enter the same values for all properties except Name Server JDBC URL and Name Server JDBC Driver.
10. Enter the values for Name Server JDBC URL and Name Server JDBC Driver properties; for example:
    ```
    Name Server JDBC URL
    jdbc:brio:oracle://brio8host:1521;SID=brio8
    Name Server JDBC driver
    com.brio.jdbc.Oracle.OracleDriver
    ```
11. Run this SQL against the repository database:
    ```sql
    update v8_jdbc set jdbc_driver='newDriverName', jdbc_url='newJdbcUrl'
    ```
    For example:
    ```sql
    ```
12. Update the variable BP_DBDRIVER.
BP_DBDRIVER is defined in Install Home/bin/set_common_env.bat (or set_common_env.sh).

By default, this is set to:

HYPERION_HOME\common\JDBC\DataDirect\3.4.1\lib\hyjdbc.jar

13 Add a JDBC driver to HYPERION_HOME\common\JDBC and set BP_DBDRIVER to the full path of the JAR files.

14 Restart the services.

Managing Jobs

Job Service compiles and executes content-creation programs or jobs. Job Service listens for Workspace job requests (such as requests initiated by users from the Scheduler module), manages program execution, returns the results to the requester, and stores the results in the repository.

Three job types that Workspace can store and run:

- **Interactive Reporting**—Jobs created with Interactive Reporting Studio.
- **SQR Production Reporting**—Secure or nonsecure jobs created with SQR Production Reporting studio.
- **Generic**—Jobs created using other applications (for example, Oracle or Crystal Reports) through a command line interface.

For Interactive Reporting jobs, no special configuration is necessary. Every Job Service is preconfigured to run Interactive Reporting jobs.

For users to run SQR Production Reporting or generic jobs, you must configure a Job Service to run the report engine or application program. One Job Service can run multiple types of jobs, as long as it is configured for each type (except Interactive Reporting).

Topics that explain how to configure Job Service to run jobs.

- “Optimizing Enterprise-Reporting Applications Performance” on page 160
- “From Adding Job Services to Running Jobs” on page 160

See also “Adding Applications for Job Service” on page 151 and “Executable Job Service Properties” on page 153.

Note:

The system automatically creates a textual log file (listed beneath the job) for every job it runs. You can suppress all job log files by adding the Java system property, -Dhqlogfile_isprimary=false, to the common services startup script. You must stop and restart all services. See Chapter 2, “Administration Tools and Tasks”, for more information on stopping and starting the services.
Optimizing Enterprise-Reporting Applications Performance

The Reporting and Analysis architecture is designed for distributed, enterprise implementations. For optimum performance:

- Replicate Job Services (multiple Job Services assigned to a given data source on different computers) to increase overall reliability and decrease job turn-around time.
- Install Job Service on the same computer as the database to conserve valuable network resources.

Note:

Normally, there should be one Job Service on a given host. You can configure a Job Service to run multiple applications.

To run jobs against an enterprise application, configure these parameters:

- **Host**—Physical computer identified to the system by host name.
- **Job Service**—Job Service on the host using RSC.
- **Application**—Third-party vendor application designed to run in the background. Application examples include SQR Production Reporting, Oracle Reports, or public domain application shells such as PERL.
- **Program**—Source used to drive an invocation of an application. For example, a user might submit a SQR Production Reporting program that generates a Sales report to a SQR Production Reporting application on a given host through Job Service.

From Adding Job Services to Running Jobs

This topic synthesizes the configuration process for running jobs into one set of steps, taking you from service installation to successful job execution.

To run jobs in Workspace, complete these steps, which are explained in detail in other topics throughout this document:

1. Install the report application’s executable on the host where you run Job Service. Use the installation program that comes with the report application.
2. On the host, install Job Service software from the Reporting and Analysis services installation DVD.
3. Configure Job Service using RSC.
4. Start Job Service.
5. For generic jobs, add an application and executable. For SQR Production Reporting jobs, add a SQR Production Reporting executable, and define a database server and database connectivity properties. Interactive Reporting jobs do not need special configuration.
6. Import a job (a report or program) to run against the application.
   - This can be an Interactive Reporting, SQR Production Reporting, or generic job.
7. Users can now run the job from Explore.
Using the ConfigFileAdmin Utility

Topics that explain how to modify the config.dat file and how to configure access to process documents and job output using the ConfigFileAdmin utility:

- “About config.dat” on page 161
- “Modifying config.dat” on page 162
- “Specifying Explicit Access Requirements for Interactive Reporting Documents and Job Output” on page 163
- “Setting the ServletUser Password when Interactive Reporting Explicit Access is Enabled” on page 163

About config.dat

Regardless of whether services are running on Windows or UNIX, and whether they are running in the common services’ process or in separate processes, RSC services always use config.dat to begin their startup process.

config.dat resides in \BIPplus\common\config. All RSC services on a host (within an Install Home) share a config.dat file. If you distribute RSC services across several computers, each computer has its own config.dat.

When Name Service starts, it reads config.dat to get database connectivity and logon information. All other RSC services read this file to get their password, host, and port for Name Service. Name Service gets its configuration information directly from the database. Other RSC services connect to Name Service to get their configuration information.

config.dat uses plain ASCII text. Passwords contained in the file are encrypted, and you can modify them only with RSC or the ConfigFileAdmin utility. This ensures that only people who know the config.dat password can modify the service passwords in the file. See “Modifying config.dat” on page 162.

To modify configuration information in config.dat, modify service properties using RSC. RSC writes your changes to config.dat.

Sample config.dat File

```
[Setup]
Key=30721481
Password=AD5FA5E0B71DE9E7F142DD39548571725AC01E801EBAB4345FEA588317398F002C8E468B6F8D7A

[NameServer]
Name=NS1_ggodbee1.hyperion.com
Host=ggodbee1.hyperion.com
Password=36873B0EB76584AC8E468B6F8D7A
SAPassword=C0CBBC76515E6DCEB7A5E3DF4017F56AC54BA329010DA09794989183D1BFF61A02C8E468B6F8D
JDBC_URL=jdbc:hyperion:sqlserver://localhost:1433;DatabaseName=db446
Login=db446
JDBC_DRIVER=hyperion.jdbc.sqlserver.SQLServerDriver
Port=1498
```
Modifying config.dat

You view or modify information in config.dat by using a simple utility run from a command line, named ConfigFileAdmin.bat (Windows) or ConfigFileAdmin.sh (UNIX). This file is in \BIPlus\bin.

To run the ConfigFileAdmin utility, specify the config.dat password on a command line after the file name. For example, with the default password, you would type ConfigFileAdmin.bat administrator (on Windows) or ConfigFileAdmin.sh administrator (on UNIX).

Tasks you can accomplish with the ConfigFileAdmin utility:

- Deleting services
- Changing services passwords
- Changing the password for access to config.dat
- Changing the ServletUser password

The main menu of the ConfigFileAdmin utility offers these commands:

0) Exit
1) Create New Config File
2) Load Existing Config File
3) Get Name Server Data
4) Modify Name Server Data
5) Add Service Agent
6) Delete Service Agent
7) List Service Agents
8) Get Service Agent Password
9) Change Service Agent Password
10) Change Config File Password
11) Validate Password
12) Encode Password
13) Encrypt Password
14) Miscellaneous Commands Menu

To list the properties of Name Service, such as its database logon name and password, select option 3.

When the Reporting and Analysis installation creates a config.dat file, it assigns a default password of administrator. This differs from the administrator account password. As a matter
of system security, you should change the config.dat password using the ConfigFileAdmin utility, by selecting option 10.

You can use option 4 to modify the database password that Name Service uses to connect to the repository database, or you can use RSC to do so.

**Specifying Explicit Access Requirements for Interactive Reporting Documents and Job Output**

By default, no explicit access to Interactive Reporting database connections is required to process Interactive Reporting documents or job outputs using the plug-in or Workspace. To require explicit access, as when a database is associated with Interactive Reporting documents or job output, use the ConfigFileAdmin utility.

➤ To require explicit Interactive Reporting database connection access to process documents and job out:

1. **At a command line, go to the \BIPlus\bin directory of the Reporting and Analysis server. Enter:**
   
   configfileadmin password

2. **Type 14.**

   -
   -
   -
   1) Validate Password
   12) Encode Password
   13) Encrypt Password
   14) Miscellaneous Commands Menu

   Supply the requested information for the database (user) name, database password, database URL, and database driver. You can find this information in the \<xref> section of the server.xml file.

3. **Type 1.**

   0) Exit
   1) Toggle the SC_ENABLED flag for ServletUser (enables/disables feature)
   2) Update the ServletUser password and re-generate properties file.

   This flag is stored in the repository.

4. **After toggling, restart the server, because Repository Service caches this information.**

**Setting the ServletUser Password when Interactive Reporting Explicit Access is Enabled**

The special user ServletUser has read-only administrative permissions. When the SC_ENABLE flag is set to true, ServletUser sends a request for access to Interactive Reporting documents or job output on behalf of users without explicit access to the Interactive Reporting database
connection associated with the document or job output. When the SC_ENABLE flag is set to false, ServletUser cannot make such a request. Only users with explicit access given by the importer to the Interactive Reporting database connection associated with the Interactive Reporting document or job output have access.

The password for ServletUser is updated in the repository and stored, encrypted, in the sc.properties file. The directory in which this file is located depends on the servlet engine you are using:

- For all application servers except WebLogic 8.x, deployed Web applications and associated files are in HYPERION_HOME/deployments/AppServNameAndVersion.
- For WebLogic 8.x, deployed web applications and associated files are in ProductHome/AppServer/InstalledApps/AppServName/version.

➤ To change the ServletUser password:

1. **At a command line, go to the \BIPlus\bin directory of the Workspace server. Enter:**
   
   `configfileadmin password`

2. **Type 14.**
   
   11) Validate Password
   12) Encode Password
   13) Encrypt Password
   14) Miscellaneous Commands Menu

3. **Type 2.**
   
   0) Exit
   1) Toggle the SC_ENABLED flag for ServletUser (enables/disables feature)
   2) Update the ServletUser password and re-generate properties file.

4. **Enter the information requested.**

5. **Manually update the sc.properties file on all Workspace servlet installations.**
About LSC

LSC enables you to modify properties of installed LSC services:

- Analytic Bridge Service (ABS)—Also known as Extended Access for Hyperion Interactive Reporting Service
- Assessment (Harvester) Service (HAR)
- Authentication Service (AN)
- Authorization Service (AZ)
- Global Service Manager (GSM)
- Hyperion Interactive Reporting Service (BI)
- Hyperion Interactive Reporting Data Access Service (DAS)
- Hyperion Interactive Reporting Job Service (BI)
- Local Service Manager (LSM)
- Logging Service (LS)
- Publisher Service (PUB)
- Session Manager (SM)
- Super Service (BPS)
- Update (Transformer) Service (TFM)
- Usage Service (UT)

LSC only modifies LSC service properties; it neither creates nor removes LSC services. To add services, use the Reporting and Analysis installation program. To remove services, see “Using the ConfigFileAdmin Utility” on page 161.

LSC cannot configure services on a remote host (nor in another Install Home on the same host) or on a system with no GUI capability. LSC edits repository information and server.xml (in
which holds configuration information only for services in that Install Home.

**Note:**

Multiple Reporting and Analysis installations, or *Install Homes*, may reside on one host computer. A server installation is a set of installed services in one Install Home directory that run in one process space. If a host has two Install Home directories, they require two separate process spaces. LSC always edits `server.xml` for its own Install Home.

## Starting LSC

To start LSC:

1. **Start Service Configurator.**
   - *UNIX:* Run the `ServiceConfigurator.sh` file, installed in `/BIPlus/bin`.

2. **Select Module > Local Service Configurator,** or click ![module](image) 

3. **Enter your user name and password.**

**Note:**

If you log on with a normal user account, some fields, such as the Trusted Password and Pass-through configuration information, are read-only. For full access to all functionality, you must be logged in as a user who is provisioned with the Reporting and Analysis Global Administrator role.

## Using LSC

LSC lists services that are installed in the Reporting and Analysis installation (*Install Home*) from which LSC is running, along with basic properties of the highlighted service. Toolbar buttons represent functions you perform using LSC.

<table>
<thead>
<tr>
<th>LSC Toolbar Button</th>
<th>Tooltip</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="exit" /></td>
<td><strong>Exit</strong></td>
<td>Closes LSC after user's confirmation</td>
</tr>
<tr>
<td><img src="image" alt="host" /></td>
<td><strong>Show host properties</strong></td>
<td>Displays the properties of the current Install Home for editing</td>
</tr>
<tr>
<td><img src="image" alt="item" /></td>
<td><strong>Show item properties</strong></td>
<td>Displays properties of the selected service for editing</td>
</tr>
</tbody>
</table>
## Modifying LSC Service Properties

Administrators use LSC to modify LSC service properties. Not every service has all property groups. The property groups that all or most LSC services have, and other properties of each service, are described in these topics:

- “Common LSC Properties” on page 167
- “Assessment and Update Services Properties” on page 168
- “Hyperion Interactive Reporting Service Properties” on page 168
- “Hyperion Interactive Reporting Data Access Service Properties” on page 170
- “Hyperion Interactive Reporting Job Service Properties” on page 172

➤ To view or modify most LSC service properties, double-click the service name or select the service name and click ![File].

➤ To view or modify GSM or LSM properties (which do not appear in the Local Service list box), click ![Plus] to display General Properties.

➤ To modify host properties, click ![Remove].

A properties page is displayed. Select a tab to view or modify a group of properties.

### Common LSC Properties

All LSC services have general properties, of which three are standard:

- **Service Name**—Read-only name of the service, assigned during installation
- **Run Type**—Controls whether a service is started with other services (by the startCommonServices script or with Core Services)

  Setting Run Type to Start makes the service active, so it starts with the others. Setting Run Type to Hold inactivates the service, so it does not start with the others. The Hold setting is useful for troubleshooting, to temporarily limit which services start.

- **Log Level**—See “Configuring Logging Levels” on page 204

Services with only standard general properties:

- Analytic Bridge Service
Assessment and Update Services Properties

In addition to standard general properties (that is, Service Name, Run Type, and Log Level; see “Common LSC Properties” on page 167), Assessment Service and Update Service have these general properties:

- **Work directory**—Name of the directory where the service’s temporary files are stored
- **Max concurrent threads**—Maximum number of concurrent threads the service supports
- **Request Queue polling interval**—Frequency with which the service checks for the “Request Queue lock timeout” setting
  
  For example, to set the service to poll every 30 seconds, type **30**.
- **Request Queue lock timeout**—Number of seconds after which the Request Queue lock timeout expires
- **Clear log entries after**—Number of hours after which log entries should be cleared

Hyperion Interactive Reporting Service Properties

This service has additional general properties, and special font considerations for UNIX systems, as described in these topics:

- “Hyperion Interactive Reporting Service General Properties” on page 168
- “Fonts for UNIX” on page 169

Hyperion Interactive Reporting Service General Properties

In addition to standard general properties (that is, Service Name, Run Type, and Log Level; see “Common LSC Properties” on page 167), Hyperion Interactive Reporting Service properties has these general properties:

- **Cache Location**—Directory name where the service’s temporary files are stored
  
  For example, to set cache location to the D drive, type **D:\temp**.
- **Max Concurrent Requests**—Maximum number of concurrent requests this service supports; requests that exceed this setting are blocked; default is 5,000
  
  For example, to block the number of concurrent requests after 4999, type **5000**.
- **Polling Interval**—Frequency in seconds with which the service checks the “Document Unload Timeout” setting
  
  Default is 180 seconds; cannot be less than 15 seconds.

- **Min. Disk space (MB)**—Minimum disk space required to service requests; default is 10

  For example, to allocate 10 MB as the minimum disk space, type `10`.

- **Document Unload Timeout**—Inactive time in seconds after which documents are unloaded from memory to conserve system resources; default is 900

  For example, to retain documents in memory no longer than 30 minutes after last use, type `1800`.

- **Document Unload Threshold**—Number of open documents that activates the document unloading mechanism; default is 15

  For example, to set the maximum number of open documents to 15, type `15`.

### Fonts for UNIX

If UNIX users want Interactive Reporting documents to have a consistent look and feel, you must make Type1, TrueType, or OpenType fonts available to Hyperion Interactive Reporting Service. For a Windows-like look and feel, download Microsoft’s TrueType Web fonts.

If you have Type1, TrueType, or OpenType fonts, and a `fonts.dir` file, perform step 5 on page 169 and step 6 on page 170 to make these fonts available to Hyperion Interactive Reporting Service.

If you have Type1, TrueType, or OpenType fonts, but no `fonts.dir` file, you must perform step 4 on page 169, step 5 on page 169, and step 6 on page 170.

➤ To make Microsoft’s TrueType Web fonts available to Hyperion Interactive Reporting Service when you do not have Type1, TrueType, or OpenType fonts:

1. Download Microsoft TrueType Web fonts from [http://sourceforge.net/projects/corefonts/](http://sourceforge.net/projects/corefonts/) or other source.

2. Create a directory, `directory`.

3. Extract each CAB file (*.exe) into the newly created directory using the `cabextract` utility in `\BIPlus\bin`.

   ```
   \BIPlus\bin\cabextract -d directory <CAB file>
   ```

4. Create a `fonts.dir` file in the directory containing font files using the `ttmkfdir` utility in `\BIPlus\bin`.

   ```
   \BIPlus\bin\ttmkfdir -d directory -o directory\fonts.dir
   ```

5. Set the environmental variable `BQ_FONT_PATH` to the directory where `fonts.dir` was created.

   Add this variable to the start-up script to save your changes.

   ```
   BQ_FONT_PATH=directory, export BQ_FONT_PATH
   ```
This environment variable can contain colon-separated paths to directories containing fonts.dir.

6 Restart Hyperion Interactive Reporting Service.

Hyperion Interactive Reporting Data Access Service Properties

Topics that discuss Hyperion Interactive Reporting Data Access Service properties:

- “Hyperion Interactive Reporting Data Access Service General Properties” on page 170
- “Hyperion Interactive Reporting Data Access Service Data Source Properties” on page 170
- “Adding Data Sources for Hyperion Interactive Reporting Data Access Service” on page 172

Hyperion Interactive Reporting Data Access Service General Properties

In addition to standard general properties (that is, Service Name, Run Type, and Log Level; see “Common LSC Properties” on page 167), these general properties can be used to fine-tune Hyperion Interactive Reporting Data Access Service performance.

- **Relational Partial Result Size**—Maximum number of bytes that a block of results data from a query can contain when sent from Hyperion Interactive Reporting Data Access Service to the client
  
  Default value is 500000; minimum is 1

- **Multidimensional Partial Result Row Count**—Maximum number of multidimensional data table rows that a block of results data from a query can contain when sent from Hyperion Interactive Reporting Data Access Service to the client
  
  Default value is 512; minimum is 1

- **Minimum Idle Time**—Minimum number of seconds to retain query data in memory for the client retrieval before assuming that the client is disconnected
  
  - DAS_RESULT_RESOURCE_MINIMUM_IDLE_TIME SECONDS
  
  - Default value is 180; minimum is 0

- **Data Source Connection Reap Interval**—Frequency in seconds with which the system checks for unused database connections and closes them.
  
  Default value is 180; minimum is 60

Hyperion Interactive Reporting Data Access Service Data Source Properties

Data source properties page lists all defined data sources for Hyperion Interactive Reporting Data Access Service. From this page, you can modify data source properties, or create and remove data sources.
These properties apply to all Hyperion Interactive Reporting Data Access Service data sources:

- **Connectivity Type**—Data source database driver; must be installed on the host for Hyperion Interactive Reporting Data Access Service
- **Database Type**—Database type for the data source
  Whether Hyperion Interactive Reporting Data Access Service can connect to databases is determined by Interactive Reporting database connections and database drivers installed.
- **Hostname/Provider**—Database host name or logical data source name
  For OLE DB database connections, this is the OLE DB Provider identifier.
- **Server/File** (OLE DB only)—Server file or data source name used for database connections

**Note:**
Connectivity Type, Database Type, Name of Data Source, and Server/File properties are used only to route requests to Hyperion Interactive Reporting Data Access Service. Database client software to connect to the requested database must be installed and properly configured on each host where Hyperion Interactive Reporting Data Access Service is configured to accept routed requests for database access.

