HYPERION® PERFORMANCE SUITE™

RELEASE 8.5 AND RELEASE 8.3.1

ADDENDUMS

Hyperion®

P/N: D835485000

Copyright 1989-2006 Hyperion Solutions Corporation.
All rights reserved.

"Hyperion," the Hyperion "H" logo, and Hyperion's product names are trademarks of Hyperion. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

No portion hereof may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the recipient's personal use, without the express written permission of Hyperion.

The information contained herein is subject to change without notice. Hyperion shall not be liable for errors contained herein or consequential damages in connection with the furnishing, performance, or use hereof.

Any Hyperion software described herein is licensed exclusively subject to the conditions set forth in the Hyperion license agreement.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable Hyperion license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

Hyperion Solutions Corporation
5450 Great America Parkway
Santa Clara, California 95054

Printed in the U.S.A.

Contents

PART I Release 8.5 Addendum ix

CHAPTER 1 Hyperion Performance Suite Installation Guide Changes 7
Upgrade Overview 8
General Upgrade Recommendations 8
Upgrade Recommendations for Distributed Systems 8
Upgrade Prerequisites 9
Upgrade Instructions 10
Installing Hyperion Performance Suite Server Components: New Express Installation 10
Installing the Hyperion Performance Suite Client Tools 13
JRun 14
Installation Directory Structure 14
Apache Tomcat 4.1.30 15

CHAPTER 2 System-Wide Changes 17
User Defined Languages 18
User Interface Language 23
Unicode Enabling Software 27
BQY Sort Order Language 28
Encoding OCE Files 29
Hyperion Intelligence Application Configuration Files 29
Unicode Migration Utility for BQYs and OCEs 30
Unicode File Path Handling on UNIX 31
Unicode Compliant Fonts 31
Unicode Font Configuration 31
Unicode Compliant Font Preferences 31
File Import 34
File Export 35
Connectivity 36
DB Connectivity 36
Migration Tool 37
National Strings for MS SQL Server 39

Bidirectional Text 39
Understanding Bidirectional Text 40
Bidirectional Configuration Settings 40
Controlling Document Versions in the Hyperion Intelligence Client 41
ADR Control Flags 41
Row Level Security Enhancements 42
OR Logic Between Groups 42
JOINOPER Operator 42
Software Developer Kit Changes 43

CHAPTER 3 Hyperion Foundation Enhancements 45
Alerting for Relational Queries and Reporting 46
Job Content (Cascade) Deletion 46
BQY Job Log 46
Job Queuing 46
Scheduled Jobs 47
Background Jobs 47
Foreground Jobs 47
Disabling the Job Service 48
Job Limits 48
Acquiring Jobs 48
Jobs Only Intelligence Service 48
Native Authentication Passwords 49
Creating a Mini-Administrator 49
Triggering Events 49
TriggerExternalEvent.java Sample API 49

CHAPTER 4 Hyperion SQR Enhancements 51
Secure Report Bursting 52
Functional and Platform Improvements to SAP BW Reporting 53
Increased Platform Support of SAP BW to 3.0B, 3.1 and 3.5 53
Certification for SAP NetWeaver (OLAP BAPIs) 53
Data Cursoring for High Volumes of Data Processing 53
Access to ODS Objects, InfoSets, and SAP Variables 54
Currency Conversion 55
Unit of Measure Conversion 56
ANCESTOR and DESCENDANT Clauses 57
Support for Key Figure Attributes – Currency, Formatted, Unit, Value Type 57
Enhancements to SQR Developer 59
Streamlined Query Builder 59
Report Layout Explorers 60
Editor Unicode Support 63
User Interface Enhancements 63
SAP BW Enhancements 64

New Excel Output Type 78
New Charting Functionality 78
New Attributes for the Number of Columns and Rows in a Legend 78
New Attributes for Pie Chart Label Location and Pie Chart Percentage Precision 80
Ability to Specify New Colors and Fonts 81
Greater Control of the Name and Color for the “Other” Pie Chart Slice 81
UFUNC Unicode Support 82
Support for New Encoding 82
Ability to Call C++ Libraries from an SQR Program 82
Ability to Add a Last Page Value to a Bar Code 84
New Paper Size in DECLARE-LAYOUT 85