- **Maximum Connections to DB**—Maximum number of connections permitted from Hyperion Interactive Reporting Data Access Service process to the data source, using the current driver
  Default value is 2048; minimum is 0
- **Maximum Queue Size**—Maximum number of requests that can simultaneously wait to obtain a connection to the database server
  Default value is 100; minimum is 0
- **Minimum Idle Time**—Minimum number of seconds to keep open unused database connections
  - DAS_MINIMUM_CONNECTION_IDLE_TIME_SECONDS
  - Default value is 180; minimum is 0
- **Maximum Connections in Pool**—Maximum number of unused database connections to keep open for a database user name and Interactive Reporting database connection combination
  Default value is 1000; minimum is 0
- **Minimum Pool Idle Time**—Minimum number of seconds to keep unused connections for a database user name and Interactive Reporting database connection combination in memory
  - DAS_MINIMUM_EMPTY_CONNECTION_POOL_IDLE_TIME_SECONDS
  - Default value is 180; minimum is 0
Adding Data Sources for Hyperion Interactive Reporting Data Access Service

When adding data sources, these Hyperion Interactive Reporting Data Access Service properties, which are set using LSC, must match the specified corresponding Interactive Reporting database connection properties, which are set in Interactive Reporting Studio:

<table>
<thead>
<tr>
<th>Hyperion Interactive Reporting Data Access Service Properties (in LSC)</th>
<th>Interactive Reporting Database Connection Properties (in Interactive Reporting Studio)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity type</td>
<td>Connection software</td>
</tr>
<tr>
<td>Database type</td>
<td>Database type</td>
</tr>
<tr>
<td>Hostname/Provider</td>
<td>Host or provider (OLE DB)</td>
</tr>
</tbody>
</table>

Interactive Reporting Studio uses Interactive Reporting database connections to determine which Hyperion Interactive Reporting Data Access Service to use; Hyperion Interactive Reporting Data Access Service uses Interactive Reporting database connections to connect to databases.

Hyperion Interactive Reporting Job Service Properties

In addition to standard general properties (that is, Service Name, Run Type, and Log Level; see “Common LSC Properties” on page 167), the general properties for Hyperion Interactive Reporting Job Service properties match those for Hyperion Interactive Reporting Service (see “Hyperion Interactive Reporting Service General Properties” on page 168) with the exception of one additional property:

Max Concurrent Job Requests—Maximum number of concurrent Interactive Reporting jobs to run; default is 5.

Modifying Host Properties

Services that are installed in one Install Home directory on a host computer are collectively called an Install Home and run in one process space. Most hosts have only one Install Home. For hosts that have multiple Install Homes, host properties belong to the Install Home whose LSC you are running, rather than the host computer.

Use LSC to modify these host properties:

- “Host General Properties” on page 173
- “Host Database Properties” on page 173
- “Host Shared Services Properties” on page 174
- “Host Authentication Properties” on page 175
To modify host properties:

1. From the LSC main window, click
2. Modify General, Database, Shared Services, or Authorization properties as necessary.
3. Click OK.

### Host General Properties

Host general properties include specification of the system’s GSM and LSM:

- **Installation Directory**—Read-only path to the directory where Reporting and Analysis services are installed
- **Cache Files Directory**—Directory where temporary files are stored for caching of user interface elements and content listings
- **Root Log Level**—Logging level for all services (see “Configuring Logging Levels” on page 204)
- **GSM: Name**—Read-only name of GSM that manages this Install Home’s services
- **GSM: Service Test Interval**—Frequency in minutes with which GSM checks that registered services on all hosts are running
- **GSM: Host**—Computer on which GSM is installed
- **GSM: Port**—Port number on which GSM is running; default is 6800
- **LSM: Log Level**—Logging level for LSM (see “Configuring Logging Levels” on page 204)
- **LSM: Service Test Interval**—Frequency in minutes with which LSM checks that other services are running
- **LSM: GSM Sync Time**—Frequency in seconds with which LSM synchronizes its information with GSM

### Host Database Properties

Host database properties, such as the database driver and the database password used by the services, relate to the repository database:

- **Database Driver**—Name of the driver used to access the database
  
  This is database-dependent, and should be changed only by experienced administrators. If you change the database driver, you must change other files, properties, data in the database, and the Java classpath. See “Changing the Repository Database Driver or JDBC URL” on page 158.
- **JDBC URL**—URL for Java access to the database using the JDBC driver
  
  If you change the JDBC URL, you must change other files, properties, and data in the database. See “Changing the Repository Database Driver or JDBC URL” on page 158.
- **User Name**—User name that services use to access the database that contains their metadata
This name must match for all installations using the same GSM.

- **Password**

Host database properties rarely should be changed, but if modifications are necessary, then edit these files, which contain database information for services, to keep them in sync:

- **server.xml**—Modify using LSC
- **config.dat**—Modify using ConfigFileAdmin utility
- **Every RSC service**—You must set properties on every RSC service individually
- **startCommonServices script**
- **All service-specific start scripts**

Instructions for changing some of the database properties are given in “Changing the Services Repository Database Password” on page 157, and in “Changing the Repository Database Driver or JDBC URL” on page 158.

### Host Shared Services Properties

Shared Services properties provide information about the computer that hosts the Shared Services installation to which this Reporting and Analysis installation (Install Home) is registered.

Modifying the Host, Port, and CSS Config File URL properties changes only the values stored in the repository. It does not re-register the application to Shared Services on the specified host and port. You must use Oracle’s Hyperion® Configuration Utility™ (see the Shared Services Registration and Deregistration tasks) to do so.

**Note:**

You can edit Shared Services properties only if you have the Reporting and Analysis Global Administrator role.

- **Host**—Name of the computer hosting Shared Services
- **Port**—Port for Oracle’s Hyperion® Shared Services User Management Console; default port number is 58080
- **Project name**—Shared project name; defined through Shared Services
- **Application name**—Shared application name; defined through Shared Services
- **CSS Config File URL**—URL used to retrieve external configuration information from Shared Services
  - **Default URL**—URL stored in the database and used by all services
  - **Use this URL instead for this server**—Used to override the URL just for this Install Home (typically, it is not necessary to set this property)

The CSS Config File URL is stored in `BpmServer.properties`, the location of which depends on your servlet engine. For example, with Apache Tomcat, this file is in:
Host Authentication Properties

Host authentication properties relate to the use of trusted password and pass-through configuration values, which apply to jobs and Interactive Reporting documents:

- **Set trusted password**—Enables the use of a trusted password
- **Use users’ login credentials for pass-through**—Enables pass-through using the user’s logon credentials
- **Allow users to specify credentials for pass-through**—Enables pass-through using the credentials the user specifies in Preferences

If no credentials are specified in Preferences, an error message displays each time users attempt to open Interactive Reporting documents or run jobs.

Modifying Properties in portal.properties

A few properties are modified by editing the `portal.properties` text file, in `BIPlus\lib\msgs`.

To edit `portal.properties`, use a plain text editor. To change a property value, edit the string that follows its equal sign (=). Change only value strings; do not modify the file in any other way. When saving the file, be sure to preserve its name and file extension.

Properties configured in `portal.properties`:

- **bqDocsTimeOut**—Interval in seconds that services should wait for Hyperion Interactive Reporting Service to open Interactive Reporting documents; default is 1000
- **bqlogfilenameprefix**—Log file name for Interactive Reporting job output, without the file extension; default is `bqlog`
- **bqExpireDurationForEmailNotice**—Interactive Reporting job’s expire duration to send e-mail notification in hours; default is 24
- **defaultCategoryUuid**—Root folder name; default is `REPORTMART`
- **listenerThreadPollingPeriod**—Frequency in minutes with which the system should poll for externally triggered events; default is 3
- **multiValueSQRParamSeparator**—Character to use as a separator between values of a multi-value parameter in SQR Production Reporting jobs; default is a plus sign (+)
- **outputDescr**—Text that should precede job output descriptions; default is `Output from`
- **outputLabel**—Name of a set of job output files, which is composed of `outputLabel` value followed by job name; default is `Output`

- **outputLabel1**—Part of a job output label identifying a *cycle* of a multi-cycle Interactive Reporting job, where 0 represents the cycle name and 1 represents the job name; default job name is `Output`

- **schedJobDataSeparator**—Delimiter used to generate the schedule job type strings used in queuing; default is a colon (``)

- **UseUserLocaleForSorting**—Enables Workspace listings to be sorted based on the end-user's locale; default is 0 (sorting is based on the locale of the metadata database)

  When Workspace contains data from multiple locales, set this property to 1 (true) to enable multi-locale sorting.
Configuring the Servlets

In This Chapter

Using Servlet Configurator ..................................................................................................... 177
Modifying Properties with Servlet Configurator ................................................................................ 178
Zero Administration and Interactive Reporting ................................................................................ 189
Load Testing Interactive Reporting ............................................................................................. 191

Using Servlet Configurator

Configuring the servlets enables Workspace to more precisely meet the needs of your
organization. Configuration settings depend on aspects of your organization’s environment such
as how the system handles user passwords, usage volume, and how users interact with
Workspace.

Note:
For information on customizing parameter forms for SQR Production Reporting and generic
jobs, see the Hyperion Workspace User’s Guide. For information on customizing Web module
user interfaces, see the Hyperion Workspace Developer’s Guide.

You can configure many details of servlet behavior with Servlet Configurator, which configures
all locally installed servlets. Servlet Configurator and the configuration file it edits, ws.conf, are
installed automatically when the servlets are deployed.

The location of ws.conf depends on your servlet engine. For example, for Apache Tomcat,
ws.conf is in:

Install Home\AppServer\InstalledApps\Tomcat\5.0.28\Workspace\webapps
\workspace\WEB-INF\config

Note:
If you replicated the servlets in your system and want to make the configurations match, copy
the ws.conf file from one servlet host to the other, and check for host-specific settings.
To start Servlet Configurator:

- **Windows:** Select Start > Programs > Hyperion > Reporting and Analysis > Utilities and Administration > Servlet Configurator.
- **UNIX:** Run the `config.sh` file, installed in `/BIPlus/bin`.

The configuration toolbar is displayed above the navigation pane and contains these buttons:

<table>
<thead>
<tr>
<th>Servlet Configurator Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📊</td>
<td>Saves all configuration settings (Keyboard shortcut: Alt+S)</td>
</tr>
<tr>
<td>🔄</td>
<td>Sets the visible configuration settings (that is, those currently displayed in the right-hand frame) to their default values</td>
</tr>
<tr>
<td>🔄</td>
<td>Sets all configuration settings to their default values</td>
</tr>
<tr>
<td>🌐</td>
<td>Displays the online help</td>
</tr>
</tbody>
</table>

## Modifying Properties with Servlet Configurator

Servlet Configurator displays a list of Property folders that you use to view and modify servlet properties.

To view or modify the properties in a folder:

1. Click 📁 next to the folder.
2. Make changes.

   See these topics for property descriptions:

   - “User Interface Properties” on page 178
   - “Personal Pages Properties” on page 182
   - “Internal Properties” on page 184
   - “Cache Properties” on page 186
   - “Diagnostics Properties” on page 188
   - “Applications Properties” on page 188

3. Save your settings.
4. Make the settings effective by restarting the servlets.

### User Interface Properties

- “User Interface: Login Properties” on page 179
User Interface: Login Properties

Login properties pertain to the common logon mechanism for all servlets:

- **LoginPolicy class for $CUSTOM_LOGIN$**—Name of the class that implements the LoginPolicy interface (the fully package-qualified name without the .class extension); specify only if you are using a custom logon implementation.

  For more information about custom logon, see the loginsamples.jar file in BIPlus\docs\samples.

- **Custom username policy**—Possible values are CUSTOM_LOGIN$ (the custom policies), $HTTP_USER$, $REMOTE_USER$, $SECURITY_AGENT$, or to $NONE$:
  - Set to $NONE$ unless you implement a custom logon or configure transparent logon.
  - If set to a value other than $NONE$, uses the specified user name policy to obtain the user name for all users logging on to Workspace servlets.
  - Use $CUSTOM_LOGIN$ only if you use a custom implementation for the username value.
  - If set to $SECURITY_AGENT$, the “Custom password policy” must be set to $TRUSTEDPASS$.

- **Custom password policy**—Possible values are CUSTOM_LOGIN$ (the custom policies), $HTTP_PASSWORD$, $TRUSTEDPASS$, $USERNAME$, or to $NONE$:
  - Set this option to $NONE$ unless you implemented a custom logon or configured transparent logon.
  - If set to a value other than $NONE$, uses the specified password policy to obtain the password for all users logging on to Workspace servlets.
  - Use $CUSTOM_LOGIN$ only if you use a custom implementation for the password value.
  - If the custom user name policy is set to $SECURITY_AGENT$, the “Custom password policy” must be set to $TRUSTEDPASS$.

- **Allow users to change their password**—Displays the Change Password link in Workspace Preferences for native users in Shared Services:
  - If you do not select this option, the change password link is not available to users.
  - If you configured transparent logon, do not select this option.
- **Set default server** to—IP address or name for the server hosting GSM, and optional port number, where server and port are separated by a colon (:); if port number is omitted, the default GSM port number of 6800 is used, for example:
  
  `apollo:2220`—Uses port 2220
  
  `apollo`—Uses default port 6800

### User Interface: Configuration Properties

The “AlternateHSServiceURL” property enables you to launch Shared Services from within Workspace using a URL that differs from the URL used to register Shared Services (and the URL used by the Workspace Web application to access Shared Services internally).

This is useful if you want to use a secure HTTP connection (`https:`) for security-sensitive communication.

### User Interface: Localization Properties

Localization properties enable you to customize time, date, and language settings for locales:

- **Format times using**—Servlets can display time fields in a 12-hour (AM/PM) format or in a 24-hour format; for example, in a 24-hour format, the servlets display 6:30 PM as 18:30

- **Date display order**—Servlets can display dates in month day year order (for example, May 1 2004) or day month year order (for example, 1 May 2004)

- **Use locale-sensitive sort**—Sorts names using the default locale (locale-sensitive sorts are slightly slower but more user-intuitive; For example, ‘A’ and ‘a’ are sorted together in a locale-sensitive sort, but not in a lexicographical sort)

If no locale-sensitive sort is defined, the servlets use a lexicographical sort.

- **Default local language code**—Lowercase, two-letter code for the language most commonly used by servlet end users (for example, `en` for English or `fr` for French)


  Users can use the servlets in the language of their choice (if templates exist in that language) by setting their browser language option. (In Internet Explorer, select Tools > Internet Options, General tab, Languages button. In Firefox, select Tools > Options, Language button.)

  Used in conjunction with country codes and local variants to determine (1) the set of templates the servlet reads upon startup, and (2) in what language to display pages. The system checks for localization settings in this order (until a non-default value is found):

  1. User browser
  2. Localization properties for the servlet (iHTML or Data Access)
  3. Default localization properties for Workspace servlets
  4. Default locale specified on the Web server

  Localization settings found are used in this order (until a default value is found):
1. Language code
2. Country code
3. Local variant

For example, Explore checks the user browser first. If it has no language setting, then Explore, which does not have its own localization settings, checks the default localization settings. This check begins with “Default local language code.” If that setting is specified (is not Default), Explore checks “Default local country code” to refine localization. If it too is specified, Explore checks “Default local variant.” If, on the other hand, “Default local language code” is set to Default, Explore skips the default localization settings and checks the locale for which the servlets host is configured.

- **Default local country code**—Uppercase, two-letter code for the country (for example, US for United States, CA for Canada, and so on)

  Used in conjunction with the language code and local variant parameters to obtain and display user data

  For a complete list of codes, go to:


  Used only if “Default local language code” is specified (is not set to Default); if country code is set to Default, the iHTML servlet uses the language code value to determine user.

- **Default local variant**—Optional localization property used for a finer granularity of localization in messages for a user audience with matching language and country codes; for example, if you specify a variant of WEST_COAST, the system uses it to deliver specialized data, such as time for the local time zone

  Used only if “Default local country code” is not set to Default; if “Default local variant” is set to Default, the servlet uses the Default local language code and Default local country code values to determine the user locales.

**User Interface: Subscription Properties**

The “Enable subscription features” option enables users to subscribe to items from Explore. If this option is not selected, users cannot receive notifications when items are modified.

**User Interface: Job Output Properties**

Job Output properties enable you to customize the format and display of job output:

- Display HTML icon when displaying SQR Production Reporting job output in listing pages
- Display SPF icon when displaying SQR Production Reporting job output in listing pages
- Output format to display after a SQR Production Reporting job is run
User Interface: SmartCut Properties

The “Show SmartCut as link” property displays SmartCuts as links. If set to off, which is the default setting, SmartCuts display as plain text.

User Interface: Color Properties

Color properties enable you to customize the colors of the servlets’ user interface main frame. These properties apply only to user interface HTML templates, not to JSPs. Therefore, for consistency across the user interface, when you change colors here (in Servlet Configurator), do so in the stylesheets, in the css directory (for example, Install Home\servlets \deployment\CSS).

● General Properties
  ○ Main frame: Background color—Background color of the main frame (or pane). Does not apply to Personal Pages. If you leave this option blank, your platform’s default background color is used.
  ○ Personal Page wizard: Background color—Personal Page wizard is the sequence of pages displayed after a user chooses New Personal Page. Wizard pages have two colors, a main background color and the color of the top and bottom borders.
  ○ Personal Page wizard: Border color—See preceding paragraph.
● Title Property—Sets the underline color when titles are underlined.
● Text Properties
  ○ Regular text color—Regular text is most of the text on servlet pages. If you leave this option blank, the browser default is used.
  ○ Link text color—Color of links which the user has not (recently) chosen.

Personal Pages Properties

Personal Pages properties enable you to control user capabilities in Personal Pages:

● “Personal Pages: General Properties” on page 182
● “Personal Pages: Publish Properties” on page 183
● “Personal Pages: Generated Properties” on page 183
● “Personal Pages: Syndicated Content Property” on page 184
● “Personal Pages: Color Scheme Properties” on page 184

Personal Pages: General Properties

General properties for Personal Pages:

● Max Personal Pages per user—Set to 20 or less; default is 5
● **Max initial published Personal Pages**—Maximum number of Personal Pages to be copied from published Personal Pages when a user first logs on; set to at least 1 less than the value of "Max Personal Pages per user"; default is 2.

● **Users can choose default Personal Page**—Default is enabled
  ○ Users change their default by putting the desired default Personal Page at the top of the list on the My Personal Pages page in the servlets
  ○ When disabled, users cannot delete or reorder the default Personal Page
  ○ To ensure that users see the Personal Page containing the Broadcast Messages every time they log on, disable this option

● **Show headings of Content Windows on Personal Pages**—Content windows are displayed with headings (title bars); enabled by default

**Personal Pages: Publish Properties**

Publish properties control the options available to end-users for publishing their Personal Pages for others to use; at least one of the last three properties must be enabled:

● **Location**—Folder path and name that contains published Personal Pages; must be located in the `/Broadcast Messages` folder.
  
  Default value is `/Broadcast Messages/Personal Page Content`, which is not browsable by default

● **Show publisher’s groups**—Enables end users to give permissions to their own groups; enabled by default

● **Allow publisher to enter group name**—Enables end users to give permission to a specified group; enabled by default

● **Allow publishing to all users**—Enables end users to give permissions to all users; enabled by default

**Personal Pages: Generated Properties**

Generated Personal Page properties involve the Personal Page that is generated by the servlet the first time that users log on to Workspace. You can prepare versions of this page for different users.

● **Show My Bookmarks**—Generated Personal Page includes the *My Bookmarks* content window; enabled by default

● **Show Exceptions Dashboard**—Generated Personal Page includes the *Exceptions Dashboard*; enabled by default

● **Number of folders**—Number of pre-configured folders (subfolders of the `/Broadcast Messages` folder) that are displayed on the generated Personal Page; default is 3

● **Number of File Content Windows**—Number of displayable items in pre-configured folders (subfolders of the `/Broadcast Messages` folder) that are displayed as content windows on the generated Personal Page; default is 1
● **Default color scheme**—Default color scheme for generated Personal Page and the *Edit Personal Page* page

**Personal Pages: Syndicated Content Property**

The “Syndicated Content” property specifies the location of syndicated content, the default for which is /Broadcast Messages/Syndicated Content.

**Personal Pages: Color Scheme Properties**

Color scheme properties let you prepare sets of colors for users to apply to their Personal Pages. Four color schemes are set to default values. You can rename each one and set its colors for parts of a Personal Page.

- **Name**—Required
- **Headings color**—Background color of the heading (title bar) of each content window
- **Background color**—Background color of content windows in the main (wide) column
- **Text color**—Color of servlet-generated text on Personal Pages, such as the names of content windows
- **Link color**—Color of the text of servlet-generated links on a Personal Page, such as bookmarks in *My Bookmarks*
- **Broadcast Messages color**—Color of the heading of each Broadcast Messages content window
- **Header background color**—Background color of content windows in the optional header area at the top of a Personal Page
- **Footer background color**—Background color of content windows in the optional footer area at the bottom of the page
- **Left column background color**—Background color of content windows in the optional narrow column on the left side of a Personal Page
- **Right column background color**—Background color of content windows in the optional narrow column on the right-hand side of a Personal Page

**Internal Properties**

Internal properties control how servlets or the Workspace server works:

- “Internal: Redirect Property” on page 185
- “Internal: Cookies Properties” on page 185
- “Internal: Transfer Property” on page 185
- “Internal: Jobs Property” on page 185
- “Internal: Upload Property” on page 186
- “Internal: Temp Property” on page 186
Note:
The session-timeout value (see “Workspace Configuration Information” on page 22) is configured on the servlet engine. All Reporting and Analysis Web applications should have session timeouts set to greater than 10 minutes.

**Internal: Redirect Property**

The “Redirect URLs using” property enables the servlets to redirect URLs using HTTP or JavaScript. HTTP redirection is more efficient and therefore preferred.

**Internal: Cookies Properties**

These properties concern the cookies that the servlets create and use:

- **Keep cookies between browser sessions**—Saves information between browser sessions. The user name last used to log on is saved and used for subsequent logon instances.
- **Encrypt cookies**—Encrypts saved cookies.

**Internal: Transfer Property**

The transfer property, “Pass data using streams instead of files,” controls how data is passed between services and servlets.

If enabled, servlets retrieve files from services using streamed input and output (I/O) and a direct connection instead of temporary file system storage. Data is transferred out-of-band over a separate socket connection between Repository Service and servlets.

If disabled, data is transferred in-band and stored in a file (or in memory if the data is less than 500 KB) for servlets (in a temporary directory) and Service Broker. Data is transferred from Repository Service to Service Broker, and to servlets.

In general, you should enable this option because streamed I/O is more efficient. If your system has a firewall between the servlets and the services, however, and the servlets cannot open additional sockets for file transfer, then you should disable this option.

**Note about firewalls:** When this option is enabled, the system opens a socket for a file transfer. The operating system generates the port number and you cannot control this number. A firewall, however, prohibits access through random port numbers. Therefore, you must disable this option, which causes file transfers to use the open socket in use by Service Broker.

**Internal: Jobs Property**

The jobs property, “Show confirmation screens for,” sets the number of seconds that the confirmation screens are displayed when running a background job.
**Internal: Upload Property**

The upload property, “Max file size allowed for publish,” sets the maximum size for files that users can import into the repository. Default setting is 100 MB.

**Internal: Temp Property**

The temp property specifies the location of the Reporting and Analysis /temp directory.

**Cache Properties**

Cache properties set limits on how long the servlets can cache various data. These properties affect the responsiveness of the user interface, so setting them involves a trade-off between performance and the freshness of displayed data.

*Cache folders for* property can be described in three ways: (1) maximum time to cache folders, in seconds; (2) maximum delay between when a modification is made to a folder in the repository and when the user sees the change in Explore; (3) maximum time interval during which users see old folder contents.

Increasing the value of *Cache folders for* makes pages display more quickly to the user, but increases the length of time that the user sees stale folder contents. Decreasing the value of *Cache folders for* reduces the duration that the user can see stale folder contents, but slows the display of pages.

Topics that describe Cache properties:

- “Cache: Objects Properties” on page 186
- “Cache: System Property” on page 187
- “Cache: Templates Property” on page 187
- “Cache: Notification Property” on page 187
- “Cache: Browser Property” on page 187

**Cache: Objects Properties**

Objects properties concern the caching of object types:

- **Number of folders cached**—Size of the cache for folders; default is 200
- **Cache folders for**—Maximum time in seconds to cache folders (that is, the limit for the delay between changes to a folder’s contents and Explore’s display of the changes); set to zero or greater; default is 3600
  
  User sees old folder contents for no more than the number of seconds specified here.
- **Cache browse queries for**—Maximum time in seconds for changes to browse queries in the Workspace servers to be reflected in the servlets; set to zero or greater; default is 60
- **Cache jobs for**—Maximum time in seconds for changes to jobs in the Workspace servers to be reflected in the servlets; set to zero or greater; default is 60
Cache parameter lists for—Maximum time in seconds that the servlets cache job parameter lists; default is 60

Cache published Personal Pages for—Maximum time in seconds that the servlets cache the content of the Personal Page Content folder; must be greater than zero; default is 60

Note that this cache is refreshed whenever a Personal Page is published using the Personal Pages servlet.

Cache Content Windows on Personal Pages for—Maximum time in seconds for changes to Broadcast Messages on a Personal Page to be reflected in the Personal Pages servlet; must be greater than zero; default is 60

Cache Content Windows being modified for—Maximum time in seconds that Explore or Administer module caches content while it is being modified; default is 180

Cache list items for—Maximum time in seconds that item or resource lists are cached; default is 900

Max items to cache for listing—Maximum number of items in a listing that are cached; default is 100

**Cache: System Property**

The system property, “Cache system properties for,” specifies the number of seconds the servlets should retain system property information before refreshing it from the server. The default value is 1200.

Note that refreshing system properties makes the updated settings effective only for users who have not yet logged on. Users who are logged on when a refresh occurs are not affected.

**Cache: Templates Property**

The Templates property, “Cache parsed HTML templates,” controls whether the servlets cache templates. It is enabled by default. While testing customized templates, however, it is useful to disable this option so that template changes display immediately.

**Cache: Notification Property**

The Notification property, “Refresh notifications every,” specifies how frequently the View Jobs Status page in the Schedule module is refreshed; that is, the maximum amount of time between Event Service issuing a notification and the notification appearing on View Jobs Status pages. Values can be zero or greater seconds.

**Cache: Browser Property**

The Browser property, “Max Interactive Reporting job outputs listed for modification,” specifies the maximum number of job output collections to list in the Versions area of an Interactive Reporting job’s properties page. Accordingly, the maximum number of output collections whose properties can be modified.
**Diagnostics Properties**

Configuration Log properties are used for diagnostic purposes:

- **Logging Service Server**—Host name of the server on which Logging Service resides
- **Configuration**—Path of Servlet Configurator log configuration file, `servletLog4jConfig.xml` (the default can be used)

**Applications Properties**

- “Applications: URL Properties” on page 188
- “Applications: iHTML Properties” on page 188
- “Applications: Data Access Servlet Properties” on page 188

**Applications: URL Properties**

The URL properties must match the servlet locations specified in `web.xml`:

- Browse
- Administer
- Personal Pages
- Job Manager
- iHTML
- Data Access

**Applications: iHTML Properties**

These properties pertain only to iHTML servlet:

- **Clear disk cache after**—Maximum time interval between clearing of disk cache, in seconds
  - Changes the `WebClient.Applications.iHTML.DiskCachePollingPeriod` property in `ws.conf`
  - Default is 300 seconds
- **Terminate idle iHTML session after**—Number of seconds for iHTML servlet to wait for a response from Hyperion Interactive Reporting Service before timing out
  - Changes the `WebClient.Applications.iHTML.BQServiceResponseTimeout` property in `ws.conf`
  - Default is 1800 seconds
  - If exceeded, Hyperion Interactive Reporting Service does not respond

**Applications: Data Access Servlet Properties**

These properties pertain only to Data Access servlet:
- **DAS Response timeout**—Number of seconds that Data Access servlet should wait for a response before timing out:
  - Changes the `WebClient.Applications.DAServlet.DASResponseTimeout` property in `ws.conf`
  - Default is 1800 seconds
- **Enable Zero Administration**—Identifies the release number of the most up-to-date version of Interactive Reporting on the server and triggers the downloading of the Interactive Reporting Web Client when a user selects a link to an Interactive Reporting document
- **Hyperion Interactive Reporting Client Polling Time**—Frequency (in seconds) with which Interactive Reporting Web Client sends polling requests to prevent idle connection timeouts
  Administrators can control the frequency of messages that keep an Interactive Reporting Web Client connection active, thereby allowing queries of varied lengths to process from Interactive Reporting Web Client when running behind a proxy or firewall with finite idle-connection timeouts.
  - Changes the `WebClient.Applications.DAServlet.PollingTimeSec` property in `ws.conf`
  - Default is 1800 seconds

**Note:**
This value should be less than other timeouts between Interactive Reporting Web Client and Data Access servlet; for example, Application Server session timeout, Web Server connection timeout, Web Server thread idle timeout, Proxy connection idle timeout, and so on.

- **Allow multiple browser windows for Interactive Reporting Web Client**—Determines whether Interactive Reporting documents open in one or multiple browser windows when using Interactive Reporting Web Client:
  - Changes the `WebClient.Applications.DAServlet.OpenNewWindow` property in `ws.conf`
  - Default is true (Interactive Reporting documents open in multiple browser windows)
Use this feature to display multiple instances of one Interactive Reporting document, or several Interactive Reporting documents, simultaneously. When multiple browser windows are opened for the same Interactive Reporting document, the last saved document is the one stored in the repository.

---

**Zero Administration and Interactive Reporting**

Zero Administration identifies the release number of the most up-to-date version of Interactive Reporting on the server. When a user chooses a link to an Interactive Reporting document or job from Workspace or by using a SmartCut, Zero Administration is triggered and the Oracle's Hyperion® Interactive Reporting Web Client download starts. The user has the option to download the online help files or use the help files from the Web server.
Zero Administration files (JSP, HTML, XPI, and CAB files) are hosted on the Web server file system. Interactive Reporting release numbers are stored in the registry for Firefox and Internet Explorer browsers.

Available Interactive Reporting capabilities are determined by the user roles and adaptive states. The higher-level access functions include processing database queries and the full analytical features of Oracle's Hyperion® Interactive Reporting Studio.

Topics that provide details on Zero Administration:

- “6x Server URL Mapping” on page 190
- “Client Processing” on page 190

### 6x Server URL Mapping

Users may encounter problems using locally saved, release 6x Interactive Reporting documents when their Web server deployment changes, or when you migrate to another release of Reporting and Analysis. You can configure URL mappings to automatically redirect to other URLs when Interactive Reporting is installed.

To configure URL redirection, add commands to zeroadmin.jsp that establish required redirections for each deployment of the servlets. (The location of this file depends on your servlet engine. For example, for Apache Tomcat, this file is in Install Home\AppServer\InstalledApps\Tomcat\5.0.28\Workspace\webapps\workspace\jsp\dataaccess\zeroadmin.)

These mappings are made by adding calls to the Map6xUrlTo8() method and should be added to the CustomizeInstallForIE(insight) function.

The Map6xUrlTo8(Old_URL, New_URL) method establishes a URL mapping. Passing an empty string as New_URL cancels the URL redirection. Clear6xUrlMap() function removes all URL redirections established so far. The CustomizeInstallForIE(insight) function only runs when Interactive Reporting is downloaded. Mappings are saved in the Windows registry for use with locally saved documents. If the mappings are to be updated dynamically (once per session), then the call to the CustomizeInstallForIE(insight) function should also be made from the Zero Administration main function.

**Example**

```javascript
function CustomizeInstallForIE(insight) {
    insight.Map6xUrlTo8("http://<brio6x_host>:<brio6x_web_port>/ods-isapi/ods.ods","http://<hyperion9x_host>:<hyperion 9x_web_port>/workspace/dataaccess/Browse")
}
```

### Client Processing

When an Interactive Reporting document is opened in Explore, the Web browser retrieves and parses the HTML documents from the Web server. The JSP logic for Zero Administration, which is included in these HTML files, runs in the client’s Web browser.
The `zeroadmin.jsp` file is retrieved from the Web server. Release numbers from that file are compared to release numbers on the client computer. There are three possible outcomes:

- If no release number is found on the client, the user is prompted to install.
- If the numbers are equal (meaning the client release number matches the `zeroadmin.jsp` file), or if the client release is greater than the `zeroadmin.jsp` version, the Interactive Reporting document is opened using the previously installed Interactive Reporting release.
- If the release number on the client is less than that in `zeroadmin.jsp`, the user is prompted to upgrade their client product.

Web browsers can interrogate Interactive Reporting to find out the release number. You can view this information by locating the DLL files (for example, `axbqs32.dll` under Internet Explorer, or `npbqs32.dll` under Firefox) and displaying their file properties.

Most popular Web browsers allow automatic download and installation and provide a digital certificate for an extra layer of security.

The JSP automatically provides the correct application (plug-ins for Windows in a browser-compatible file format).

---

**Load Testing Interactive Reporting**

Interactive Reporting uses unique request IDs for requests sent to the server, and expiring keys to encrypt the DB credentials and SQL string sent over the wire. Because of this, customers using load-testing tools such as Mercury LoadRunner, Seque SilkPerformer, and so on, have difficulty conducting their tests.

To load test Interactive Reporting:

1. **Assign unique request IDs to the `CURRENT_REQUEST_ID` URL parameter.**

   One way to generate the unique IDs for LoadRunner is to use this date and time stamp, and virtual user name:

   ```
   Url=Browse?REQUEST_TYPE=getSectionMap&DOC_NAME={BQY_Files}.bqy&DOC_UUID={par_sDocUUID}&DOC_VERSION=1&MULTI_PART=0&CURRENT_REQUEST_ID=
   (DateTimeStamp){UserRuntimeID}"", "Referer="", ENDITEM,
   ```

   where:

   - **DateTimeStamp** is the date and time stamp parameter type with format `%Y%m%d%H%M%S`
   - **UserRuntimeID** is a virtual user ID parameter type with format `%03s`

2. **Enable static key encryption for recording the scripts and running the scripts within Workspace.**

   This setting is not recommended for production environments.

3. **Set the three properties described in these topics:**

   - “Data Access Servlet Property” on page 192
   - “Hyperion Interactive Reporting Data Access Service Property” on page 192
“Hyperion Interactive Reporting Service Property” on page 192

Note:
Setting only one of these properties can cause processing (running of Interactive Reporting jobs, querying from Interactive Reporting, querying from Workspace) to fail, because the source and target encryption schemes do not match.

Data Access Servlet Property

➢ To load test the Data Access servlet, add this property to `ws.conf` and restart the Web server:
`WebClient.Applications.DAServlet.UseStaticKeyForEncryption=true`

Hyperion Interactive Reporting Data Access Service Property

➢ To load test the Hyperion Interactive Reporting Data Access Service:

1 Add this property to `server.xml`

   `<property defid="0ad70321-0001-08aa-000000e738090110" name="USE_STATIC_KEY_FOR_ENCRYPTION">true</property>`

   Make sure the property is defined inside the `<properties>` subnode of the `<service type="DataAccess">` node and outside of this node: `<propertylist defid="0ad70321-0002-08aa-000000e738090110" name="DAS_EVENT_MONITOR_PROPERTY_LIST">`

2 Restart the Hyperion Interactive Reporting Data Access Service for all Install Homes.

Hyperion Interactive Reporting Service Property

➢ To load test Hyperion Interactive Reporting Service:

1 Add this property to `server.xml`

   `<property defid="0ad70321-0001-08aa-000000e738090110" name="USE_STATIC_KEY_FOR_ENCRYPTION">true</property>`

   The property must be inside the `<properties>` subnode of `<service type="BrioQuery">` node and outside of this node: `<propertylist defid="0ad70321-0002-08aa-000000e738090110" name="BQ_EVENT_MONITOR_PROPERTY_LIST">`

2 Restart Hyperion Interactive Reporting Service for all Instal Homes.
Backup Strategies

In This Chapter

Overview ....................................................................................................................... 193
What to Back Up................................................................................................................ 193
General Backup Procedure ......................................................................................................194
Maintenance Mode for System Backups ......................................................................................194
Backing Up the Reporting and Analysis File System..........................................................................195
Sample Backup Script...........................................................................................................197
Backing Up the Repository Database ..........................................................................................198
Backing Up the Workspace Servlets ........................................................................................198

Overview

Standard data center policies for database backups include incremental daily backups and weekly full backups with off-site storage to protect an organization’s investment. When you back up Reporting and Analysis, you should plan the backup in the same way that you plan other database backups.

What to Back Up

You must back up the following items in your system:

● File system, which contains Reporting and Analysis content and other system information (including files in other directories and on other hosts)
● Repository database, which contains user and item metadata
● Report registry keys from the same point in time (Windows only)
● Shared Services

Note:

For information about backing up Shared Services, see the Hyperion Shared Services Installation Guide.

Reporting and Analysis maintains an item repository in the native file system and stores metadata, or descriptive information, about each user and artifact in an RDBMS.
Note:
To recover data, restore the database and file system backups (and registry if required), and restart the services.

General Backup Procedure

➤ To back up Reporting and Analysis:
1 Shut down Reporting and Analysis services.
   You can shut down services manually, or you can place services in maintenance mode. See “Maintenance Mode for System Backups” on page 194.
2 Back up the Reporting and Analysis file system.
3 Back up the Reporting and Analysis repository database.
4 Save the backup (on tape or CD).

Note:
If you use Windows, export the Reporting and Analysis registry key. If you use UNIX, backup the /etc/rc or the /etc/init.d boot startup scripts.

Maintenance Mode for System Backups

Rather than manually shutting down the Reporting and Analysis services before backing up your Reporting and Analysis system, you can dynamically suspend the services by putting the services in maintenance mode.

To place services in maintenance mode, the first time, you must run a SQL statement to insert a row in the v8_service_mode table (in the Reporting and Analysis repository schema) with the values of v8_service_mode.name="MAINTENANCE" and v8_service_mode.value = 1 (true) or 0 (false). You can delay the start time by setting the value of v8_service_mode.start_delay=nn (where nn is minutes)

For example, if you set v8_service_mode.value=1 and v8_service_mode.start_delay=30, the services start rejecting requests after 30 minutes

Note:
Workspace checks once every minute to determine whether maintenance mode needs to be activated; however, Hyperion recommends that you wait for 5 to 10 minutes before starting the backup.

When services are in maintenance mode:
• New users cannot log on to Workspace.
Users who are logged on cannot perform any actions.

All requests to the services that could result in read/write calls to the database fail.

All services background activity that results in database updates are placed on hold (for example, repository garbage collection, Usage tracking garbage collection).

All Event Service activity, like events fired during maintenance mode, are placed on hold.

All service logs log messages indicating that the service is in maintenance mode. Messages are logged when maintenance mode is turned off.

---

**Backing Up the Reporting and Analysis File System**

There are five backup types, distinguished by when you perform them:

- **Complete**—Backs up the entire system. Your organization’s policies and procedures determine whether and how often you perform a complete backup.

- **Post-installation**—Backs up certain directories, performed after completing an installation and before using the system.

- **Daily incremental**—Backs up only files that are new or modified since the previous day. Daily incremental backups involve directories that contain frequently changing information, such as repository content and log files.

- **Weekly full**—Backs up all files in the directories for which you do incremental backups on a daily basis.

- **As Needed**—Backs up data only after changes are made, rather than on a regular schedule. As-needed backups involve directories containing files that are customizable but are not modified regularly.

HYPERION_HOME contains the Reporting and Analysis products you installed on the host. Subdirectories of HYPERION_HOME include \BIPlus and \common.

---

**Complete Backup**

To back up your system comprehensively, back up HYPERION_HOME. This is the default installation directory for all Hyperion products on a given host.

**Post-Installation**

Immediately after installing, back up these directories:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIPlus\install</td>
<td>All configuration information defined during installation; back up on all hosts and compress each backup</td>
</tr>
</tbody>
</table>
**BIPlus\bin**
Start batch scripts for each service, and the ConfigFileAdmin utility used by the administrator to decode and change passwords (typically, the only password of interest is the RDBMS login password)

**BIPlus\common\config**
Service configuration files used at service startup:
- server.xml
- config.dat
- server.dat

**BIPlus\lib**
JAR files required by one or more Reporting and Analysis components and library files for Job Utilities, LSC, and RSC

**HYPERION_HOME\common\JDBC**
JDBC drivers required to run the Reporting and Analysis services

**HYPERION_HOME\common\ODBC**
Required ODBC drivers

**BIPlus\common\SQR\lib**
Files necessary to manipulate the metadata for versions of SQR Production Reporting

---

**Weekly Full and Daily Incremental**
Back up these directories fully once a week and incrementally every day:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIPlus\logs</td>
<td>Log files for services operating on a computer</td>
</tr>
<tr>
<td>BIPlus\data\RMx_host.hyperion.com</td>
<td>Content (repository files)</td>
</tr>
</tbody>
</table>

---

**As Needed**
Back up the following directories as needed:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIPlus\bin</td>
<td>Start batch scripts for each service, and the ConfigFileAdmin utility used by the administrator to decode and change passwords (typically, the only password of interest is the RDBMS login password)</td>
</tr>
<tr>
<td>BIPlus\common\config</td>
<td>Service configuration files used at service startup, server.xml, and config.dat</td>
</tr>
<tr>
<td>BIPlus\data</td>
<td>Directories associated with services</td>
</tr>
</tbody>
</table>

---

**Reference Table for All File Backups**
The following table lists the directories for all backup types.
<table>
<thead>
<tr>
<th>Directory</th>
<th>Contents</th>
<th>Backup Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIPlus\lib</td>
<td>Library files for Job Utilities</td>
<td>After initial installation</td>
</tr>
<tr>
<td>BIPlus\bin</td>
<td>Reporting and Analysis startup batch scripts for each service, and the ConfigFileAdmin utility used by the administrator to decode and change passwords (typically, the only password of interest is the RDBMS login password) On Windows systems: Setup.exe program file, used to create or delete services running as Windows Services, and to update the Windows Registry information</td>
<td>After initial installation and after any changes are made to start scripts</td>
</tr>
<tr>
<td>HYPERION_HOME\common\JDBC</td>
<td>All drivers required to run the Reporting and Analysis services</td>
<td>After initial installation</td>
</tr>
<tr>
<td>HYPERION_HOME\common\ODBC</td>
<td>All drivers required to run the Reporting and Analysis services</td>
<td>After initial installation</td>
</tr>
<tr>
<td>BIPlus\data\RMx_host.hyperion.com</td>
<td>Content (repository files)</td>
<td>Daily incremental, weekly full (consistent with company backup policy)</td>
</tr>
<tr>
<td>BIPlus\common\config</td>
<td>Service configuration files used at service startup, server.xml, and config.dat.</td>
<td>After initial installation, before and after subsequent service configuration changes that focus on adding and removing services to a given domain</td>
</tr>
<tr>
<td>BIPlus\install</td>
<td>Configuration information defined during installation</td>
<td>Perform after initial installation on each host; back up on each host and compress each backup</td>
</tr>
<tr>
<td>BIPlus\lib</td>
<td>JAR files required by Hyperion components</td>
<td>After initial installation</td>
</tr>
<tr>
<td>BIPlus\logs</td>
<td>Log files for services operating on a computer</td>
<td>Daily incrementals, weekly fulls (consistent with company backup policy)</td>
</tr>
<tr>
<td>BIPlus\common\SQR\lib</td>
<td>Files necessary to manipulate the metadata for versions of Oracle’s Hyperion® SQR® Production Reporting – System 9</td>
<td>After initial installation</td>
</tr>
</tbody>
</table>

**Sample Backup Script**

Sample script for a file-system backup of a Sun Solaris deployment of Reporting and Analysis:

```
#!/bin/sh

Sample Backup Script
```

**Backup Utility:** Solaris Dump  
**Backup Type:** Full Level 0 Dump  
**Backup Frequency:** Weekly run on Saturday at 1 AM

```bash
#!/bin/sh
```
PARMS="0ucbsdf 126 5000 61000"
DEVISE="/dev/rmt/0hn"
CMD="/usr/sbin/ufsdump"
FileSystems="/Hyperion/BIPlus/logs /Hyperion/BIPlus/data/RM1_Solar12"
# ---------------------------------------------------------
#       Perform Level 0 Dump of all listed filesystems
# ---------------------------------------------------------
echo "Starting Backup set for the following filesystems:"
echo ""
for i in $FileSystems
do
echo "backing up filesystem: "
echo $i
$CMD $PARMS $DEVISE $i
done

**Backing Up the Repository Database**

Backup the repository database according to your company policy for database backups, taking into account repository usage volume.