CHAPTER 5 Hyperion Intelligence Changes 87
Exporting to Microsoft Office HTML File Formats 88
Formula Mappings 94
Custom Formulas 99
Customizing Chart Colors 102
General Chart Properties 104
Computed Items 104
Document Properties 104
Hyperion Intelligence Web Client Coexistence 105
Locally Saved BQYs 106
Hyperion Intelligence Web Client Mismatches 106
Increasing the Idle Connection Time-outs for Hyperion Intelligence Web Client 107
Dashboards and Object Model Enhancements 107
UnicodeEnabled (Property) 108
SortOrderLang (Property) 109
DefaultSortOrderLang (Property) 111
UILanguage (Property) 117
ResetDefaultSortOrderLang (Method) 128
Export (Method) 129
ExportToStream (Method) 132
ImportDataFile (Method) 134
ImportSQLFile (Method) 136
Embedded Pivot Section Interactivity 137
OnCellDoubleClick (Method) 138
SideLabelValues (Array) 139
TopLabelValues (Array) 140
CellValue (Property) 143
FactName (Property) 145
Printout (Method) 148
Object Model Exclusion List 148

PART II Release 8.3.1 Addendum 153

CHAPTER 6 Hyperion Performance Suite Installation Guide Changes 155
Apache Tomcat Port Conflict with Oracle 9.2.x 156
Apache Tomcat Precompile Script 156
BEA WebLogic 6.*x* with WebLogic SP3 or Later Manual Configuration Instructions 157
BEA WebLogic 7.*x* Manual Configuration Instructions 157
IBM WebSphere 5.0.1, 5.0.2, and 5.1 Manual Configuration Instructions 157
Configuring IBM WebSphere 5.0.1, 5.0.2, or 5.1 157
Configuring IBM WebSphere 5.0.1, 5.0.2, or 5.1 Manually 158
Oracle Configuration 162
Cessation of Services Before Upgrading 162
Configuring SunONE_iPlanet 6.1 on Solaris 163
Deploying the Hyperion Web Applications on SunONE iPlanet 6.1 (Solaris) 163
Precompiling SunONE iPlanet 6.1 166
Removing the Deployed Hyperion Web Applications 167
UNIX Temporary Directories 168

CHAPTER 7 Hyperion Performance Suite Enhancements 169
Accessibility Changes 170
Backward Compatibility 170
Compatibility Between Different Client and Server Releases 170
Compatibility Within BQY Documents Across Versions 171
Storage of Content Created by One Software Release in a Repository Managed by a Different Release 172
Default Settings for Services 172
Default Settings for Servlets 173
Tips on Distribution and Replication of Services 174
Supported Web Servers 174

CHAPTER 8 Hyperion Foundation Enhancements 175
New Look 176
Browse Module Enhancements 176
Browse Preference to Hide Job Output Listings 177
SQR Jobs Output Format 177
SQR and Generic Job Output Folder 178
Automatically Deleting Job Outputs for Schedules 178
New Mime Type for BQY Documents Generated by Generic Jobs 178
External Authentication Nested Group Support 179
Viewing or Modifying Group Memberships 179
Publish Stand-alone BQY Document Job Action 180
Replace BQY Document or Job 180
End-User Changes for Replace BQY 180
Administrator Changes for Replace BQY 181

CHAPTER 9 Hyperion Intelligence Enhancements 183
New Icons 184
Null Values in Pivot Section 184
Object Model Enhancements 185
Width (Property) 185
Session (Object) 186

CHAPTER 10 Hyperion SQR Enhancements 187
SAP BW Data Source Enhancements 188
New Syntax to Sort by Dimension Properties or Selected Measures 188
New Ranking Operators for Filtering 190
Hierarchies Under Characteristics 191
Restricting Data Retrieved 191
Support for BINARY Data 193
Specifying Whether a Variable or Column Supports BINARY Data 193
New Keywords in the SQR.INI File 193
Modified SQR Commands 194
Specification of a Separator Value for CSV File Generation 195
Deprecated Thin-Client Analysis Options in SQR Developer 196

Index 197

Part I

Release 8.5 Addendum

1 Hyperion Performance Suite Installation Guide Changes

This chapter explains how to upgrade to the Hyperion Performance Suite 8.3 or 8.5 releases.

Caution! These installation options rely on the InstallShield-generated `vpd.properties` file (in `C:\WINNT` or your UNIX *user home*) as well as the *`INSTALL_HOME`*`\log\InstallHistory.xml` file. Do not change or delete these files; doing so may cause undesirable results. In addition, Hyperion recommends that you do not cancel the installation process when doing an upgrade as doing so may reset the default values from the previous installation.