A backup of the Reporting and Analysis repository database is RDBMS- (or vendor-) dependent. For details about the backup procedure for your particular RDBMS, see its documentation.

**Backing Up the Workspace Servlets**

The backup needs of the Workspace servlets are minimal. You should perform a standard post-installation full backup according to your company policy. Thereafter, the only files you need to backup are these servlet files:

- Servlet configuration file, `ws.conf` on Windows, or `wsrun_platform` on UNIX, located in the `/WEB-INF/config` directory of your servlet engine deployment (under `BIPLUS/AppServer`)
- `/WEB-INF/conf/BpmServer.properties`
- Modified files in `/BIPLUS/AppServer`
- `/WEB-INF/web.xml`
- Customized JSPs
- Customized HTML templates
Overview

Administrators can generate log files throughout Reporting and Analysis to help technicians identify system or environmental problems or to help developers debug reports or API programs.

Complete configuration information about your Workspace installation, including GSM and Oracle’s Hyperion® Shared Services information, is available at this URL:

http://hostname:port/workspace/browse/configInfo

where hostname is the name of the Workspace server, and port is the TCP port on which the application server is listening. The default port for Oracle’s Hyperion® Workspace is 19000 if using Apache Tomcat.

Logging Architecture

All log messages are routed through Logging Service and stored in one location. Logging Service writes log messages to one or more files, which can be read using a viewer.

Log4j (version 1.2) is used as the basis for the logging framework and configuration files. Log Management Helper is used by C++ services (Hyperion Interactive Reporting Service and Hyperion Interactive Reporting Data Access Service) in conjunction with the log4j framework and Logging Service.

Reporting and Analysis comes with preconfigured loggers and appenders. Loggers correspond to areas in code (class) where log messages originated. Appenders correspond to output destinations of log messages. You can troubleshoot system components by setting the logging level of loggers.
Log4j

The log4j package enables logging statements to remain in shipped code without incurring heavy performance costs. As part of the Jakarta project, log4j is distributed under the Apache Software License, a popular open source license certified by the Open Source Initiative. Logging behavior is controlled through XML configuration files at runtime. In configuration files, log statements can be turned on and off per service or class (through the loggers) and logging levels for each logger can be set, which provide the ability to diagnose problems down to the class level. Multiple destinations can be configured for each logger.

Main components of log4j:

- **Loggers**—Control which logging statements are enabled or disabled. Loggers may be assigned levels ALL, DEBUG, INFO, WARN, ERROR, FATAL, or INHERIT.
- **Appenders**—Send formatted output to their destinations.

Go to www.apache.org or see *The complete log4j manual* by Ceki Gülcü (QOS.ch, 2003).

Logging Service

Logging Service stores all log files in one location. If Logging Service is unavailable, log messages are sent to backup log files. When Logging Service is restored, messages in backup files are automatically sent to Logging Service, which stores them in log files and deletes the backup files.

Logging Service cannot be replicated.

Log Management Helper

Log Management Helper (LMH) consolidates all logs from Hyperion Interactive Reporting Data Access Service or Hyperion Interactive Reporting Service and sends them to Logging Service.

One LMH process exists for each Hyperion Interactive Reporting Data Access Service and for each Hyperion Interactive Reporting Service per Install Home.

Logging Service consolidates all log messages in separate log files for Hyperion Interactive Reporting Data Access Service and Hyperion Interactive Reporting Service per Reporting and Analysis.

Server Synchronization

Because log files are time-stamped and written in chronological order, time synchronization between servers, which is the responsibility of the administrator, is important. Many products, free and commercial, are available to manage server clock synchronization.

Log File Basics

Topics that provide information about using log files for troubleshooting:
Log File Location

All log files are in HYPERION_HOME\logs\BIPlus on the computer where Logging Service is running.

Services, servlets, process monitors, and Web services log messages centrally using Logging Service.

LSC, RSC, and Calendar Manager log messages locally.

Hyperion Interactive Reporting Service and Hyperion Interactive Reporting Data Access Service Local Log Files

Hyperion Interactive Reporting Service and Hyperion Interactive Reporting Data Access Service have additional log files that are stored in BIPlus\logs, and which collect log messages before these services connect to Logging Service. Log messages in these files are not routed to Logging Service log files. Start-up problems are collected in BIstartup.log and DASstartup.log.

Log File Naming Convention

Each service or servlet has its own log file. In a multiple Install Home installation, all services of one type log their messages to one file. Separate log files are generated for license information, configuration and/or environment information, and stdout messages.

Services and servlets log filenames format:

server_messages_OriginatorType.log

where

OriginatorType is one of these components:

- Servlets
  - BrowseServlet
  - AdministrationServlet
  - PersonalPagesServlet
  - DataAccessServlet
  - iHTMLServlet
- Services:
  - AnalyticBridgeService
  - AuthenticationService
- AuthorizationService
- CommonServices
- DataAccessService
- EventService
- GSM
- HarvesterService
- IntelligenceService
- IRJobService
- IRServiceHelper
- JobService
- LSM
- NameService
- PublisherService
- RepositoryService
- SessionManager
- ServiceBroker
- TransformerService
- Usage Service

- Miscellaneous
  - CalendarManager
  - WebService
  - SDK
  - EventComponent
  - LocalServiceConfigurator
  - RemoteServiceConfigurator
  - Installer

Special log files are:

- **license_messages.log**—Contains license information
- **configuration_messages.log**—Contains basic environment and configuration information
- **name_backupMessages_host_domain__port.log** (where name is the process name)—Contains logging messages when Logging Service is unavailable (for example, BI_PM_sla1_backupMessages_myserver_hyperion_com_6800.log).
- **stdout_console.log**—Contains messages sent to stdout and stderr.
Log Message File Format

All log messages contain this information in the order shown:

- **Logger**—Name of the logger that generated the logging message
- **Timestamp**—Time stamp in coordinated universal time (UTC); ensures that messages from differing time zones can be correlated
  
  The administrator is responsible for time synchronization between servers.
- **Level**—Logging level
- **Thread**—Thread name
- **Sequence number**—Unique number to identify messages with matching time stamps
- **Time**—Time the log message was generated
- **Context**—Information about which component generated the log message
  - **Subject**—User name
  - **Session ID**—UUID of the session
  - **Originator Type**—Component type name
  - **Originator Name**—Component name
  - **Host**—Host name
- **Message**—Log message
- **Throwable**—Stack trace of a throwable error

The format for backup log files match the format for regular log files.

Configuration Log

Basic configuration information is logged to `configuration_messages.log` in `BIPlus/logs`. The file format matches service and servlet log file formats.

This log file contains Java system property information, JAR file version information, and database information.

Configuring Log Properties for Troubleshooting

To troubleshoot Reporting and Analysis, you can configure these logging properties:

- **Logging levels**
- **Loggers**
- **Appenders**
- **Log rotation**

Loggers, logging levels, and appenders are configured in XML files. The log rotation property is a Java system property and is configured in `startcommonservices.bat`. Logging levels for
LSC services, RSC services, and the root logger are configured using LSC and RSC. All other configuration changes are made by editing XML files.

**Configuration Files**

Configuration file types are main and imported: Imported files are used by main files and organize the loggers and appenders into separate XML files.

Main configuration files:
- `serviceLog4jConfig.xml` — Main configuration file for services; in `\BIPlus\common\config\log4j`
- `remoteServiceLog4jConfig.xml` — Main configuration file for Hyperion Interactive Reporting Service and Hyperion Interactive Reporting Data Access Service, and for RSC services when started remotely; in `\BIPlus\common\config\log4j`
- `adminLog4jConfig.xml` — Main configuration file for LSC, RSC, and Calendar Manager
- `servletLog4jConfig.xml` — Main configuration file for the servlets; in `\WEB-INF\config` of the servlet engine deployment

**Note:**
If you change the location of `serviceLog4jConfig.xml` or `remoteServiceLog4jConfig.xml`, you must update the path information stored in `server.xml`. If you change the location of `servletLog4jConfig.xml`, you must update the path information in `ws.conf`.

Imported configuration files:
- `appenders.xml` — Imported by `serviceLog4jConfig.xml`, `servletLog4jConfig.xml`, and `remoteServiceLog4jConfig.xml`
  Appenders can be added by referencing them in `<logger>` and `<root>` elements using `<appender-ref>` elements.
- `serviceloggers.xml` — Imported by `serviceLog4jConfig.xml` and `remoteServiceLog4jConfig.xml`; configure through LSC
- `debugLoggers.xml` — Contains definitions for loggers that can be enabled to debug problems in the services; imported by `serviceLog4jConfig.xml` file and `remoteServiceLog4jConfig.xml`; in `\BIPlus\common\config\log4j`
- `debugLoggers.xml` — Contains definitions for loggers that can be enabled to debug problems in the servlets; imported by `servletLog4jConfig.xml`; in the `\WEB-INF\config` folder of your servlet engine deployment

**Configuring Logging Levels**

Logging levels specify the amount and type of information to write to log files. Except for the inherit level, levels in Table 11 are listed from most verbose to least verbose, and logging levels are cumulative. The default logging level, which is set on root, is WARN; therefore, messages at...
that level or lower (ERROR, FATAL) appear in the log. You can change this for the entire system or per service or servlet. If a given logger is not assigned a level (or its level is set to INHERIT), it inherits the level from its closest ancestor with an assigned level. The root logger resides at the top of the logger hierarchy and always has an assigned level.

Table 11  Logging Levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INHERIT</td>
<td>Uses the logging level set at its closest ancestor with an assigned level; not available at the root level</td>
</tr>
<tr>
<td>ALL</td>
<td>All messages levels</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Minor and frequently occurring normal events; use only when troubleshooting</td>
</tr>
<tr>
<td>INFO</td>
<td>Normal significant events of the application</td>
</tr>
<tr>
<td>WARN</td>
<td>Minor problems caused by factors external to the application</td>
</tr>
<tr>
<td>ERROR</td>
<td>Usually, Java exceptions that do not necessarily cause the application to crash; the application may continue to service subsequent requests</td>
</tr>
<tr>
<td>FATAL</td>
<td>Implies the imminent crash of the application or the relevant sub-component; rarely used</td>
</tr>
</tbody>
</table>

Configuring Loggers

Use RSC to configure RSC service logging levels, which are stored in the database (see “Advanced RSC Properties” on page 147).

Use LSC to configure LSC service logging levels (stored in serviceLoggers.xml) and the root logger (see “Host General Properties” on page 173).

Configure the servlet root logger level in servletLog4JConfig.xml. Configure other servlet loggers in the servlet debug configuration file (debugLoggers.xml).

➤ To configure the servlet root logger:

1. Open \WEB-INF\config\servletLog4JConfig.xml
2. Scroll to the end of the file and change the root logging level.
   For example, change WARN to INFO:

```xml
<root>
    <level value="WARN"/>
    <appender-ref ref="LOG_REMOTELY"/>
</root>
```
3. Save the file.

Configuring Debug Loggers

Debug loggers are activated by changing the logging level from INHERITED to DEBUG. Use these loggers only with help from Hyperion Solutions Customer Support.
Note:

Some Java properties, such as `print_config`, `print_query debug`, and `echo`, are mapped to debug loggers in `\BIPlus\common\config\log4j\debugLoggers.xml`.

**Configuring Appenders**

You can send log messages to multiple destinations by adding appenders, defined in `appenders.xml`, to loggers.

➤ To add appenders to loggers:

1) Locate an appender in `appenders.xml` and copy its name.

2) Open the XML file of the logger to which you want to add this appender.

3) Paste the name of the appender after `<appender-ref ref= under the logger to which you want to add this appender.

For example:

```xml
<appender-ref ref="LOG_LOCALLY_BY_LOGGING_SERVICE"/>
```

4) Save the file.

**Configuring Synchronous or Asynchronous Messaging**

Log messages can be sent synchronously (the default) or asynchronously. Asynchronous mode offers performance advantages, while synchronous mode provides reliability in that all messages get logged. You can change the BufferSize parameter to limit message loss.

➤ To enable asynchronous messaging:

1) Open `appenders.xml` and locate the asynchronous appender.

```xml
<appender name="SEND_TO_LOGGING_SERVICE_ASYNC"
class="org.apache.log4j.AsyncAppender">
```

2) Optional: Change BufferSize.

```xml
<param name="BufferSize" value="128"/>
```

3) Copy the appender name, "SEND_TO_LOGGING_SERVICE_ASYNC".

4) Locate the root logger.

You can change the default appender for the service or the servlet root logger in the XML file.

5) Replace the name of the default appender, "LOG_LOCALLY_BY_LOGGING_SERVICE", with the name of the asynchronous appender, "SEND_TO_LOGGING_SERVICE_ASYNC".

6) Save the file.
Configuring Root Level Appenders

In the services main configuration file, serviceLog4jconfig.xml, the default appender for the root level logs locally by Logging Service. If the server does not contain Logging Service, the appender LOG_REMOTELY is uncommented. You can also uncomment the second appender, LOG_LOCALLY, to log messages remotely and locally.

This code, from serviceLog4jconfig.xml, shows the root level appenders:

```xml
<!-- The following appender should be enabled if the server does not contain the logging service -->
<!-- <appender-ref ref="LOG_REMOTELY"/> -->
<!-- The following appender can be enabled in conjunction with the remote appender to also send log messages locally -->
<!-- <appender-ref ref="LOG_LOCALLY"/> -->
<!-- The following appender should only be enabled if the server contains the logging service -->
<appender-ref ref="LOG_LOCALLY_BY_LOGGING_SERVICE"/>
```

Configuring Log Rotation

You can roll and delete log files by time intervals or by file size. File size log rotation is controlled by CompositeRollingAppender. Time interval log rotation is controlled by CompositeRollingAppender and a Java property in the common services start file.

By default, the system rolls logs every 12 hours, and deletes the oldest log file when the number of logs exceeds five. Log files are created and deleted by originator type (see “Log File Naming Convention” on page 201).

All appenders in XML configuration files are configured to use default values for CompositeRollingAppender. You can configure CompositeRollingAppender properties for each appender separately.

**Note:**

If you want all log files to rotate using matching criteria, change the configuration for each CompositeRollingAppender defined in both appenders.xml files.

➤ To change log rotation settings:

1. Open appenders.xml in \BIPlus\common\config\log4j (for services) or in \WEB-INF\config (for servlets).

2. Locate the CompositeRollingAppender definition and change the properties.

<table>
<thead>
<tr>
<th>RollingStyle</th>
<th>There are three rolling styles:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Roll the logs by size</td>
</tr>
<tr>
<td>2</td>
<td>Roll the logs by time</td>
</tr>
<tr>
<td>3</td>
<td>Roll the logs by size and time</td>
</tr>
</tbody>
</table>
RollingStyle 3 could provide confusing results because naming conventions for logs rolled by time and size differ, and deletion counters do not count logs rolled differently together.

<table>
<thead>
<tr>
<th><strong>DatePattern value</strong></th>
<th>If RollingStyle is 2 or 3, set the time interval to write log messages to another log file. Set the Date Pattern value using the string, yyyy-MM-dd-mm; for example, yyyy-MM-dd-mm means every 60 minutes, yyyy-MM-dd-a means every 12 hours, and yyyy-mm-dd means every 24 hours. Default is every 12 hours.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaxFileSize</strong></td>
<td>If RollingStyle is 1 or 3, when the maximum file size is reached, the system writes log messages to another file. Default is 5MB. You can use KB (kilobyte), MB (megabyte), or GB (gigabyte).</td>
</tr>
<tr>
<td><strong>MaxSizeRollBackups</strong></td>
<td>If RollingStyle is 1 or 3, when the maximum number of log files per originator type (plus one for the current file) is reached, the system deletes the oldest file. Default is 5. Log files rolled by time are not affected by this setting.</td>
</tr>
</tbody>
</table>

The appenders.xml files for server and servlets tell the server when to create another log file, which two parameters. The best practice rolling style is 3, which toggles log files by time or size. The default 5MB log file size is the default for software packages such as e-mail and Web servers.

**Note:**

Best practices recommend that RollingStyle for all entries be set to 3, and that default log file size be set to 1 MB. Log files that exceed 1 MB may slow down the server, with possible outages (the service crashes or needs to be restarted) occurring after the log exceeds 25 MB. Large log files can be problematic to open in a text editor such as Notepad or vi.

Sample CompositeRollingAppender definition:

```xml
<appender name="BACKUP_MESSAGES_FILE" class="org.apache.log4j.CompositeRollingAppender">
  <param name="File" value="${directory}/log/${name}_backupMessages.log"/>
  <!-- Select rolling style (default is 2): 1=rolling by size, 2=rolling by time, 3=rolling by size and time. -->
  <param name="RollingStyle" value="1"/>
  <!-- If rolling style is set to 2 then by default log file will be rolled every 12 hours. -->
  <param name="DatePattern" value=".'yyyy-MM-dd-a"/>
  <!-- If rolling style is set to 1 then by default log file will be rolled when it reaches size of 5MB. -->
  <param name="MaxFileSize" value="5MB"/>
  <!-- This is log file rotation number. This only works for log files rolled by size.-->
  <param name="MaxSizeRollBackups" value="5"/>
  <layout class="com.brio.one.mgmt.logging.xml.XMLFileLayout"></layout>
</appender>
```

3 If RollingStyle is 2 or 3, set the maximum log rotation number in /BIPlus/bin/startcommonservices.bat.

set BP_ROTATIONNUM=-Dlog_rotation_num=5
Analyzing Log Files

This section details how to view log files, log files that are always generated, and log files to look at for troubleshooting.

For information about logs for Shared Services, users, groups, or roles, see the Hyperion Installation and Configuration Troubleshooting Guide.

Viewing Log Files

You can view log messages directly in log files or by using a log viewer. This version of Oracle's Hyperion® Reporting and Analysis – System 9 contains the log4j viewer, LogFactor5, which provides a way to filter and sort log messages.

➤ To use LogFactor5:

1. Copy the name of the LogFactor5 appender, `<appender-ref ref="LF5APPENDER"/>`.
2. Paste the copied code line under the logger in which to use LogFactor5.

```xml
<root>
  <level value="WARN"/>
  <appender-ref ref="LF5APPENDER"/>
  <appender-ref ref="LOG_REMOTELY"/>
</root>
```

LogFactor5 starts automatically when the component to which you added the appender is started. If the component is ongoing, LogFactor5 starts in 30 seconds. The LogFactor5 screen is displayed when logging initializes. Log messages are displayed as they are posted.

Standard Console Log File

The `stdout_console.log` is always generated regardless of the operation being performed or logging level, and represents standard output and standard errors (console output). Some errors that are caught by the application are logged here, as are start-up failures.

Logs for Importing General Content

When creating, modifying, and deleting files or folders, use these logs to analyze errors:

- Server logs
  - `server_messages_PublisherService.log`
  - `server_messages_RepositoryService.log`
  - `server_messages_ServiceBroker.log`
- Client log—`server_messages_BrowseServlet.log`
**Logs for Importing Interactive Reporting Content**

When creating, modifying, and deleting Oracle's Hyperion® Interactive Reporting – System 9 documents or jobs, use these logs to analyze errors:

- Logs for importing general content
- Server logs
  - `hostname_BI1_LSM.log`
  - `hostname_DAS1_LSM.log`
  - `N_BIService.log`
  - `N_das.log`
  - `N_IRJob.log`
  - `N_bi_stderr.txt`
  - `N_bi_stout.txt`
  - `N_bijobs_sterr.txt`
  - `N_bijobs_stout.txt`
  - `N_das_sterr.txt`
  - `N_das_stout.txt`
  - `N_logutil_sterr.txt`
  - `N_logutil_stout.txt`
  - `server_DataAccessService.log`
  - `server_IntelligenceService.log`

- Client logs
  - `server_messages_DataAccessServlet.log`
  - `server_messages_iHTMLServlet.log`

**Logs for Running Jobs**

Job Service runs jobs directly or through Event Service. Use these logs to analyze errors:

- Server logs
  - `server_messages_EventService.log`
  - `server_messages_JobService.log`
  - `server_messages_ServiceBroker.log`
  - `server_messages_DataAccessService.log`
  - `server_messages_IntelligenceService.log`
  - `server_messages_IRJobService.log`
  - `BI1_hostname.log`
Logs for Logon and Logoff Errors

User logon instances require information from multiple areas in the system, each of which can cause errors and logon attempts to fail. Use these logs to analyze logon and logoff errors:

- Server logs
  - server_messages_SessionManager.log
  - server_messages_GSM.log
  - server_messages_LSM.log
  - server_messages_Authentication.log
  - server_messages_Authorization.log
  - server_messages_Publisher.log
  - server_messages_ServiceBroker.log
  - server_messages_RepositoryService.log

- Client logs (servlet)
  - server_messages_BrowseServlet.log
  - server_messages_AdministrationServlet.log
  - server_messages_PersonalPagesServlet.log

Logs for Access Control

Access control is maintained by Authorization Service. Use these logs to analyze access permission errors:

- Server logs
  - server_messages_Authorization.log
  - Logs for the service involved in the operation being performed

- Client logs (servlet)
  - server_messages_BrowseServlet.log
  - server_messages_AdministrationServlet.log
  - server_messages_PersonalPagesServlet.log
Logs for Configuration

Configuration errors for RSC services show at startup in `stdout_console.log` or `server_messages_NameService.log`; `configuration_messages.log` might be helpful.

Information Needed by Customer Support

If a problem occurs and you need help from Hyperion Solutions Customer Support, send all application server logs for the instance being used. If applicable, compress the log directory.

For services and servlets, compress and send all logs under `\BIPlus\logs`. 
Glossary

See bang character (!).

See missing data (#MISSING).

access permissions  A set of operations that a user can perform on a resource.

accessor  Input and output data specifications for data mining algorithms.

account  A dimension that represents an accounting container that identifies the location and primary nature of the data.

account blocking  The process by which accounts accept input data in the consolidated file. Blocked accounts do not receive their value through the additive consolidation process.

account eliminations  Accounts which have their values set to zero in the consolidated file during consolidation.

account type  How an account’s value flows over time, and its sign behavior. Account type options can include expense, income, asset, liability, and equity.

accountability map  A visual, hierarchical representation of the responsibility, reporting, and dependency structure of the accountability teams (also known as critical business areas) in an organization.

accounts dimension  A dimension type that makes accounting intelligence available. Only one dimension can be defined as Accounts.

active service  A service whose Run Type is set to Start rather than Hold.

active user  A user who is entitled to access the system.

active user/user group  The user or user group identified as the current user by user preferences. Determines default user preferences, dynamic options, access, and file permissions. You can set the active user to your user name or any user group to which you belong.

activity-level authorization  Defines user access to applications and the types of activities they can perform on applications, independent of the data that will be operated on.

ad hoc report  An online analytical query created on-the-fly by an end user.

adaptive states  Interactive Reporting Web Client level of permission.

adjustment  See journal entry (JE).

Advanced Relational Access  The integration of a relational database with an Essbase multidimensional database so that all data remains in the relational database and is mapped to summary-level data residing in the Essbase database.

agent  An Essbase server process that starts and stops applications and databases, manages connections from users, and handles user-access security. The agent is referred to as ESSBASE.EXE.

aggregate cell  A cell comprising several cells. For example, a data cell that uses Children(Year) expands to four cells containing Quarter 1, Quarter 2, Quarter 3, and Quarter 4 data.

aggregate function  A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

aggregate limit  A limit placed on an aggregated request line item or aggregated metatopic item.
**aggregate storage database**  The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.

**aggregate view**  A collection of aggregate cells based on the levels of the members within each dimension. To reduce calculation time, values are pre-aggregated and stored as aggregate views. Retrievals then start from aggregate view totals and add up from there.

**aggregation**  The process of rolling up and storing values in an aggregate storage database; the stored result of the aggregation process.

**aggregation script**  In aggregate storage databases only, a file that defines a selection of aggregate views to be built into an aggregation.

**alias**  An alternative name. For example, for a more easily identifiable column descriptor you can display the alias instead of the member name.

**alias table**  A table that contains alternate names for members.

**alternate hierarchy**  A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

**ancestor**  A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

**appender**  A Log4j term for destination.

**application**  (1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system. (2) A related set of dimensions and dimension members that are used to meet a specific set of analytical and/or reporting requirements.

**application currency**  The default reporting currency for the application.

**Application Migration Utility**  A command-line utility for migrating applications and artifacts.

**area**  A predefined set of members and values that makes up a partition.

**arithmetic data load**  A data load that performs operations on values in the database, such as adding 10 to each value.

**artifact**  An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

**asset account**  An account type that stores values that represent a company's assets.

**attribute**  Characteristics of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

**attribute association**  A relationship in a database outline whereby a member in an attribute dimension describes a characteristic of a member of its base dimension. For example, if product 100-10 has a grape flavor, the product 100-10 has the Flavor attribute association of grape. Thus, the 100-10 member of the Product dimension is associated with the Grape member of the Flavor attribute dimension.

**Attribute Calculations dimension**  A system-defined dimension that performs these calculation operations on groups of members: Sum, Count, Avg, Min, and Max. This dimension is calculated dynamically and is not visible in the database outline. For example, using the Avg member, you can calculate the average sales value for Red products in New York in January.

**attribute dimension**  A type of dimension that enables analysis based on the attributes or qualities of dimension members.

**attribute reporting**  A reporting process based on the attributes of the base dimension members. See also *base dimension*.

**attribute type**  A text, numeric, Boolean, date, or linked-attribute type that enables different functions for grouping, selecting, or calculating data. For example, because the Ounces attribute dimension has the type numeric, the number of ounces specified as the attribute of each product can be used to calculate the profit per ounce for that product.
authentication Verification of identity as a security measure. Authentication is typically based on a user name and password. Passwords and digital signatures are forms of authentication.

authentication service A core service that manages one authentication system.

auto-reversing journal A journal for entering adjustments that you want to reverse in the next period.

automated stage A stage that does not require human intervention, for example, a data load.

axis (1) A straight line that passes through a graphic used for measurement and categorization. (2) A report aspect used to arrange and relate multidimensional data, such as filters, pages, rows, and columns. For example, for a data query in Simple Basic, an axis can define columns for values for Qtr1, Qtr2, Qtr3, and Qtr4. Row data would be retrieved with totals in the following hierarchy: Market, Product.

backup A duplicate copy of an application instance.

balance account An account type that stores unsigned values that relate to a particular point in time.

balanced journal A journal in which the total debits equal the total credits.

bang character (!) A character that terminates a series of report commands and requests information from the database. A report script must be terminated with a bang character; several bang characters can be used within a report script.

bar chart A chart that can consist of one to 50 data sets, with any number of values assigned to each data set. Data sets are displayed as groups of corresponding bars, stacked bars, or individual bars in separate rows.

base currency The currency in which daily business transactions are performed.

base dimension A standard dimension that is associated with one or more attribute dimensions. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

base entity An entity at the bottom of the organization structure that does not own other entities.

batch calculation Any calculation on a database that is done in batch; for example, a calculation script or a full database calculation. Dynamic calculations are not considered to be batch calculations.

batch file An operating system file that can call multiple ESSCMD scripts and run multiple sessions of ESSCMD. On Windows-based systems, batch files have BAT file extensions. On UNIX, batch files are written as a shell script.

batch POV A collection of all dimensions on the user POV of every report and book in the batch. While scheduling the batch, you can set the members selected on the batch POV.

batch processing mode A method of using ESSCMD to write a batch or script file that can be used to automate routine server maintenance and diagnostic tasks. ESSCMD script files can execute multiple commands and can be run from the operating system command line or from within operating system batch files. Batch files can be used to call multiple ESSCMD scripts or run multiple instances of ESSCMD.

block The primary storage unit which is a multidimensional array representing the cells of all dense dimensions.

block storage database The Essbase database storage model categorizing and storing data based on the sparsity of data values defined in sparse dimensions. Data values are stored in blocks, which exist only for sparse dimension members for which there are values.

Blocked Account An account that you do not want calculated in the consolidated file because you want to enter it manually.

book A container that holds a group of similar Financial Reporting documents. Books may specify dimension sections or dimension changes.

book POV The dimension members for which a book is run.

bookmark A link to a reporting document or a Web site, displayed on a personal page of a user. The two types of bookmarks are My Bookmarks and image bookmarks.
bounding rectangle  The required perimeter that encapsulates the Interactive Reporting document content when embedding Interactive Reporting document sections in a personal page, specified in pixels for height and width or row per page.

broadcast message  A simple text message sent by an administrator to a user who is logged on to a Planning application. The message displays information to the user such as system availability, notification of application refresh, or application backups.

budget administrator  A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.

build method  A method used to modify database outlines. Choice of a build method is based on the format of data in data source files.

business process  A set of activities that collectively accomplish a business objective.

business rules  Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

cache  A buffer in memory that holds data temporarily.

calc script  A set of commands that define how a database is consolidated or aggregated. A calculation script may also contain commands that specify allocation and other calculation rules separate from the consolidation process.

Calculated Accounts  You cannot alter the formulas in Calculated Accounts. These formulas are fixed in order to maintain the accounting integrity of the model you are building. For example, the formula for Net Income, a Calculated Account, is modeled into Strategic Finance and can not be changed in either historical or forecast periods.

calculated member in MaxL DML  A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

calculation  The process of aggregating data, or of running a calculation script on a database.

calculation status  A consolidation status that indicates that some values or formula calculations have changed. You must reconsolidate to get the correct values for the affected entity.

calendar  User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

cascade  The process of creating multiple reports for a subset of member values.

Catalog pane  Displays a list of elements available to the active section. If Query is the active section, a list of database tables is displayed. If Pivot is the active section, a list of results columns is displayed. If Dashboard is the active section, a list of embeddable sections, graphic tools, and control tools are displayed.

categories  Groupings by which data is organized. For example, Month

cause and effect map  Depicts how the elements that form your corporate strategy relate and how they work together to meet your organization’s strategic goals. A Cause and Effect map tab is automatically created for each Strategy map.

CDF  See custom-defined function (CDF).

CDM  See custom-defined macro (CDM).

cell  (1) The data value at the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet. (2) A logical group of nodes belonging to one administrative domain.

cell note  A text annotation for a cell in an Essbase database. Cell notes are a type of LRO.

CHANGED status  Consolidation status that indicates data for an entity has changed.

chart  A graphical representation of spreadsheet data. The visual nature expedites analysis, color-coding, and visual cues that aid comparisons.

chart template  A template that defines the metrics to display in Workspace charts.

child  A member with a parent above it in the database outline.
**choice list** A list of members that a report designer can specify for each dimension when defining the report’s point of view. A user who wants to change the point of view for a dimension that uses a choice list can select only the members specified in that defined member list or those members that meet the criteria defined in the function for the dynamic list.

**clean block** A data block that where the database is fully calculated, if a calculation script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calculation script.

**cluster** An array of servers or databases that behave as a single resource which share task loads and provide failover support; eliminates one server or database as a single point of failure in a system.

**clustered bar charts** Charts in which categories are viewed side-by-side; useful for side-by-side category analysis; used only with vertical bar charts.

**code page** A mapping of bit combinations to a set of text characters. Different code pages support different sets of characters. Each computer contains a code page setting for the character set requirements of the language of the computer user. In the context of this document, code pages map characters to bit combinations for non-Unicode encodings. See also **encoding**.

**column** A vertical display of information in a grid or table. A column can contain data from one field, derived data from a calculation, or textual information.

**committed access** An Essbase Kernel Isolation Level setting that affects how Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

**computed item** A virtual column (as opposed to a column that is physically stored in the database or cube) that can be calculated by the database during a query, or by Interactive Reporting Studio in the Results section. Computed items are calculations of data based on functions, data items, and operators provided in the dialog box and can be included in reports or reused to calculate other data.

**configuration file** The security platform relies on XML documents to be configured by the product administrator or software installer. The XML document must be modified to indicate meaningful values for properties, specifying locations and attributes pertaining to the corporate authentication scenario.

**connection file** See **Interactive Reporting connection file (.oce)**.

**consolidated file (Parent)** A file into which all of the business unit files are consolidated; contains the definition of the consolidation.

**consolidation** The process of aggregating data from dependent entities to parent entities. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year.

**consolidation file (*.cns)** The consolidation file is a graphical interface that enables you to add, delete or move Strategic Finance files in the consolidation process using either a Chart or Tree view. It also enables you to define and modify the consolidation.

**consolidation rule** Identifies the rule that is executed during the consolidation of the node of the hierarchy. This rule can contain customer specific formulas appropriate for the correct consolidation of parent balances. Elimination processing can be controlled within these rules.

**content** Information stored in the repository for any type of file.

**context variable** A variable that is defined for a particular task flow to identify the context of the taskflow instance.

**contribution** The value added to a parent from a child entity. Each child has a contribution to its parent.

**conversion rate** See **exchange rate**.

**cookie** A segment of data placed on your computer by a Web site.

**correlated subqueries** Subqueries that are evaluated once for every row in the parent query; created by joining a topic item in the subquery with a topic in the parent query.
Cost of Debt Value determined by using a weighted average Yield to Maturity (YTM) of a company's entire debt portfolio. Use is the current YTM rate rather than the nominal cost of debt. The coupon rate determines the interest payment, but it does not always reflect the actual cost of the company's debt today. As required returns change, the price of a debt issue also changes so that the actual interest payments and anticipated proceeds, at maturity, yield the investors their revised required return. Therefore, the YTM fully reflects the current return demanded by debt holders and the rate at which existing debt would have to be replaced.

Cost of Equity The return an investor expects to earn on an individual stock. Using the CAPM method, the Cost of Equity is equal to:

Cost of Preferred Represents the expected return to preferred stockholders. Like debt, you need to enter the yield to maturity on preferred stock, but without the tax shielding.

critical business area (CBA) An individual or a group organized into a division, region, plant, cost center, profit center, project team, or process; also called accountability team or business area.

critical success factor (CSF) A capability that must be established and sustained to achieve a strategic objective; owned by a strategic objective or a critical process and is a parent to one or more actions.

crosstab reporting Categorizes and summarizes data in table format. The table cells contain summaries of the data that fit within the intersecting categories. For example, a crosstab report of product sales information could show size attributes, such as Small and Large, as column headings and color attributes, such as Blue and Yellow, as row headings. The cell in the table where Large and Blue intersect could contain the total sales of all Blue products that are sized Large.

cube A block of data that contains three or more dimensions. An Essbase database is a cube.

currency conversion A process that converts currency values in a database from one currency into another. For example, to convert one U.S. dollar into the European euro, the exchange rate (for example, 0.923702) is multiplied with the dollar (1 * 0.923702). After conversion, the European euro amount is .92.

Currency Overrides In any input period, the selected input method can be overridden to enable input of that period's value as Default Currency/Items. To override the input method, enter a pound sign (#) either before or after the number.

currency partition A dimension type that separates local currency members from a base currency, as defined in an application. Identifies currency types, such as Actual, Budget, and Forecast.

custom calendar Any calendar created by an administrator.

custom dimension A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

custom property A property of a dimension or dimension member that is created by a user.

custom report A complex report from the Design Report module, composed of any combination of components.

custom-defined function (CDF) Essbase calculation functions developed in Java and added to the standard Essbase calculation scripting language using MaxL. See also custom-defined macro (CDM).

custom-defined macro (CDM) Essbase macros written with Essbase calculator functions and special macro functions. Custom-defined macros use an internal Essbase macro language that enables the combination of calculation functions and they operate on multiple input parameters. See also custom-defined function (CDF).

cycle through To perform multiple passes through a database while calculating it.

dashboard A collection of metrics and indicators that provide an interactive summary of your business. Dashboards enable you to build and deploy analytic applications.

data cache A buffer in memory that holds uncompressed data blocks.

data cell See cell.

data file cache A buffer in memory that holds compressed data (PAG) files.
**data form** A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

**data function** That computes aggregate values, including averages, maximums, counts, and other statistics, that summarize groupings of data.

**data load rules** A set of criteria that determines how to load data from a text-based file, a spreadsheet, or a relational data set into a database.

**data lock** Prevents changes to data according to specified criteria, such as period or scenario.

**data mining** The process of searching through an Essbase database for hidden relationships and patterns in a large amount of data.

**data model** A representation of a subset of database tables.

**data value** See *cell*.

**database connection** File that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.

**Default Currency Units** Define the unit scale of data. For example, if you select to define your analysis in Thousands, and enter “10”, this is interpreted as “10,000”.

**dense dimension** In block storage databases, a dimension likely to contain data for every combination of dimension members. For example, time dimensions are often dense because they can contain all combinations of all members. Contrast with sparse dimension.

**dependent entity** An entity that is owned by another entity in the organization.

**descendant** Any member below a parent in the database outline. In a dimension that includes years, quarters, and months, the members Qtr2 and April are descendants of the member Year.

**Design Report** An interface in Web Analysis Studio for designing custom reports, from a library of components.

**destination currency** The currency to which balances are converted. You enter exchange rates and convert from the source currency to the destination currency. For example, when you convert from EUR to USD, the destination currency is USD.

**detail chart** A chart that provides the detailed information that you see in a Summary chart. Detail charts appear in the Investigate Section in columns below the Summary charts. If the Summary chart shows a Pie chart, then the Detail charts below represent each piece of the pie.

**dimension** A data category used to organize business data for retrieval and preservation of values. Dimensions usually contain hierarchies of related members grouped within them. For example, a Year dimension often includes members for each time period, such as quarters and months.

**dimension build** The process of adding dimensions and members to an Essbase outline.

**dimension build rules** Specifications, similar to data load rules, that Essbase uses to modify an outline. The modification is based on data in an external data source file.

**dimension tab** In the Pivot section, the tab that enables you to pivot data between rows and columns.

**dimension table** (1) A table that includes numerous attributes about a specific business process. (2) In Essbase Integration Services, a container in the OLAP model for one or more relational tables that define a potential dimension in Essbase.

**dimension type** A dimension property that enables the use of predefined functionality. Dimensions tagged as time have a predefined calendar functionality.

**dimensionality** In MaxL DML, the represented dimensions (and the order in which they are represented) in a set. For example, the following set consists of two tuples of the same dimensionality because they both reflect the dimensions (Region, Year): { (West, Feb), (East, Mar) }

**direct rate** A currency rate that you enter in the exchange rate table. The direct rate is used for currency conversion. For example, to convert balances from JPY to USD, in the exchange rate table, enter a rate for the period/scenario where the source currency is JPY and the destination currency is USD.
dirty block A data block containing cells that have been changed since the last calculation. Upper level blocks are marked as dirty if their child blocks are dirty (that is, they have been updated).

display type One of three Web Analysis formats saved to the repository: spreadsheet, chart, and pinboard.

dog-ear The flipped page corner in the upper right corner of the chart header area.

domain In data mining, a variable representing a range of navigation within data.

drill-down Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

drill-through The navigation from a value in one data source to corresponding data in another source.

duplicate alias name A name that occurs more than once in an alias table and that can be associated with more than one member in a database outline. Duplicate alias names can be used with duplicate member outlines only.

duplicate member name The multiple occurrence of a member name in a database, with each occurrence representing a different member. For example, a database has two members named “New York.” One member represents New York state and the other member represents New York city.

duplicate member outline A database outline containing duplicate member names.

Dynamic Calc and Store members A member in a block storage outline that Essbase calculates only upon the first retrieval of the value. Essbase then stores the calculated value in the database. Subsequent retrievals do not require calculating.

Dynamic Calc members A member in a block storage outline that Essbase calculates only at retrieval time. Essbase discards calculated values after completing the retrieval request.

dynamic calculation In Essbase, a calculation that occurs only when you retrieve data on a member that is tagged as Dynamic Calc or Dynamic Calc and Store. The member’s values are calculated at retrieval time instead of being precalculated during batch calculation.

dynamic hierarchy In aggregate storage database outlines only, a hierarchy in which members are calculated at retrieval time.

dynamic member list A system-created named member set that is based on user-defined criteria. The list is refreshed automatically whenever it is referenced in the application. As dimension members are added and deleted, the list automatically reapplies the criteria to reflect the changes.

dynamic reference A pointer in the rules file to header records in a data source.

dynamic report A report containing data that is updated when you run the report.