In This Chapter

Upgrade Overview 8
General Upgrade Recommendations 8
Upgrade Recommendations for Distributed Systems 8
Upgrade Prerequisites 9
Upgrade Instructions 10
JRun 14
Installation Directory Structure 14
Apache Tomcat 4.1.30 15

Note: For Hyperion SQR installation instructions, see the *Hyperion SQR Installation Guide*.

Upgrade Overview

The upgrade option is available only when you have an existing *INSTALL_HOME* that is a different release from the Hyperion Performance Suite installer release. The installer automatically displays the upgrade option based on the *INSTALL_HOME* that you select as your destination directory. If you select an *INSTALL_HOME* whose release number does not match the installer release, the installer automatically sets the installation setup type to Upgrade. The Upgrade option updates binaries and the repository, adding new columns or indexes as necessary.

Note: Upgrading to Hyperion Performance Suite release 8.5 is only available for installations of Hyperion Performance Suite 8.3x.

General Upgrade Recommendations

When upgrading to Hyperion Performance Suite releases 8.3 or 8.5, please keep these points in mind:

- If you are upgrading your operating system to match one that is newly supported in release 8.5, first upgrade your operating system and test your Hyperion Performance Suite installation; then upgrade to Hyperion Performance Suite releases 8.3 or 8.5.
- Hyperion Performance Suite release 8.5 supports different Web servers than previous releases. If your previous Web server is no longer supported in release 8.5, you can either upgrade your Web server or install a new Web server; or manually deploy the servlets on your older Web server.
- Add new components only after the upgrade is complete and tested. This helps to ensure the integrity of the data in your system.
- If you are upgrading from ReportMart 1.x, first upgrade to Brio Portal 6.0 using the 6.0 CD, then upgrade to Brio Portal 7.x, then install Hyperion Performance Suite and migrate your Brio Portal 7.x installation. Please contact Hyperion Solutions Consulting Services for assistance.
- Always back up your file system and database prior to updating your system in case an upgrade fails.
- Save any file customizations in an old installation (for example, server.xml or ws.conf). An upgrade may restore these file to the new install state and wipe out the customizations.

Upgrade Recommendations for Distributed Systems

Hyperion recommends that you upgrade if you have a distributed system.

➤ To upgrade a distributed system:

1. **Shut down all services and Web servers.**
2. **Install or upgrade the services (or new Web server, if needed).**

3 **Test the servlets. (This can be done on a different Web server than the one used in your pre-release 8.5 installation; that is, it can be migrated at installation time.**

4 **Run the installer on each services host.**

5 **Test the services.**

Upgrade Prerequisites

Table 1 lists the prerequisites that you must meet before upgrading to Hyperion Performance Suite releases 8.3 or 8.5.

Table 1 Upgrade Requirements

Platform	Requirements
All platforms	When upgrading, please make sure that you complete all prerequisite tasks specified in "Required Pre-Installation Configuration Procedures" on page 35 of the *Hyperion Performance Suite Release 8.3 Installation Guide*.

IBM DB2	Tables v8_param_values and v8_jobinpmultvals created before release 8.2 are in a tablespace with only 4KB page size. For release 8.2.x and later, these tables need to be in a tablespace with a page size of 8KB.
	Hyperion has provided a script to migrate these tables to your new tablespace. First, follow the DB2 pre-installation instructions for creating the required regular and system temporary tablespaces (see "Configuring IBM DB2" on page 38). Then, copy the script from the Servers CD Disk 1 `\Misc\DB2` directory to a temporary location on your system.
	Caution! To avoid corruption of the required tables, Hyperion recommends running `pre_install_db2_upgrade.cmd` line by line using the DB2 console. Make sure that the current statement processes successfully before running the next one.
	To use the script on Microsoft Windows systems:
	Edit this file to define proper values for parameters that appear in angle brackets.
	NOTE: Make sure that the user you are connected as when you execute this script is the owner of modified tables and has the appropriate permissions to perform import, export, and delete operations.
	Run this command:
	`db2cmd pre_install_db2_upgrade.cmd`
	To use the script on UNIX systems:
	Edit this file to define proper values for parameters that are enclosed in angle brackets.
	NOTE: Make sure that the user you are connected as when you execute this script is the owner of modified tables and has appropriate permissions to perform import, export, or delete operations.
	Run this command to add executable attributes to the `pre_install_db2_upgrade.cmd` file:
	`chmod +x pre_install_db2_upgrade.cmd`
	NOTE: If the file contains line feed characters, use the dos2unix utility to remove them.
	Run this command to define standard DB2 environment:
	`source <SQLLIB>/db2cshrc`
	Run this command:
	`./pre_install_db2_upgrade.cmd`

Upgrade Instructions

These instructions assume that you have a a previous installation of Hyperion Performance Suite.

➤ To upgrade to Hyperion Performance Suite Release 8.5:

1 **Run the setup file for your system from the Hyperion Performance Suite Servers CD.**

Setup file names are:

- **AIX**—`setupAixPlatform.bin`
- **HP**—`setupHpPlatform.bin`
- **Linux**—`setupLinuxPlatform.bin`
- **Solaris**—`setupSolarisPlatform.bin`
- **Windows**—`setupWinPlatform.exe`

2 **Install to your existing *INSTALL_HOME*.**

3 **If prompted, select the same installation setup type (Express or Custom) that you previously chose.**

The installer performs an Express or Custom upgrade, prompting for information as necessary.

4 **After the upgrade is complete, proceed to Chapter 7, "Post Installation Tasks," in the Hyperion Performance Suite Release 8.3 Installation Guide for information about additional tasks that you must perform and for instructions on testing your upgraded installation.**

Installing Hyperion Performance Suite Server Components: New Express Installation

The following steps replace the ones shown on page 64 of the *Hyperion Performance Suite Release 8.3 Installation Guide*.

➤ To install Hyperion Performance Suite server components:

1 **Run the setup file for your system from the Hyperion Performance Suite Servers CD.**

Setup file names are as follows:

- **AIX**—`setupAixPlatform.bin`
- **HP**—`setupHpPlatform.bin`
- **Linux**—`setupLinuxPlatform.bin`
- **Solaris**—`setupSolarisPlatform.bin`
- **Windows**—`setupWinPlatform.exe`

Note: On UNIX systems, to avoid conflicts with `/tmp` directories created by other users, or access problems to the `/tmp` directory, use this syntax to specify the temporary directory you use when running the installer:

```
setupSolarisPlatform.bin -is:tempdir /home/user/tmp
```

where `/tmp` is a temporary directory with at least 1GB of space available. InstallShield also uses the `/var` directory for some temporary files, so make sure that `/var` is not full. You should make sure this temporary directory does not contain important files because the directory contents may be removed during an uninstallation.

When installing using a remote UNIX session, make sure that you set up display (--DISPLAY) to the terminal where you want the window displayed. Otherwise, the installation seems to freeze because the display is waiting for input on the UNIX host computer.

The InstallShield wizard starts and displays a progress bar while the JVM loads. It is followed by the Hyperion Performance Suite Servers splash screen and then by the Installation Language dialog box.

2 **Select the language that you want the installer to use during the installation process, then click Next to display the Welcome dialog box.**

3 **Click Next on the Welcome dialog box to proceed with the installation.**

4 **Select the country where you are installing the software.**

5 **Read the Software License Agreement. To continue with the installation, select the I accept the terms of the license agreement option and click Next.**

6 **In the Licence Key Entry dialog box, enter one or more license keys and click OK.**

You must enter at least one license key to install the Hyperion Performance Suite files to your computer. When prompted by the installation software, enter the license key provided with the installation media. The key should be in the following format:

6L55-1P2E2

where dashes separate license keys. You may have multiple license keys, depending on the number of Hyperion Performance Suite components that you purchased. You may add more keys at any time.

License keys are stored in *user_home*`/InstallHistory.xml` where *user_home* means `C:/Documents and Settings/`*`username`* on Windows or your user home (often `/home/`*`username`*) on UNIX.

Note: License keys are remembered for all subsequent installations on this computer. License keys are not case-sensitive.

7 **Verify that the correct components are listed, then click Next.**

To install a product that is not listed, click **Add License Key**, enter the appropriate license key, then click **OK**.

This sample installation installs the following products on a single computer:

- Hyperion Foundation
- Hyperion Intelligence Server
- Hyperion Intelligence iServer
- Hyperion SQR iServer

8 **In the Setup Type dialog box, select the Express installation setup type and click Next.**

9 **Select the destination directory (that is, an *INSTALL_HOME*), then click Next.**

Destination directory choices are as follows:

- **Default installation directory**—`C:\Program Files\Brio\Brio8` on Windows; `/opt/Brio/Brio8` on UNIX.
- **New directory**—Type the full path in the Directory Name text box.
- **Pre-existing directory**—Select **Browse,** then navigate to the desired directory and click **OK**.