Dynamic Time Series A process that performs period-to-date reporting in block storage databases.

dynamic view account An account type indicating that account values are calculated dynamically from the data that is displayed.

Eliminated Account An account that does not appear in the consolidated file.

elimination The process of zeroing out (eliminating) transactions between entities within an organization.

employee A user responsible for, or associated with, specific business objects. Employees need not work for an organization; for example, they can be consultants. Employees must be associated with user accounts for authorization purposes.

encoding A method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. See also code page and locale.

ending period A period enabling you to adjust the date range in a chart. For example, an ending period of “month”, produces a chart showing information through the end of the current month.
Enterprise View  An Administration Services feature that enables management of the Essbase environment from a graphical tree view. From Enterprise View, you can operate directly on Essbase artifacts.

dimension A dimension representing organizational units. Examples: divisions, subsidiaries, plants, regions, products, or other financial reporting units.

Equity Beta  The riskiness of a stock, measured by the variance between its return and the market return, indicated by an index called “beta”. For example, if a stock’s return normally moves up or down 1.2% when the market moves up or down 1%, the stock has a beta of 1.2.

essbase.cfg  An optional configuration file for Essbase. Administrators may edit this file to customize Essbase Server functionality. Some configuration settings may also be used with Essbase clients to override Essbase Server settings.

EssCell  A function entered into an Essbase Spreadsheet Add-in to retrieve a value representing an intersection of specific Essbase database members.

ESSCMD  A command-line interface for performing Essbase operations interactively or through batch script files.

ESSLANG  The Essbase environment variable that defines the encoding used to interpret text characters. See also encoding.

ESSMSH  See MaxL Shell.

exceptions  Values that satisfy predefined conditions. You can define formatting indicators or notify subscribing users when exceptions are generated.

exchange rate  A numeric value for converting one currency to another. For example, to convert 1 USD into EUR, the exchange rate of 0.8936 is multiplied with the U.S. dollar. The European euro equivalent of $1 is 0.8936.

exchange rate type  An identifier for an exchange rate. Different rate types are used because there may be multiple rates for a period and year. Users traditionally define rates at period end for the average rate of the period and for the end of the period. Additional rate types are historical rates, budget rates, forecast rates, and so on. A rate type applies to one point in time.

expense account  An account that stores periodic and year-to-date values that decrease net worth if they are positive.

Extensible Markup Language (XML)  A language comprising a set of tags used to assign attributes to data that can be interpreted between applications according to a schema.

external authentication  Logging on to Oracle’s Hyperion applications with user information stored outside the applications, typically in a corporate directory such as MSAD or NTLM.

externally triggered events  Non-time-based events for scheduling job runs.

Extract, Transform, and Load (ETL)  Data source-specific programs for extracting data and migrating it to applications.

extraction command  An Essbase reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database; begins with the less than (<) character.

fact table  The central table in a star join schema, characterized by a foreign key and elements drawn from a dimension table. This table typically contains numeric data that can be related to all other tables in the schema.

field  An item in a data source file to be loaded into an Essbase database.

file delimiter  Characters, such as commas or tabs, that separate fields in a data source.

filter  A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

flow account  An unsigned account that stores periodic and year-to-date values.

folder  A file containing other files for the purpose of structuring a hierarchy.

footer  Text or images at the bottom of report pages, containing dynamic functions or static text such as page numbers, dates, logos, titles or file names, and author names.

format  Visual characteristics of documents or report objects.
formula A combination of operators, functions, dimension and member names, and numeric constants calculating database members.

frame An area on the desktop. There are two main areas: the navigation and workspace frames.

free-form grid An object for presenting, entering, and integrating data from different sources for dynamic calculations.

free-form reporting Creating reports by entering dimension members or report script commands in worksheets.

function A routine that returns values or database members.

generation A layer in a hierarchical tree structure that defines member relationships in a database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members.

generation name A unique name that describes a generation.

generic jobs Non-SQR Production Reporting or non-Interactive Reporting jobs.

global report command A command in a running report script that is effective until replaced by another global command or the file ends.

grid POV A means for specifying dimension members on a grid without placing dimensions in rows, columns, or page intersections. A report designer can set POV values at the grid level, preventing user POVs from affecting the grid. If a dimension has one grid value, you put the dimension into the grid POV instead of the row, column, or page.

group A container for assigning similar access permissions to multiple users.

GUI Graphical user interface

highlighting Depending on your configuration, chart cells or ZoomChart details may be highlighted, indicating value status: red (bad), yellow (warning), or green (good).

Historical Average An average for an account over a number of historical periods.

holding company An entity that is part of a legal entity group, with direct or indirect investments in all entities in the group.

host A server on which applications and services are installed.

host properties Properties pertaining to a host, or if the host has multiple Install_Homes, to an Install_Home. The host properties are configured from the LSC.

Hybrid Analysis An analysis mapping low-level data stored in a relational database to summary-level data stored in Essbase, combining the mass scalability of relational systems with multidimensional data.

hyperlink A link to a file, Web page, or an intranet HTML page.

Hypertext Markup Language (HTML) A programming language specifying how Web browsers display data.

identity A unique identification for a user or group in external authentication.

image bookmarks Graphic links to Web pages or repository items.

IMPACTED status Indicates changes in child entities consolidating into parent entities.

implied share A member with one or more children, but only one is consolidated, so the parent and child share a value.

inactive group A group for which an administrator has deactivated system access.

inactive service A service suspended from operating.

INACTIVE status Indicates entities deactivated from consolidation for the current period.

inactive user A user whose account has been deactivated by an administrator.

income account An account storing periodic and year-to-date values that, if positive, increase net worth.


index cache A buffer containing index pages.
index entry  A pointer to an intersection of sparse dimensions. Index entries point to data blocks on disk and use offsets to locate cells.

index file  An Essbase file storing block storage data retrieval information, residing on disk, and containing index pages.

index page  A subdivision in an index file. Contains pointers to data blocks.

input data  Data loaded from a source rather than calculated.

Install_Home  A variable for the directory where Oracle’s Hyperion applications are installed. Refers to one instance of Oracle’s Hyperion application when multiple applications are installed on the same computer.

integration  Process that is run to move data between Oracle’s Hyperion applications using Shared Services. Data integration definitions specify the data moving between a source application and a destination application, and enable the data movements to be grouped, ordered, and scheduled.

intelligent calculation  A calculation method tracking updated data blocks since the last calculation.

Interactive Reporting connection file (.oce)  Files encapsulating database connection information, including: the database API (ODBC, SQL*Net, etc.), database software, the database server network address, and database user name. Administrators create and publish Interactive Reporting connection files (.oce).

intercompany elimination  See elimination.

intercompany matching  The process of comparing balances for pairs of intercompany accounts within an application. Intercompany receivables are compared to intercompany payables for matches. Matching accounts are used to eliminate intercompany transactions from an organization’s consolidated totals.

intercompany matching report  A report that compares intercompany account balances and indicates if the accounts are in, or out, of balance.

interdimensional irrelevance  A situation in which a dimension does not intersect with other dimensions. Because the data in the dimension cannot be accessed from the non-intersecting dimensions, the non-intersecting dimensions are not relevant to that dimension.

intersection  A unit of data representing the intersection of dimensions in a multidimensional database; also, a worksheet cell.

Investigation  See drill-through.

isolation level  An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

iteration  A “pass” of the budget or planning cycle in which the same version of data is revised and promoted.

Java Database Connectivity (JDBC)  A client-server communication protocol used by Java based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

job output  Files or reports produced from running a job.

job parameters  Reusable, named job parameters that are accessible only to the user who created them.

jobs  Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

join  A link between two relational database tables or topics based on common content in a column or row. A join typically occurs between identical or similar items within different tables or topics. For example, a record in the Customer table is joined to a record in the Orders table because the Customer ID value is the same in each table.

journal entry (JE)  A set of debit/credit adjustments to account balances for a scenario and period.

JSP  Java Server Pages.

latest  A Spreadsheet key word used to extract data values from the member defined as the latest time period.

layer (1)  The horizontal location of members in a hierarchical structure, specified by generation (top down) or level (bottom up). (2) Position of objects relative to other objects. For example, in the Sample Basic database, Qtr1 and Qtr4 are in the same layer, so they are also in the same generation, but in a database with a ragged hierarchy, Qtr1 and Qtr4 might not be in same layer, though they are in the same generation.
**legend box** A box containing labels that identify the data categories of a dimension.

**level** A layer in a hierarchical tree structure that defines database member relationships. Levels are ordered from the bottom dimension member (level 0) up to the parent members.

**level 0 block** A data block for combinations of sparse, level 0 members.

**level 0 member** A member that has no children.

**liability account** An account type that stores “point in time” balances of a company’s liabilities. Examples of liability accounts include accrued expenses, accounts payable, and long term debt.

**life cycle management** The process of managing application information from inception to retirement.

**line chart** A chart that displays one to 50 data sets, each represented by a line. A line chart can display each line stacked on the preceding ones, as represented by an absolute value or a percent.

**line item detail** The lowest level of detail in an account.

**link** (1) A reference to a repository object. Links can reference folders, files, shortcuts, and other links. (2) In a task flow, the point where the activity in one stage ends and another begins.

**link condition** A logical expression evaluated by the taskflow engine to determine the sequence of launching taskflow stages.

**linked data model** Documents that are linked to a master copy in a repository.

**linked partition** A shared partition that enables you to use a data cell to link two databases. When a user clicks a linked cell in a worksheet, Essbase opens a new sheet displaying the dimensions in the linked database. The user can then drill down those dimensions.

**linked reporting object (LRO)** A cell-based link to an external file such as cell notes, URLs, or files with text, audio, video, or pictures. (Only cell notes are supported for Essbase LROs in Financial Reporting.)

**local currency** An input currency type. When an input currency type is not specified, the local currency matches the entity’s base currency.

**local report object** A report object that is not linked to a Financial Reporting report object in Explorer. Contrast with **linked reporting object (LRO)**.

**local results** A data model’s query results. Results can be used in local joins by dragging them into the data model. Local results are displayed in the catalog when requested.

**locale** A computer setting that specifies a location’s language, currency and date formatting, data sort order, and the character set encoding used on the computer. Essbase uses only the encoding portion. See also **encoding** and **ESSLANG**.

**locale header record** A text record at the beginning of some non-Unicode-encoded text files, such as scripts, that identifies the encoding locale.

**location alias** A descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by DBAs at the database level using Administration Services Console, ESSCMD, or the API.

**locked** A user-invoked process that prevents users and processes from modifying data.

**locked data model** Data models that cannot be modified by a user.

**LOCKED status** A consolidation status indicating that an entity contains data that cannot be modified.

**Log Analyzer** An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

**LRO** See **linked reporting object (LRO)**.

**LSC services** Services configured with the Local Service Configurator. They include Global Services Manager (GSM), Local Services Manager (LSM), Session Manager, Authentication Service, Authorization Service, Publisher Service, and sometimes, Data Access Service (DAS) and Interactive Reporting Service.

**managed server** An application server process running in its own Java Virtual Machine (JVM).
**manual stage** A stage that requires human intervention to complete.

**Map File** Used to store the definition for sending data to or retrieving data from an external database. Map files have different extensions (.mps to send data; .mpr to retrieve data).

**Map Navigator** A feature that displays your current position on a Strategy, Accountability, or Cause and Effect map, indicated by a red outline.

**Marginal Tax Rate** Used to calculate the after-tax cost of debt. Represents the tax rate applied to the last earned income dollar (the rate from the highest tax bracket into which income falls) and includes federal, state and local taxes. Based on current level of taxable income and tax bracket, you can predict marginal tax rate.

**Market Risk Premium** The additional rate of return paid over the risk-free rate to persuade investors to hold “riskier” investments than government securities. Calculated by subtracting the risk-free rate from the expected market return. These figures should closely model future market conditions.

**master data model** An independent data model that is referenced as a source by multiple queries. When used, “Locked Data Model” is displayed in the Query section’s Content pane; the data model is linked to the master data model displayed in the Data Model section, which an administrator may hide.

**mathematical operator** A symbol that defines how data is calculated in formulas and outlines. Can be any of the standard mathematical or Boolean operators; for example, +, -, *, /, and %.

**MaxL** The multidimensional database access language for Essbase, consisting of a data definition language (MaxL DDL) and a data manipulation language (MaxL DML). See also MaxL DDL, MaxL DML, and MaxL Shell.

**MaxL DDL** Data definition language used by Essbase for batch or interactive system-administration tasks.

**MaxL DML** Data manipulation language used in Essbase for data query and extraction.

**MaxL Perl Module** A Perl module (essbase.pm) that is part of Essbase MaxL DDL. This module can be added to the Perl package to provide access to Essbase databases from Perl programs.

**MaxL Script Editor** A script-development environment in Administration Services Console. MaxL Script Editor is an alternative to using a text editor and the MaxL Shell for administering Essbase with MaxL scripts.

**MaxL Shell** An interface for passing MaxL statements to Essbase Server. The MaxL Shell executable file is located in the Essbase bin directory (UNIX: essmsh, Windows: essmsh.exe).

**MDX (multidimensional expression)** The language that give instructions to OLE DB for OLAP- compliant databases, as SQL is used for relational databases. When you build the OLAPQuery section’s Outliner, Interactive Reporting Clients translate requests into MDX instructions. When you process the query, MDX is sent to the database server, which returns records that answer your query. See also SQL spreadsheet.

**measures** Numeric values in an OLAP database cube that are available for analysis. Measures are margin, cost of goods sold, unit sales, budget amount, and so on. See also fact table.

**member** A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include such members as Jan, Feb, and Qtr1.

**member list** A named group, system- or user-defined, that references members, functions, or member lists within a dimension.

**member load** In Essbase Integration Services, the process of adding dimensions and members (without data) to Essbase outlines.

**member selection report command** A type of Report Writer command that selects member ranges based on outline relationships, such as sibling, generation, and level.

**member-specific report command** A type of Report Writer formatting command that is executed as it is encountered in a report script. The command affects only its associated member and executes the format command before processing the member.
merge A data load option that clears values only from the accounts specified in the data load file and replaces them with values in the data load file.

metadata A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. Examples of metadata are dimension names, member names, properties, time periods, and security.

metadata sampling The process of retrieving a sample of members in a dimension in a drill-down operation.

metadata security Security set at the member level to restrict users from accessing certain outline members.

metaoutline In Essbase Integration Services, a template containing the structure and rules for creating an Essbase outline from an OLAP model.

metric A numeric measurement computed from business data to help assess business performance and analyze company trends.

migration audit report A report generated from the migration log that provides tracking information for an application migration.

migration definition file (.mdf) A file that contains migration parameters for an application migration, enabling batch script processing.

migration log A log file that captures all application migration actions and messages.

migration snapshot A snapshot of an application migration that is captured in the migration log.

MIME Type (Multipurpose Internet Mail Extension) An attribute that describes the data format of an item, so that the system knows which application should open the object. A file's mime type is determined by the file extension or HTTP header. Plug-ins tell browsers what mime types they support and what file extensions correspond to each mime type.

mining attribute In data mining, a class of values used as a factor in analysis of a set of data.

minireport A report component that includes layout, content, hyperlinks, and the query or queries to load the report. Each report can include one or more minireports.

missing data (#MISSING) A marker indicating that data in the labeled location does not exist, contains no value, or was never entered or loaded. For example, missing data exists when an account contains data for a previous or future period but not for the current period.

model (1) In data mining, a collection of an algorithm’s findings about examined data. A model can be applied against a wider data set to generate useful information about that data. (2) A file or content string containing an application-specific representation of data. Models are the basic data managed by Shared Services, of two major types: dimensional and non-dimensional application objects. (3) In Business Modeling, a network of boxes connected to represent and calculate the operational and financial flow through the area being examined.

monetary A money-related value.

multidimensional database A method of organizing, storing, and referencing data through three or more dimensions. An individual value is the intersection point for a set of dimensions.

named set In MaxL DML, a set with its logic defined in the optional WITH section of a MaxL DML query. The named set can be referenced multiple times in the query.

native authentication The process of authenticating a user name and password from within the server or application.

nested column headings A report column heading format that displays data from multiple dimensions. For example, a column heading that contains Year and Scenario members is a nested column. The nested column heading shows Q1 (from the Year dimension) in the top line of the heading, qualified by Actual and Budget (from the Scenario dimension) in the bottom line of the heading.

NO DATA status A consolidation status indicating that this entity contains no data for the specified period and account.

non-dimensional model A Shared Services model type that includes application objects such as security files, member lists, calculation scripts, and Web forms.

non-unique member name See duplicate member name.

note Additional information associated with a box, measure, scorecard or map element.
null value  A value that is absent of data. Null values are not equal to zero.

numeric attribute range  A feature used to associate a base dimension member that has a discrete numeric value with an attribute that represents a value range. For example, to classify customers by age, an Age Group attribute dimension can contain members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. Each Customer dimension member can be associated with an Age Group range. Data can be retrieved based on the age ranges rather than on individual age values.

ODBC  Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

OK status  A consolidation status indicating that an entity has already been consolidated, and that data has not changed below it in the organization structure.

OLAP Metadata Catalog  In Essbase Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

OLAP model  In Essbase Integration Services, a logical model (star schema) that is created from tables and columns in a relational database. The OLAP model is then used to generate the structure of a multidimensional database.

online analytical processing (OLAP)  A multidimensional, multiuser, client-server computing environment for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, and modeling.

Open Database Connectivity (ODBC)  Standardized application programming interface (API) technology that allows applications to access multiple third-party databases.

organization  An entity hierarchy that defines each entity and their relationship to others in the hierarchy.

origin  The intersection of two axes.

outline  The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

outline synchronization  For partitioned databases, the process of propagating outline changes from one database to another database.

P&L accounts (P&L)  Profit and loss accounts. Refers to a typical grouping of expense and income accounts that comprise a company’s income statement.

page  A display of information in a grid or table often represented by the Z-axis. A page can contain data from one field, derived data from a calculation, or text.

page file  Essbase data file.

page heading  A report heading type that lists members represented on the current page of the report. All data values on the page have the members in the page heading as a common attribute.

page member  A member that determines the page axis.

palette  A JASC compliant file with a .PAL extension. Each palette contains 16 colors that complement each other and can be used to set the dashboard color elements.

parallel calculation  A calculation option. Essbase divides a calculation into tasks and calculates some tasks simultaneously.

parallel data load  In Essbase, the concurrent execution of data load stages by multiple process threads.

parallel export  The ability to export Essbase data to multiple files. This may be faster than exporting to a single file, and it may resolve problems caused by a single data file becoming too large for the operating system to handle.

parent adjustments  The journal entries that are posted to a child in relation to its parent.

parents  The entities that contain one or more dependent entities that report directly to them. Because parents are both entities and associated with at least one node, they have entity, node, and parent information associated with them.

partition area  A subcube within a database. A partition is composed of one or more areas of cells from a portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for the data source and target to ensure that the two partitions have the same shape. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values.
partitioning  The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Essbase applications.

pattern matching  The ability to match a value with any or all characters of an item entered as a criterion. Missing characters may be represented by wild card values such as a question mark (?) or an asterisk (*). For example, “Find all instances of apple” returns apple, but “Find all instances of apple*” returns apple, applesauce, applecranberry, and so on.

percent consolidation  The portion of a child’s values that is consolidated to its parent.

percent control  Identifies the extent to which an entity is controlled within the context of its group.

percent ownership  Identifies the extent to which an entity is owned by its parent.

performance indicator  An image file used to represent measure and scorecard performance based on a range you specify; also called a status symbol. You can use the default performance indicators or create an unlimited number of your own.

periodic value method (PVA)  A process of currency conversion that applies the periodic exchange rate values over time to derive converted results.

permission  A level of access granted to users and groups for managing data or other users and groups.

persistance  The continuance or longevity of effect for any Essbase operation or setting. For example, an Essbase administrator may limit the persistence of user name and password validity.

personal pages  A personal window to repository information. You select what information to display and its layout and colors.

personal recurring time events  Reusable time events that are accessible only to the user who created them.

personal variable  A named selection statement of complex member selections.

perspective  A category used to group measures on a scorecard or strategic objectives within an application. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

pie chart  A chart that shows one data set segmented in a pie formation.

pinboard  One of the three data object display types. Pinboards are graphics, composed of backgrounds and interactive icons called pins. Pinboards require traffic lighting definitions.

pins  Interactive icons placed on graphic reports called pinboards. Pins are dynamic. They can change images and traffic lighting color based on the underlying data values and analysis tools criteria.

pivot  The ability to alter the perspective of retrieved data. When Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

planner  Planners, who comprise the majority of users, can input and submit data, use reports that others create, execute business rules, use task lists, enable e-mail notification for themselves, and use Smart View.

planning unit  A data slice at the intersection of a scenario, version, and entity; the basic unit for preparing, reviewing, annotating, and approving plan data.

plot area  The area bounded by X, Y, and Z axes; for pie charts, the rectangular area surrounding the pie.

plug account  An account in which the system stores any out of balance differences between intercompany account pairs during the elimination process.

POV (point of view)  A feature for working with dimension members not assigned to row, column, or page axes. For example, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in data forms displays data in Euro values.

precalculation  Calculating the database prior to user retrieval.

precision  Number of decimal places displayed in numbers.

predefined drill paths  Paths used to drill to the next level of detail, as defined in the data model.
**presentation** A playlist of Web Analysis documents, enabling reports to be grouped, organized, ordered, distributed, and reviewed. Includes pointers referencing reports in the repository.

**preserve formulas** User-created formulas kept within a worksheet while retrieving data.

**primary measure** A high-priority measure important to your company and business needs. Displayed in the Contents frame.

**product** In Shared Services, an application type, such as Planning or Performance Scorecard.

**Production Reporting** See SQR Production Reporting.

**project** An instance of Oracle’s Hyperion products grouped together in an implementation. For example, a Planning project may consist of a Planning application, an Essbase cube, and a Financial Reporting Server instance.

**promote** The action to move a data unit to the next review level, allowing a user having the appropriate access to review the data. For example, an analyst may promote the data unit to the next level for his supervisor’s review.

**promotion** The process of transferring artifacts from one environment or machine to another; for example, from a testing environment to a production environment.

**property** A characteristic of an artifact, such as size, type, or processing instructions.

**provisioning** The process of granting users and groups specific access permissions to resources.

**proxy server** A server acting as an intermediary between workstation users and the Internet to ensure security.

**public job parameters** Reusable, named job parameters created by administrators and accessible to users with requisite access privileges.

**public recurring time events** Reusable time events created by administrators and accessible through the access control system.

**PVA** See periodic value method (PVA).

**qualified name** A member name in a qualified format that differentiates duplicate member names in a duplicate member outline. For example, [Market].[East].[State]. [New York] or [Market].[East].[City].[New York]

**query** Information requests from data providers. For example, used to access relational data sources.

**query governor** An Essbase Integration Server parameter or Essbase Server configuration setting that controls the duration and size of queries made to data sources.

**range** A set of values including upper and lower limits, and values falling between limits. Can contain numbers, amounts, or dates.

**reconfigure URL** URL used to reload servlet configuration settings dynamically when users are already logged on to the Workspace.

**record** In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

**recurring template** A journal template for making identical adjustments in every period.

**recurring time event** An event specifying a starting point and the frequency for running a job.

**redundant data** Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

**regular journal** A feature for entering one-time adjustments for a period. Can be balanced, balanced by entity, or unbalanced.

**Related Accounts** The account structure groups all main and related accounts under the same main account number. The main account is distinguished from related accounts by the first suffix of the account number.

**relational database** A type of database that stores data in related two-dimensional tables. Contrast with multidimensional database.

**replace** A data load option that clears existing values from all accounts for periods specified in the data load file, and loads values from the data load file. If an account is not specified in the load file, its values for the specified periods are cleared.

**replicated partition** A portion of a database, defined through Partition Manager, used to propagate an update to data mastered at one site to a copy of data stored at another site. Users can access the data as though it were part of their local database.
**Report Extractor** An Essbase component that retrieves report data from the Essbase database when report scripts are run.

**report object** In report designs, a basic element with properties defining behavior or appearance, such as text boxes, grids, images, and charts.

**report script** A text file containing Essbase Report Writer commands that generate one or more production reports.

**Report Viewer** An Essbase component that displays complete reports after report scripts are run.

**reporting currency** The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

**repository** Stores metadata, formatting, and annotation information for views and queries.

**resources** Objects or services managed by the system, such as roles, users, groups, files, and jobs.

**restore** An operation to reload data and structural information after a database has been damaged or destroyed, typically performed after shutting down and restarting the database.

**restructure** An operation to regenerate or rebuild the database index and, in some cases, data files.

**result frequency** The algorithm used to create a set of dates to collect and display results.

**review level** A Process Management review status indicator representing the process unit level, such as Not Started, First Pass, Submitted, Approved, and Published.

**Risk Free Rate** The rate of return expected from “safer” investments such as long-term U.S. government securities.

**role** The means by which access permissions are granted to users and groups for resources.

**roll-up** See consolidation.

**root member** The highest member in a dimension branch.

**row heading** A report heading that lists members down a report page. The members are listed under their respective row names.

**RSC services** Services that are configured with Remote Service Configurator, including Repository Service, Service Broker, Name Service, Event Service, and Job Service.

**rules** User-defined formulas.

**runtime prompt** A variable that users enter or select before a business rule is run.

**sampling** The process of selecting a representative portion of an entity to determine the entity’s characteristics. See also metadata sampling.

**saved assumptions** User-defined Planning assumptions that drive key business calculations (for example, the cost per square foot of office floor space).

**scale** The range of values on the Y axis of a chart.

**scaling** Scaling determines the display of values in whole numbers, tens, hundreds, thousands, millions, and so on.

**scenario** A dimension for classifying data (for example, Actuals, Budget, Forecast1, and Forecast2).

**schedule** Specify the job that you want to run and the time and job parameter list for running the job.

**scope** The area of data encompassed by any Essbase operation or setting; for example, the area of data affected by a security setting. Most commonly, scope refers to three levels of granularity, where higher levels encompass lower levels. From highest to lowest, these levels are as follows: the entire system (Essbase Server), applications on Essbase Server, or databases within Essbase Server applications. See also persistence.

**score** The level at which targets are achieved, usually expressed as a percentage of the target.

**scorecard** Business Object that represents the progress of an employee, strategy element, or accountability element toward goals. Scorecards ascertain this progress based on data collected for each measure and child scorecard added to the scorecard.

**scorecard report** A report that presents the results and detailed information about scorecards attached to employees, strategy elements, and accountability elements.
secondary measure  A low-priority measure, less important than primary measures. Secondary measures do not have Performance reports but can be used on scorecards and to create dimension measure templates.

Section pane  Lists all sections that are available in the current Interactive Reporting Client document.

security agent  A Web access management provider (for example, Netegrity SiteMinder) that protects corporate Web resources.

security platform  A framework enabling Oracle’s Hyperion applications to use external authentication and single sign-on.

serial calculation  The default calculation setting Essbase divides a calculation pass into tasks and calculates one task at a time.

services  Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

servlet  A piece of compiled code executable by a Web server.

Servlet Configurator  A utility for configuring all locally installed servlets.

session  The time between login and logout for a user connected to Essbase Server.

set  In MaxL DML, a required syntax convention for referring to a collection of one or more tuples. For example, in the following MaxL DML query, SELECT { [100-10] } ON COLUMNS FROM Sample.Basic { [100-10] } is a set.

shared member  A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

Shared Services  Application enabling users to share data between supported Oracle’s Hyperion products by publishing data to Shared Services and running data integrations.

sibling  A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other’s siblings.

single sign-on  Ability to access multiple Oracle’s Hyperion products after a single login using external credentials.

slicer  In MaxL DML, the section at the end of a query that begins with and includes the keyword WHERE.

smart tags  Keywords in Microsoft Office applications that are associated with predefined actions available from the Smart Tag menu. In Oracle’s Hyperion applications, smart tags can also be used to import Reporting and Analysis content, and access Financial Management and Essbase functions.

SmartCut  A link to a repository item, in URL form.

snapshot  Read-only data from a specific time.

source currency  The currency from which values originate and are converted through exchange rates to the destination currency.

sparse dimension  In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. For example, not all customers have data for all products.

SPF files  Printer-independent files created by a SQR Production Reporting server, containing a representation of the actual formatted report output, including fonts, spacing, headers, footers, and so on.

Spotlighter  A tool that enables color coding based on selected conditions.

SQL spreadsheet  A data object that displays the result set of a SQL query.

SQR Production Reporting  A specialized programming language for data access, data manipulation, and creating SQR Production Reporting documents.

stacked charts  A chart where the categories are viewed on top of one another for visual comparison. This type of chart is useful for subcategorizing within the current category. Stacking can be used from the Y and Z axis in all chart types except pie and line. When stacking charts the Z axis is used as the Fact/Values axis.

stage  A task description that forms one logical step within a taskflow, usually performed by an individual. A stage can be manual or automated.
stage action  For automated stages, the invoked action that executes the stage.

standard dimension  A dimension that is not an attribute dimension.

standard journal template  A journal function used to post adjustments that have common adjustment information for each period. For example, you can create a standard template that contains the common account IDs, entity IDs, or amounts, then use the template as the basis for many regular journals.

Standard Template  The Standard template is the basis for the basic Strategic Finance file. The Standard template contains all default settings. All new files are created from the Standard template unless another template is selected.

Start in Play  The quickest method for creating a Web Analysis document. The Start in Play process requires you to specify a database connection, then assumes the use of a spreadsheet data object. Start in Play uses the highest aggregate members of the time and measures dimensions to automatically populate the rows and columns axes of the spreadsheet.

Status bar  The status bar at the bottom of the screen displays helpful information about commands, accounts, and the current status of your data file.

stored hierarchy  In aggregate storage databases outlines only. A hierarchy in which the members are aggregated according to the outline structure. Stored hierarchy members have certain restrictions, for example, they cannot contain formulas.

strategic objective (SO)  A long-term goal defined by measurable results. Each strategic objective is associated with one perspective in the application, has one parent, the entity, and is a parent to critical success factors or other strategic objectives.

Strategy map  Represents how the organization implements high-level mission and vision statements into lower-level, constituent strategic goals and objectives.

structure view  Displays a topic as a simple list of component data items.

Structured Query Language  A language used to process instructions to relational databases.

Subaccount Numbering  A system for numbering subaccounts using non-sequential, whole numbers.

subscribe  Flags an item or folder to receive automatic notification whenever the item or folder is updated.

Summary chart  In the Investigates Section, rolls up detail charts shown below in the same column, plotting metrics at the summary level at the top of each chart column.

super service  A special service used by the startCommonServices script to start the RSC services.

supervisor  A user with full access to all applications, databases, related files, and security mechanisms for a server.

supporting detail  Calculations and assumptions from which the values of cells are derived.

suppress rows  Excludes rows containing missing values, and underscores characters from spreadsheet reports.

symmetric multiprocessing (SMP)  A server architecture that enables multiprocessing and multithreading. Performance is not significantly degraded when a large number of users connect to an single instance simultaneously.

sync  Synchronizes Shared Services and application models.

synchronized  The condition that exists when the latest version of a model resides in both the application and in Shared Services. See also model.

system extract  Transfers data from an application’s metadata into an ASCII file.

tabs  Navigable views of accounts and reports in Strategic Finance.

target  Expected results of a measure for a specified period of time (day, quarter, etc.,)

task list  A detailed status list of tasks for a particular user.

taskflow  The automation of a business process in which tasks are passed from one taskflow participant to another according to procedural rules.
**taskflow definition**  Represents business processes in the taskflow management system. Consists of a network of stages and their relationships; criteria indicating the start and end of the taskflow; and information about individual stages, such as participants, associated applications, associated activities, and so on.

**taskflow instance**  Represents a single instance of a taskflow including its state and associated data.

**taskflow management system**  Defines, creates, and manages the execution of a taskflow including: definitions, user or application interactions, and application executables.

**taskflow participant**  The resource who performs the task associated with the taskflow stage instance for both manual and automated stages.

**Taxes - Initial Balances**  Strategic Finance assumes that the Initial Loss Balance, Initial Gain Balance and the Initial Balance of Taxes Paid entries have taken place in the period before the first Strategic Finance time period.


**template**  A predefined format designed to retrieve particular data consistently.

**time dimension**  Defines the time period that the data represents, such as fiscal or calendar periods.

**time events**  Triggers for execution of jobs.

**time scale**  Displays metrics by a specific period in time, such as monthly or quarterly.

**time series reporting**  A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

**Title bar**  Displays the Strategic Finance name, the file name, and the scenario name Version box.

**token**  An encrypted identification of one valid user or group on an external authentication system.

**top and side labels**  Column and row headings on the top and sides of a Pivot report.

**top-level member**  A dimension member at the top of the tree in a dimension outline hierarchy, or the first member of the dimension in sort order if there is no hierarchical relationship among dimension members. The top-level member name is generally the same as the dimension name if a hierarchical relationship exists.

**trace level**  Defines the level of detail captured in the log file.

**traffic lighting**  Color-coding of report cells, or pins based on a comparison of two dimension members, or on fixed limits.

**transformation**  (1) Transforms artifacts so that they function properly in the destination environment after application migration. (2) In data mining, modifies data (bidirectionally) flowing between the cells in the cube and the algorithm.

**translation**  See currency conversion.

**Transmission Control Protocol/Internet Protocol (TCP/IP)**  A standard set of communication protocols linking computers with different operating systems and internal architectures. TCP/IP utilities are used to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

**transparent login**  Logs in authenticated users without launching the login screen.

**transparent partition**  A shared partition that enables users to access and change data in a remote database as though it is part of a local database.

**triangulation**  A means of converting balances from one currency to another via a third common currency. In Europe, this is the euro for member countries. For example, to convert from French franc to Italian lira, the common currency is defined as European euro. Therefore, in order to convert balances from French franc to Italian lira, balances are converted from French franc to European euro and from European euro to Italian lira.

**triggers**  An Essbase feature whereby data is monitored according to user-specified criteria which when met cause Essbase to alert the user or system administrator.

**trusted password**  A password that enables users authenticated for one product to access other products without reentering their passwords.
trusted user  Authenticated user

tuple  MDX syntax element that references a cell as an intersection of a member from each dimension. If a dimension is omitted, its top member is implied. Examples: (Jan); (Jan, Sales); ([Jan], [Sales], [Cola], [Texas], [Actual])

two-pass  An Essbase property that is used to recalculate members that are dependent on the calculated values of other members. Two-pass members are calculated during a second pass through the outline.

unary operator  A mathematical indicator (+, -, *, /, %) associated with an outline member. The unary operator defines how the member is calculated during a database roll-up.

Unicode-mode application  An Essbase application wherein character text is encoded in UTF-8, enabling users with computers set up for different languages to share application data.

unique member name  A non-shared member name that exists only once in a database outline.

unique member outline  A database outline that is not enabled for duplicate member names.

upper-level block  A type of data block wherein at least one of the sparse members is a parent-level member.

user directory  A centralized location for user and group information. Also known as a repository or provider.

user variable  Dynamically renders data forms based on a user’s member selection, displaying only the specified entity. For example, user variable named Department displays specific departments and employees.

user-defined attribute (UDA)  User-defined attribute, associated with members of an outline to describe a characteristic of the members. Users can use UDAs to return lists of members that have the specified UDA associated with them.

user-defined member list  A named, static set of members within a dimension defined by the user.

validation  A process of checking a business rule, report script, or partition definition against the outline to make sure that the object being checked is valid.

value dimension  Used to define input value, translated value, and consolidation detail.

variance  Difference between two values (for example, planned and actual value).

version  Possible outcome used within the context of a scenario of data. For example, Budget - Best Case and Budget - Worst Case where Budget is scenario and Best Case and Worst Case are versions.

view  Representation of either a year-to-date or periodic display of data.

visual cue  A formatted style, such as a font or a color, that highlights specific types of data values. Data values may be dimension members; parent, child, or shared members; dynamic calculations; members containing a formula; read only data cells; read and write data cells; or linked objects.

Web server  Software or hardware hosting intranet or Internet Web pages or Web applications.

weight  Value assigned to an item on a scorecard that indicates the relative importance of that item in the calculation of the overall scorecard score. The weighting of all items on a scorecard accumulates to 100%. For example, to recognize the importance of developing new features for a product, the measure for New Features Coded on a developer’s scorecard would be assigned a higher weighting than a measure for Number of Minor Defect Fixes.

wild card  Character that represents any single character (?) or group of characters (*) in a search string.

WITH section  In MaxL DML, an optional section of the query used for creating re-usable logic to define sets or members. Sets or custom members can be defined once in the WITH section, and then referenced multiple times during a query.

workbook  An entire spreadsheet file with many worksheets.

write-back  The ability for a retrieval client, such as a spreadsheet, to update a database value.

ws.conf  A configuration file for Windows platforms.

wsconf_platform  A configuration file for UNIX platforms.

XML  See Extensible Markup Language (XML).
Y axis scale  Range of values on Y axis of charts displayed in Investigate Section. For example, use a unique Y axis scale for each chart, the same Y axis scale for all Detail charts, or the same Y axis scale for all charts in the column. Often, using a common Y axis improves your ability to compare charts at a glance.

Zero Administration  Software tool that identifies version number of the most up-to-date plug-in on the server.

zoom  Sets the magnification of a report. For example, magnify a report to fit whole page, page width, or percentage of magnification based on 100%.

ZoomChart  Used to view detailed information by enlarging a chart. Enables you to see detailed numeric information on the metric that is displayed in the chart.
## Index

### A
- access file system, script, 81
- access permissions, physical resources and, 42
- activating services, 167
- Administer module, 39
  - priority settings in, 40
  - common tasks, 36
  - introduced, 23
- administering public job parameters, 128
- administrator password, 162
- advanced scripting
  - access file system, 81
  - addChild(), 94
  - addProperty(), 94
  - addVersion method, 86
  - batch input files, 81
  - BqyDocument object, 98
  - calling scripts, Workspace, 74
  - cloneNode(), 95
  - close(), 98
  - compressBqy(), 99
  - convertBqyFileToUnicode method, 88
  - copy(), 99
- custom scripting environment, Workspace, 73
- custom scripts, 74
- document object model
  - access properties, 80
  - collections, 80
  - Dashboard Studio Inspector Utility, 80
  - described, 75
  - differences, 79
  - document conversion and loading strategies, 76
  - Impact Management Services, Workspace, 80
  - property types, 81
  - traverse, 77
  - xpath-style searching, 78
  - expandRequestAction(), 82
  - findFiles method, 88
  - findNodeByPattern(), 95
  - findNodesByPattern(), 95
  - folderExists method, 89
  - getBqyDocument(), 83
  - getChildren(), 95
  - getCurrentFolder method, 89
  - getFileLines(), 83
  - getFileUuid method, 89
  - getFolderContentsFor method, 90
  - getFolderUuid method, 90
  - getPathForUuid method, 90
  - getPathWithContext(), 96
  - getParameterValue(), 84
  - getParameterValues(), 84
  - hasProperty(), 97
  - impact Manager module, 73
  - isBQYProcessable(), 99
  - isCompressed(), 100
  - isDashboard(), 100
  - isDesktopMode(), 85
  - isFile method, 91
  - isFolder method, 91
  - isServerMode(), 85
  - Java code, 81
  - loadScript(), 85
  - logging, 58
  - makeFolder method, 91
  - monitor script environment, 74
  - node object, 94
  - publishBqyFile method, 92
- references
document, 104
env methods, 101
methods and properties, 101
node methods, 103
repository methods, 102
repository methods, Workspace-specific, 103
removeChild(), 97
removeProperties(), 97
replaceChildNode(), 98
repository artifact, 86
retrieveFile method, 93
retrieveVersionedDocument method, 93
script parameters, 74
scriptenvironment object, 82
scripting references, 82
sectionOECEPairInfos(), 100
setParameterValues(), 84
writeBqyDom(), 86
alternate URL, specifying for Shared Services, 180
Analytic Bridge Service. See Extended Access for Hyperion Interactive Reporting Service.
Apache, 200
APIs
exceptions, 47
triggering events with, 128
appenders, 200
applications
command strings, 153
triggering events with, 128
enterprise-reporting, 160
running jobs against enterprise applications, 160
URL property, 188
Applications properties (of servlets), 188
archives. See backing up.
ASMTMP authentication, notifications and, 48
Assessment Service, 165
associating interactive reports with Interactive Reporting database connections, 107
attachments
enabling, 48
maximum size, 48
Authentication Service, defined, 15
Authentication tab, removing, 134
Authorization Service, defined, 15
backing up
overview, 193
procedures, 194
repository database, 198
restoring data, 194
scripts, 197	servlet configuration files, 198
servlets, 198
batch files, and application command strings, 153
batch input files
scripts, 81
using, 81
bijobs_ior.txt, 33
bijobs_stderr.txt, 33
bijobs_stdout.txt, 33
bookmarks
generated Personal Page and, 137
setting up graphics for, 139
BqyDocument object
Interactive Reporting Studio, 98
Workspace, 98
Broadcast Messages
changing default Personal Pages and, 138
genrated Personal Page and, 136, 137
overview, 137, 138
pushed content, 138
renaming folder, 40
renaming folders, 139
specifying categories for, 40
subfolders, 137
understanding, 137
Browse servlet
Personal Page preconfiguration, 136, 137
Web application deployment name, 49
Browser properties, 187
buttons
LSC toolbar, 166
Servlet Configurator toolbar, 178
buttons, toolbar
LSC, 166
View Job Execution Log Entries, 126
C
Cache properties group
Browser subgroup, 187
described, 186
Notification subgroup, 187
Objects subgroup, 186
System subgroup, 187
caches
  content listings, 173
  services, 168
  user interface elements, 173
Calendar Manager, launching, 123
  calendars
    creating, 124
    deleting, 124
    end dates, 126
    managing, 123
    modifying, 124
    non-working days, 125
    periods and years, 125
    properties, 125
    user-defined weeks, 125
    week start, 125
  calling scripts
    Workspace, 74
cdsconfig.jsp, 134
classpath, 148, 173
clear log entries, 122
  client-side debug mode, disabling, 36
  color properties, described, 182
  color schemes
    customizing on Personal Pages, 141
    properties, 184
column
  data type changes, 120
delete, 118
  rename, 115
    example, 116
command strings for applications
  described, 153
  example, 152
config.dat file
  distributing services and, 161
  editing, 143, 162
  encryption and, 161
  services startup and, 161
  sync host properties, 174
ConfigFileAdmin utility, 161, 162
configfileadmin.bat. See ConfigFileAdmin utility.
ConfigFileAdmin.sh. See ConfigFileAdmin utility.
configInfo.jsp, 22, 199
configuration files
  backing up, 198
  BpmServer.properties, 174
  config.dat, 161
  output.properties, 47
  portal.properties, 175
  RMService8.properties, 30
  sc.properties, 164
  server.xml, 174
  ws.conf, 177
configuration information, 22, 199
configuration properties, 180
configuration URL, disabling, 36
configuration, ConfigFileAdmin utility, 162
Confirmation dialog box, 65
connections pool, 171
connectivity type, 171
connectivity, defining for database servers, 156
content windows, headings, 141
content, optional, Personal Pages, providing to users, 139
conventions, naming, 147
copying Personal Pages, 140
Core Services, 15
  creating
    data models, 115
    Interactive Reporting database connections, 106
  credentials, user, 175
  CSS Config File URL, 174
  custom login implementation, 179
  custom scripting environment, Workspace, 73
  custom scripts
    Interactive Reporting Studio, 74
    Workspace, 74
  custom update, 69
  choose script, 70
  create parameter file, 71
  parameters, 70
  scheduling options, 71
  sort data model topics script, 69
  verifying, 73
  workflow, 57
  customizations, user interface, 133