If the directory does not exist, the installer creates it.

10 **Enter the Hyperion Home directory, if not previously specified.**

11 **Select the language and formatting locale for your Hyperion Performance Suite system to use.**

12 **Select the repository database type, then click Next.**

Select IBM DB2, Microsoft SQL Server, or Oracle as the Hyperion Performance Suite repository database type.

This sample installation uses Oracle for the repository database.

Note: If configuring Hyperion SQR iServer and the installer has not found the location of the `sqr.ini` file, you are prompted to enter its location. Enter or browse to the file location, or choose not configure SQR iServer on this system. If the location you enter is wrong, the installer shows a warning panel; you should go back and enter the valid location.

13 **If configuring Hyperion SQR iServer, verify the path to the root directory for your database client software and click Next.**

If necessary, use the Browse button to locate the correct directory. This information must be provided to continue installation.

For this sample installation, the location of the Oracle client libraries is as follows:

C:\Oracle\Ora8.1.6

14 **Enter the repository database connection parameters, then click Next.**

15 You recorded this information when you configured your database.

For more information, see the section specific to your database type under the "Required Pre-Installation Configuration" in the *Hyperion Performance Suite Release 8.3 Installation Guide*.

If you are in doubt as to the correct information necessary to fill these fields, contact your database or system administrator.

Installing the Hyperion Performance Suite Client Tools

The following steps replace the ones shown on page132 of the *Hyperion Performance Suite Release 8.3 Installation Guide*.

➤ To install the Hyperion Performance Suite client tools:

1 **Run `setupWinPlatform.exe` from the Hyperion Performance Suite Intelligence Clients CD.**

The InstallShield Wizard starts and displays a progress bar while the JVM loads, followed by the Hyperion Performance Suite Client Tools splash screen, then by the Language dialog box.

2 **Select a language, then click Next.**

The Welcome dialog box is displayed.

3 **Click Next to proceed with the installation.**

The Software License Agreement dialog box is displayed.

4 **Select the country where you are installing the software.**

5 **Read the Software License Agreement. To continue with the installation, select the I accept the terms of the license agreement option and click Next.**

The Enter License Key dialog box is displayed.

6 **If prompted, enter your license keys, then click OK.**

You must enter at least one license key to install the Hyperion Performance Suite files to your computer. When prompted by the installation software, enter the license key provided with the installation media. The key should be in the following format:

```
6L55-1P2E2
```

where dashes separate license keys. You may have multiple license keys, depending on the number of Hyperion Performance Suite components you purchased. You may also add keys at any time.

Note: License keys are retained by your system throughout an installation session.

7 **Verify that the correct components are listed, then click Next.**

8 **To install a product that is not listed, click Add License Key, enter the appropriate license key, then click OK.**

9 **If your license key unlocks more than one client tool, select a client tool, then click Next.**

10 **Select an installation setup type, then click Next:**

- **Express** —Installs all licensed Hyperion Performance Suite client tools on this host computer, including program files and sample files.
- **Custom**—Enables you to select which Hyperion Performance Suite client tool files you want to install.

11 **Enter the Hyperion Home directory, if not previously specified.**

12 **Select the destination directory, then click Next:**

- **Default installation directory**—The default is `C:\Program Files\Brio\Brio8` on Windows or `/opt/Brio/Brio8` on UNIX.
- **New directory**—Type the full path in the **Directory Name** text box. If the directory does not exist, the installer creates it for you.
- **Pre-existing directory**—Select **Browse,** then navigate to the directory and click **OK**.

If the directory does not exist, click **Yes** when the installer asks if it should be created.

13 **If you selected the Custom installation option, select the client tools or files you want to install, then click Next.**

Note: Sample files are available only for English.

14 **Select a Windows Start menu program folder, then click Next.**

15 **Verify your installation choices, then click Next. Click Back to change any of the listed parameters.**

The installer displays a progress bar while it automatically copies files into target directories.

16 **Complete the installation.**

Any errors that occur during the installation are listed on the dialog box that is displayed. The installer automatically selects the View Log option for you to review `Install.log`. If you encounter errors, view this log for information on diagnosing the failure.

JRun

Hyperion Performance Suite no longer uses JRun as an application server. Any references to JRun in the *Hyperion Performance Suite Release 8.3 Installation Guide* are now obsolete for Release 8.5.