D
Dashboard Studio Inspector Utility (advanced scripting)
  document object model
    access properties, 80
    collections, 80
investigate, 80
property types, 81
DAT files. See server.dat; config.dat.
data model, character sets, replacement, Unicode, 121
data models
creating, 115
normalized and denormalized, 118
data sources
connectivity type, 171
listing, 170
maximum connections, 171
name, 50, 171
properties of Hyperion Interactive Reporting Data
Access Service, 170
data type column changes, 120
database properties (of host or Install_Home), 173, 174
database servers
adding, 155
associating with Job Service, 156
deleting, 157
drivers, changing, 157
environmental variables for SQR Production
Reporting, 156
managing, 155
databases
connectivity, 50
passwords, changing, 157
type, 171
types, 50
default language selections, changing, 133
default Personal Pages, changing, 138
default preferences, assigning for users and groups, 41
deleting
MIME types, 46
services, 145
diagnostics properties, 188
directories. See output directories.
directories, naming conventions for, 147
displayable items. See file content windows.
displaying HTML files on Personal Pages, 139
document object model. See DOM
documents, tracking, 52
DOM
access properties

Impact Management Services, Workspace, 80
Interactive Reporting Studio, 80
collections
Impact Management Services, Workspace, 80
Interactive Reporting Studio, 80
differences
Impact Management Services, Workspace, 79
Interactive Reporting Studio, 79
document conversion and loading strategies, 76
investigate structure, Impact Management Services, Workspace, 80
property types
Impact Management Services, Workspace, 81
Interactive Reporting Studio, 81
traversing, 77
tree structure, 75
xpath-style searching, 78
drivers, database, 148, 158, 173

enable

E
EnableNativeUserPasswordChange parameter, 134
Event Service
defined, 16
event logs, 40
service type, in server.dat file, 28
events
externally triggered, creating, 127
time, managing, 127
tracking, 52
triggering, 128
triggering with APIs, 128
exceptions
configuring, 140
described, 47
Exceptions Dashboard
described, 47
generated Personal Page and, 137
expiration times, 48
exporting registry keys, 194
Extended Access for Hyperion Interactive Reporting
Service, 17, 165
externally triggered events
creating, 127
polling for, 175
F
file content window, 139
file size, of e-mail attachments, 48
file systems
accessing within script, 81
backing up, 193, 194, 197
restoring data, 194
files
adding to folders, 138
in e-mail attachments, 48
Financial Reporting servers, defined, 17
firewalls, 185
folders
administrator-only System folder, 135
Broadcast Messages, 40
importing items in, 138
organizing, 135
preconfigured, 138
foreground jobs, 130
functional services, 16

G
generate parameter files, show impact of change
interactive reports, 114
Generated Personal Page properties (of servlets), 183
Global Service Manager. See GSM.
graphics. See images.
group preferences, default, assigning, 41
GSM
defined, 15
host general properties and, 173
name, 173
guidelines, size and time, Impact Manager module, 57

H
Harvester Service. See Assessment Service.
headings (within Personal Pages), 140
hosts
adding, 153
deleting, 154
managing, 153, 154
modifying, 154
properties of, 167, 172, 173
HTML files
customizing generated Personal Page with, 137
displaying on Personal Pages, 139
HTTP protocol, SmartCuts for e-mail notifications and, 49
HTTPS protocol, 49
Hyperion Home, 21
Hyperion Interactive Reporting Data Access Service configuring, 106
data sources, adding, 172
database connection properties, 172
defined, 17
response timeout, 189
Hyperion Interactive Reporting Service
defined, 16
physical resources and, 43
icons
changing MIME type, 45
DBCS and, 140
MIME types and, 44
files, 135
for HTML output of Production Reporting, 181
on Exceptions Dashboard, 47
Servlet Configurator, 178
images
for bookmarks, setting up, 139
replacing in logon panel, 133
Impact Management Assessment Service, 54
metadata, 54
Impact Management Services
about, 54
accessing, 59
Impact Management Update Service, 55
running, 58
script logging levels, 58
Impact Manager module, time and size guidelines, 57
importing Interactive Reporting database connections, 106
inactivating
MIME types, 45
services, 167
IncludeJavaClientItems parameter, 134
Install Home, 21, 24, 172, 195
installation
backing up immediately after, 195, 198
config.dat file, 162
installed services, 25
installed servlets, 177, 198
installed system, 40
location of components, 143, 166, 171, 177, 178
location of installed files, 175
new hosts, 153
Servlet Configurator, 177
of Zero Administration, 191
installation directory, 173
installation program, 37, 145
installed services
deleting host with, 154
Install Home, 166, 172
Interactive Reporting services, 28
LSC displays, 166
LSC toolbar, 166
recommendations for Job Service, 160
replicating Job Service, 160
RSC toolbar, 144
installed servlets, 164
Interactive Reporting
load testing, 191
Zero Administration, 189
Interactive Reporting database connections
associating with interactive reports, 107
creating, 106
explicit access property, 128
importing, 106
managing, 128
Interactive Reporting documents, changing user name
and password in, 121
interactive reports, connecting, 105
Internal properties (in Servlet Configurator)
Job subgroup, 185
Transfer subgroup, 185
Upload subgroup, 186
IP ports. See ports.
items
generated Personal Page and, 136
headings on, 140
organizing in folders, 135, 138

J
Java client installers, removing access to, 134
Java code in scripts, 81
JavaScript update, 65
choose control mode, 66
parameters, 66
scheduling options, 67
verifying, 68
workflow, 56
JDBC driver, 148, 173
JDBC URL, 148, 158, 173
job execution, job process explained, 160
Job Log
columns in, 126
dates, 126
deleting entries, 127
marking entries for deletion, 127
sorting, 126
start dates and times, 126
suppressing, 159
user displays for, 126
job output property, 181
job parameters, public, administering, 128
Job Service
applications
adding, 151
configuring, 151
properties, 151
defined, 16
executable of application, 151
properties
Application, 151
database, 150
Executable, 153
Production Reporting, 150
running jobs against enterprise applications, 160
service type, in server.dat file, 28
user name for running SQR Production Reporting
jobs, 50
jobs
e-mail file attachments, 48
Job Log, 125
jobs property, 185

K
kill commands, 29

L
languages, changing default selection of, 133
Local Service Configurator. See LSC.
Local Service Manager. See LSM.
locale-based language selection, overriding, 133
localization properties, 180
log files
  analyzing, 209
  configuration, 203
  configuration log, 203
  file formats, 203
for Interactive Reporting document output, 175
location, 201
naming convention, 201
notification log, 48
Log Management Helper, 200
log4j, 200
loggers, 200
logging
  Interactive Reporting Studio, 58
  Workspace, 58
logging events, 125
logging levels, 204
Logging Service
  configuration, 203
  defined, 16
  usage, 200
logon panel, replacing image in, 133
logon_panel.gif, 133
LSC
  described, 24
  modifying host properties, 173
server.xml file, 166, 174
services list, 165
starting, 166
toolbar buttons, 166
LSC services
  defined, 15
  list of, 165
  super service and, 168
LSM, 15, 173

M
magnifying glass icon, 178
mail server host names, 48
manage task list, 109
  undo, 110
  undo, show actions, 111
management services, 16
managing
  Interactive Reporting database connections, 128
time events, 127
metadata
  Impact Management Assessment Service, 54
methods and properties
  references, 101
  document, 104
  env methods, 101
  node methods, 103
  repository methods, 102
  repository methods, Workspace-specific, 103
Microsoft Windows. See Windows.
MIME type icons, changing, 45
MIME types
  creating, 44
  deleting, 46
  inactivating, 45
  modifying, 45
  reactivating, 45
  working with, 44
modules
  Administer, 39
  Impact Manager, 24, 57
monitor script environment
  Interactive Reporting Studio, 74
  Workspace, 74

N
Name Service
  config.dat file and, 161
  defined, 16
  service type, in server.dat file, 28
  startup process and, 162, 163
naming conventions, directories, 147
node object
  Interactive Reporting Studio, 94
  Workspace, 94
notification property, servlets, 187
notifications and subscriptions
  authentication and, 48
e-mail
  account for sending, 48
  enabling attachments, 48
  server host name, 48
events that trigger, 47
  expiration times, 48
  file attachments, 48
  other, 46
types, 47
understanding, 46

O
object properties, 186
OCEs. See Interactive Reporting database connections.
Oracle Reports, command string example, 152
organizing items and folders, 135
output directories
adding, 43
deleting, 43
modifying, 43
purpose, 42
output.properties file, 47

P
parameters, job, public, administering, 128
pass-through, 175
credentials, 129
defined, 129
passwords
database, changing, 157
crypted, 161
Interactive Reporting, changing, 121
Job Service, for running SQR Production Reporting
jobs, 50
RDBMS, 163
Repository Service, 157
services, 161
services, for database access, 148, 174
services, modifying, 162
system, 161
passwords, preventing users from changing, 134
Personal Pages
Broadcast Messages on, 138
configuration tool, 140
customized graphics, 139
default, 138
generated
customizing, 137
properties, 183
setting up, 136
graphic files on, 139
importing, 140
importing other pages, 140
multiple, 138
optional content, providing to users, 139
properties, configuring, 140
setting up items in folders, 138
viewing new user's, 140
Personal Pages properties
color scheme, 184
Publish, 183
physical resources
access permissions on, 42, 43
adding, 43
deleting, 43
modifying, 43
pinging services, 146
port assignments, services, changing, 30
portal.properties file, 175
ports, Browse servlet, 49
postLogoffURL parameter, 135
preconfigured folders, setting up, 138
printers
adding, 43
deleting, 43
modifying, 43
properties, 43
purpose, 42
priority settings, in Administer module, 40
programs, running jobs against enterprise applications, 160
properties
configuring Personal Pages, 140
generated Personal Page, 137
Job Service, 149
Servlet Configurator, 178
Shared Services, 174
standard, of LSC services, 167
viewing in Servlet Configurator, 178
protocols for SmartCuts, 49
Provisioning Manager Shared Services role, 41
publish properties, 183
Publisher Service, defined, 15
publishing Personal Pages, 140
pushed content, 135, 138
pushing content. See Broadcast Messages.

Q
queries
maximum bytes in results, 170
maximum rows in results, 170

R
RDBMS
passwords, 163
starting, 25
reactivating MIME types, 45
recovering data, 194
redirecory URLs, after users log off, 135
references
document, 104
env methods, 101
methods and properties, 101
node methods, 103
repository methods, 102
repository methods, Workspace-specific, 103
registry keys, 194
relational database management system. See RDBMS.
Remote Service Configurator. See RSC.
renaming Broadcast Messages folders, 139
replicating Repository Service, 148
replicating servlets, 177
report registry keys, 193
repository
addVersion method, 86
artifact, 86
convertBqyFileToUnicode method, 88
findFiles method, 88
folderExists method, 89
getCurrentFolder method, 89
getFileUuid method, 89
getFolderContentsFor method, 90
getFolderUuid method, 89
getRepository method, 90
getSubfolderPathsFor method, 91
isFile method, 91
isFolder method, 91
makeFolder method, 91
publishBqyFile method, 92
retrieveFile method, 93
retrieveVersionedDocument method, 93
repository artifact
Interactive Reporting Studio, 86
Workspace, 86
repository database, backing up, 198
Repository Service
defined, 15
replicating, 148
service type, in server.dat file, 28
stopping, 29
request queue polling interval, 122
restoring data, 194
RMService8.properties file, 30
root directories, 49
row-level security, 49
RSC
config.dat file and, 161
database password, 157
described, 24
pinging services, 146
Storage tab, 157
what it does, 143
RSC services
configuring, 24
setting properties, 174
run now, synchronizing, 59
Run Type property, 167
Run Type property of services, 27
S
schedules, synchronizing, 60
script logging levels, 58
script parameters, Workspace, 74
script environment object
Interactive Reporting Studio, 82
Workspace, 82
scripting references
addChild(), 94
addProperty(), 94
BqyDocument object, 98
closeNode(), 95
close(), 98
compressBqy(), 99
copy(), 99
expandRequestAction(), 82
findNodeByPattern(), 95
findNodesByPattern(), 95
getBqyDocument(), 83
getChildren(), 95
generateFileLines(), 83
getInitialTCSection(), 99
getNodeType(), 96
getParameterValue(), 84
getParameterValues(), 84
getPathWithContext(), 96
getProperty(), 96
getRepository(), 84
getScriptUuid(), 85
hasProperty(), 97
Interactive Reporting Studio, 82
isBQYProcessable(), 99
isCompressed(), 100
isDashboard(), 100
isDesktopMode(), 85
isServerMode(), 85
loadScript(), 85
node object, 94
removeChild(), 97
removeProperties(), 97
replaceChildNode(), 98
scriptenvironment object, 82
sectionOCEPairInfos(), 100
setParameterValues(), 84
Workspace, 82
writeBqyDom(), 86
scripting, advanced, 73
scripts
  batch input files, 81
  Java code, 81
sort data model topics, 69
start_BIPlus, 26
startCommon Services, 26
for stopping services, 29
server-side software components. See services.
server.dat file, 27, 28
server.xml file, 174
servers, mail, 48
Service Broker
  defined, 16
  service type, in server.dat file, 28
service configuration parameters, Impact Manager
  module and, 122
  clear log entries, 122
  request queue polling interval, 122
Service Name property, 167
service port assignments, changing, 30
services
  adding, 165
  common tasks, 36
Core, 15
deferred, 16
deleting, 145
functional, 16
message during deletion, 145
modifying properties, 146
names in BrioPlatformxxx.dat, 28
names in server.dat, 28
pinging, 146
properties
  Advanced, 147
  General, 147
removing, 143, 165
replicating, 148
Run type property, 27
running as separate processes, 29
server.dat file, 28
starting on UNIX, 26
starting on Windows, 26
starting subsets of, 27
stopping, with scripts, 29
summary, 18
types of, 28
user name for database access, 148, 173
viewing properties, 146, 173
Servlet Configurator
  defined, 177
  described, 25
  making new settings effective, 178
  starting, 178
toolbar, 178
server engine, session timeout value, 185
server-side software components. See services.
server.dat file, 27, 28
server.xml file, 174
servers, mail, 48
Service Broker
  defined, 16
  service type, in server.dat file, 28
service configuration parameters, Impact Manager
  module and, 122
  clear log entries, 122
  request queue polling interval, 122
Service Name property, 167
service port assignments, changing, 30
services
  adding, 165
  common tasks, 36
Core, 15
deferred, 16
deleting, 145
functional, 16
message during deletion, 145
modifying properties, 146
names in BrioPlatformxxx.dat, 28
names in server.dat, 28
pinging, 146
properties
  Advanced, 147
  General, 147
removing, 143, 165
replicating, 148
Run type property, 27
running as separate processes, 29
server.dat file, 28
starting on UNIX, 26
starting on Windows, 26
starting subsets of, 27
stopping, with scripts, 29
summary, 18
types of, 28
user name for database access, 148, 173
viewing properties, 146, 173
Servlet Configurator
  defined, 177
  described, 25
  making new settings effective, 178
  starting, 178
toolbar, 178
server engine, session timeout value, 185
server-side software components. See services.
server.dat file, 27, 28
server.xml file, 174
servers, mail, 48
Service Broker
  defined, 16
  service type, in server.dat file, 28
service configuration parameters, Impact Manager
  module and, 122
  clear log entries, 122
  request queue polling interval, 122
Service Name property, 167
service port assignments, changing, 30
services
  adding, 165
  common tasks, 36
Core, 15
deferred, 16
deleting, 145
functional, 16
message during deletion, 145
modifying properties, 146
names in BrioPlatformxxx.dat, 28
names in server.dat, 28
pinging, 146
properties
  Advanced, 147
  General, 147
removing, 143, 165
replicating, 148
Run type property, 27
running as separate processes, 29
server.dat file, 28
starting on UNIX, 26
starting on Windows, 26
starting subsets of, 27
stopping, with scripts, 29
summary, 18
types of, 28
user name for database access, 148, 173
viewing properties, 146, 173
Servlet Configurator
  defined, 177
  described, 25
  making new settings effective, 178
  starting, 178
toolbar, 178
server engine, session timeout value, 185
size guideline, 57
SmartCuts
  e-mail notifications, 39, 49
  in notifications, 47, 48
  servlet property, 182
  system properties, 49
Solaris (Sun), 197. See also UNIX systems.
SQR Production Reporting
  environment variables for, 156
  jobs, data sources for, 150
  persisting job output, 130
  server properties, 150
start scripts, 26, 174
start_BIPlus scripts, 26
startCommonServices script, 26, 174
starting on UNIX, 26
starting RDBMS, 25
starting system or components
  administrative tools
    LSC, 166
    RSC, 143
  Servlet Configurator, 178
services
  config.dat file and, 161
  dependencies, 29
  starting on UNIX, 26
  starting on Windows, 26
  starting subsets of, 27
  super, 168
  servlets, 30
stylesheets, 182
Subscribe page, 47
subscription property, 181
subscriptions and notifications, 46
super service, 168
synchronize metadata, 59
  run now, 59
  schedule, 60
system configuration information, 22, 199
System folder
  administrator only, described, 135
  viewing, 135
system properties
  in Administer module, 41
  described, 187
  system usage, 50
T
table, rename, 115
  example, 116
time events, managing, 127
time guideline, 57
timeout settings
  Workspace, 22
titles on items, 140
toolbar buttons
  LSC, 166
  RSC, 143
toolbars
  Service Configurator, 143, 166
  Servlet Configurator, 178
tracking
  documents, 52
  events, 52
transfer property, 185
Transformer Service. See Update Service.
triggering events, 128
troubleshooting, 199
troubleshooting code, disabling, 36
trusted passwords
  configuring, 179
  host authentication properties and, 175
U
undo
  manage task list, 110
  show actions, 111
Unicode, data model, character sets, replacement, 121
UNIX systems
  backing up clients, 198
  backup procedure, 194
  kill commands and, 29
  maximum number of file descriptors, 147
  starting Service Configurator, 143, 166
  starting Servlet Configurator, 178
  system backup procedures, 194
unknown file type messages, 46
update data models
  jobs, 65
  link data models and queries, 55
  scheduling options, 64
  specify data model, 62
  transformation, 55
undo jobs, 65
view candidates to update, 63
workflow, 56
update JavaScript
  choose control mode, 66
  parameters, 66
  scheduling options, 67
  verifying, 68
Update Service, 165
updating, data models, 61
updating, JavaScript, 65
upload property, 186
URL properties, 188
URLs, redirecting after users log off, 135
Usage Service
  defined, 16, 50
  managing, 51
  reports, 52
usage tracking
  accessing Organization tab, 51
  properties, 51
user interface customizations, 133
user names, Interactive Reporting, changing, 121
user preferences, default, assigning, 41
users
  common tasks, 36
  Job Log displays for, 126

V
View Manager, pushed content, 135
virus protection, 135

W
Web modules. See servlets.
Windows (Microsoft)
  backup procedure, 193
  exporting Registry key, 194
  plug-ins, 191
  Windows 2000, 197
Workspace configuration information, 22, 199
Workspace test module, disabling, 36
ws.conf file, 177
wsconf_platform file, 177

Z
Zero Administration