JRun was also only installed with a serverCD install that selected servlets to install and selected JRun as the appserver. To uninstall JRUN, you must uninstall the servlets, and install them fresh. Use your Add and Remove program option to do this.

Installation Directory Structure

The Installation Directory Structure shown on page 110 of the *Hyperion Performance Suite Release 8.3 Installation Guide* has changed. Pre-release 8.5 installers did not use CIS and had no Hyperion Home.

Apache Tomcat 4.1.30

In pre-release 8.5 installs, if you chose to install tomcat, a tomcat folder was created in Install_Home. For Release 8.5 Hyperion Performance Suite uses and configures the one in Hyperion_Home, that is, in the appServers/Tomcat/4.1.30 folder.

Tomcat can only be uninstalled by way of the services installer. Prior to 8.5, tomcat was bundled with the serverCD installer and installed only if tomcat was selected, and you used the tomcat version provided by Hyperion. To uninstall, you have to uninstall the serverCD.

Often, customer install only the servlets together in a home on one machine, in which case you have to uninstall servlets, then install them fresh. Use your Add and Remove program option to do this.

2 System-Wide Changes

This chapter describes the general changes and enhancements made to Hyperion Performance Suite Release 8.5.

In This Chapter

User Defined Languages 18
User Interface Language 23
Unicode Enabling Software 27
Connectivity 36
Migration Tool 37
National Strings for MS SQL Server 39
Bidirectional Text 39
Controlling Document Versions in the Hyperion Intelligence Client 41
Row Level Security Enhancements 42
Software Developer Kit Changes 43

User Defined Languages

Previous versions of Hyperion Performance Suite binaries supported a limited set of out-of-box user interface languages, but some of them did not have corresponding translation files. You could use one of the pre-defined user interface languages, but were unable to select other user interface languages.

As of Release 8.5, any BQY Sort Order Language (also known as BQY document language) can be used as a user interface language. Dynamic loading of the user defined briottbl.txt translation files is also supported.

The following table shows user defined supported languages and includes:

- Built in — This column shows the corresponding user interface language is built into the installation and has the corresponding briottbl.txt translation file (except for English).
- Subdirectory Name — This column shows a subdirectory that can be searched in the Translation directory.
- Briottbl.txt header — This column shows a language name that must be present in the header of the corresponding briottbl.txt translation file. If the column contains several items, then any one of the listed names can be used. Also the character case is ignored. For example, you could type: “simpchinese”, “SimpChinese” or “SIMPCHINESE”.

Table 2 User-Defined User Interface Languages

UI Language Name	Built in	Subdirectory Name	Briottbl.txt Header
Afrikaans		Afrikaans	Afrikaans
Albanian		Albanian	Albanian
Arabic		Arabic	Arabic
Basque		Basque	Basque
Catalan		Catalan	Catalan
Chinese (Simplified)	Yes	Chinese	SimpChinese Chinese (Simplified)
Chinese (Traditional)	Yes	Chinesetrad	TradChinese Chinese (Traditional)
Croatian		Croatian	Croatian
Czech		Czech	Czech
Danish		Danish	Danish
Dutch		Dutch	Dutch
English	Yes	English	English
English (UK)		EnglishUK	UKEnglish
Estonian		Estonian	Estonian
Finnish		Finnish	Finnish

Table 2 User-Defined User Interface Languages *(Continued)*

UI Language Name	Built in	Subdirectory Name	Briottbl.txt Header
French	Yes	French	French
German	Yes	German	German
Greek		Greek	Greek
Hebrew		Hebrew	Hebrew
Hungarian		Hungarian	Hungarian
Icelandic		Icelandic	Icelandic
Indonesian		Indonesian	Indonesian
Italian	Yes	Italian	Italian
Japanese	Yes	Japanese	Japanese
Korean	Yes	Korean	Korean
Latvian		Latvian	Latvian
Lithuanian		Lithuanian	Lithuanian
Macedonian		Macedonian	Macedonian
Malay		Malay	Malay
Norwegian (Bokmal)		NorwegianBokmal	NorwegianBokmal Norwegian (Bokmal)
Norwegian (Nynorsk)		NorwegianNynorsk	NorwegianNynorsk Norwegian (Nynorsk)
Polish		Polish	Polish
Portuguese	Yes	Portuguese	Portuguese
Romanian		Romanian	Romanian
Russian		Russian	Russian
Serbian (Cyrillic)		SerbianCyrillic	Serbian (Cyrillic)
Serbian (Latin)		SerbianLatin	Serbian (Latin)
Slovak		Slovak	Slovak
Slovenian		Slovenian	Slovenian
Spanish	Yes	Spanish	Spanish
Swedish		Swedish	Swedish
Thai		Thai	Thai
Turkish	Yes	Turkish	Turkish
Ukrainian	Yes	Ukrainian	Ukrainian
Vietnamese		Vietnamese	Vietnamese

➤ To configure a user interface language supported by the Hyperion Performance Suite binaries but not installed automatically:

1 **Create an appropriate briottbl.txt translation file and place it into the corresponding subdirectory.**

2 **Select the user interface language (see "User Interface Language" on page 23).**

Briottbl.txt Files

The Hyperion Performance Suite binaries expect to find briottbl.txt files in the corresponding language dependant subdirectories of the Translations directory. Depending on the binary, the Translations directory resides in:

Binary	Location
BIService/Das	<BaseDirectory>\..\lib\msgs\Translations (for example, D:\hps_83_340\lib\msgs) where <BaseDirectory> is a directory specified by way of the –basedir parameter. This default value can be overridden by the TRANSLATIONS_DIR setting in the server.xml file.
Designer/Web Client	<StdAppPath>\Translations where <StdAppPath> is a directory from which the binary is inititated.

Translation directories have the following structure:

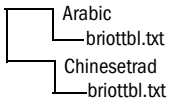
```
Translations
├─Chinese
│  └─briottbl.txt
├─Chinesetrad
│  └─briottbl.txt
├─French
│  └─briottbl.txt
├─German
│  └─briottbl.txt
├─Italian
│  └─briottbl.txt
├─Japanese
│  └─briottbl.txt
├─Korean
│  └─briottbl.txt
├─Portuguese
│  └─briottbl.txt
└─Spanish
   └─briottbl.txt
```

➤ For example, if you have an Arabic briottbl.txt translation file and need to configure an Arabic user interface language:

1 **Verify the structure of the briottbl.txt to be configured. The header must contain the correct user interface language string identifier (in this case, Arabic).**

2 **Create an Arabic subdirectory in the Translations directory and copy the briottbl.txt file into it.**

After completing the steps above, the Translations directory might look like:



User Preferences

The UILanguage property of the corresponding User Preferences defines the user interface language that is used by the Release 8.5 binaries to translate user interface messages. The value of this property must be chosen from the UI language name column. As a rule, unknown values are ignored and the default user interface language is used instead.

To add this property through the user interface, see "User Interface Language" on page 23.

Depending on the application, the UILanguage property belongs to the [Regional Settings] section of the following INI files:

Application	File Name
Designer	brioqry_ev.ini (The UI language setting property can be also set by selecting the Tools > Options > Program Options > UI Language option.)
Insight	brioqplg_ev.ini (The UI language setting property can be also set by selecting the Tools > Options > Program Options > UI Language option.)
BIService Das	brioqry_ev.ini (BIService and Das read this property at startup to retrieve the default UI language settings).

Briottbl.txt File Format

The Briottbl.txt translation file is a plain text file in the corresponding Windows ANSI encoding. The file consists of a one-line header and a set of translation lines.

The header line consists of a reserved word briolanguage (or language) followed by the TAB character, followed by UI language name. The character case is not considered, that is you can use uppercase, lowercase or a combination of both: "Hyperionlanguage", "HyperionLanguage" or "HYPERIONLANGUAGE".

The translation lines have the same structure: <string-to-translate> <TAB> <translated-string>.

Strings on either side of the TAB character can contain \r, \n, or \t escape sequences that denote 0x0D, 0x0A and 0x09 ASCII characters respectively. Invalid lines (for example, a no TAB character) are ignored.

Server.xml and Custom User Languages

The user languages displayed in the login screen and in User Preferences can be extended by modifying server.xml located on the Global Services Manager (GSM).

Below is an example of how to customize the list. A new entry must include all the entries that are bolded within the "LOCALE_LIST" propertylist tag, although the "VARIANT" node is optional and may not apply to all locales. The strings in red represent the locale and language specification. All "defid"s must be consistent with the example below.

```
<propertylist name="LOCALE_LIST" defid="000000f324d3875d-0000-114a-
0ad72214">
<propertystruct name="LOCALE" defid="000000f324d56aa3-0000-114a-
0ad72214">
    <property name="ISO_LANG_CODE" defid="000000f324d65997-0000-114a-
0ad72214">zh</property>
    <property name="ISO_COUNTRY_CODE" defid="000000f324d6f616-0000-114a-
0ad72214">CN</property>
    <property name="DEFAULT_LOCALE" defid="000000f324d778d3-0000-114a-
0ad72214">false</property>
    <property name="VARIANT" defid="000001072f707a7f-0000-08c5-
ac1b10de">simplified</property>
</propertystruct>
</propertylist>
```

The "DEFAULT_LOCALE" tells the GSM that this value is the locale that should be used as the default in the JVM.

The "VARIANT" may not apply to most locales. This value is used for different sorting routines within the same language for Hyperion Intelligence.

JSPResources.properties and Resource files

For the language list to be displayed correctly to the user, the JSPResources.properties in the "WEB-INF/classes/msgs" folder must include the corresponding language text. For each "LOCALE" propertystuct that is added, the following entries must be added into the JSPResources.properties file:

```
<Language>
<Country>
<Locale>
```

For example, to add Simplified Chinese to the language list, you need to add:

```
language.zh.string=Chinese
country.CN.string=CHINA
language.zh_CN.string=Simplified Chinese
```

The strings in red need to correspond to the server.xml entry. The last entry in the JSPResource.properties file represents the combination of the "ISO_LANGUAGE_CODE" and the "ISO_COUNTRY_CODE" that was added to the server.xml file.

Although these modifications add a new language entry into the login screen and the user preference page, they do not guarantee that you can see localized strings based on this new entry within the entire system. In order for those changes to take place, the localized resource files must exist for the new language and locale entry in server.xml, since only a few localized language sets of properties files are released.

Foundation uses the JAVA Locale representation (language_country combination) to look for the correct localized resource files. For example, if the locale is “en_US”, the entry in server.xml has “en” for the language and “US” for the country.

The servlets look for resource files named “<resourcefilename>_en_US.properties”. If the files do not exist, the servlet looks for “<resourcefilename>_en.properties”. If this file does not exist, it uses the default “<resourcefilename>.properties” and displays the string. In this case, the new language strings in the JSPResource.properties file are added for the English representation. If you want to display another language, add it to the “JSPResource_<languagecode>_<countrycode>.properties file for that language. Otherwise it defaults to the JSPResource.properties file.

User Interface Language

The localization language used for menus and messages is determined at installation for the client and Hyperion Performance Suite products. You can also set the user interface language manually.

Note: The user interface language cannot be set in the Intelligence iServer, or in the Hyperion Intelligence Web Client (available offline only).

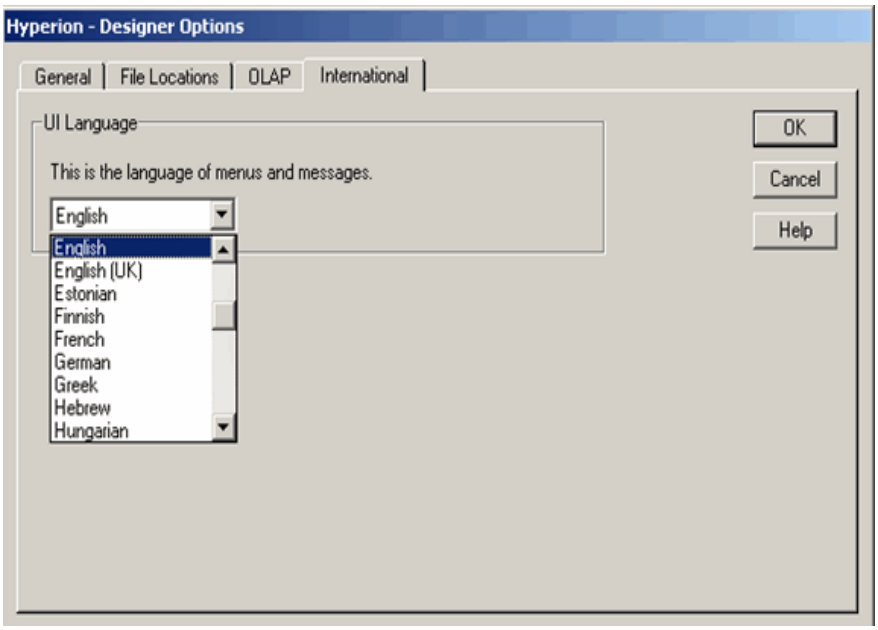
Hyperion Intelligence Designer and Web Client (Offline Only)

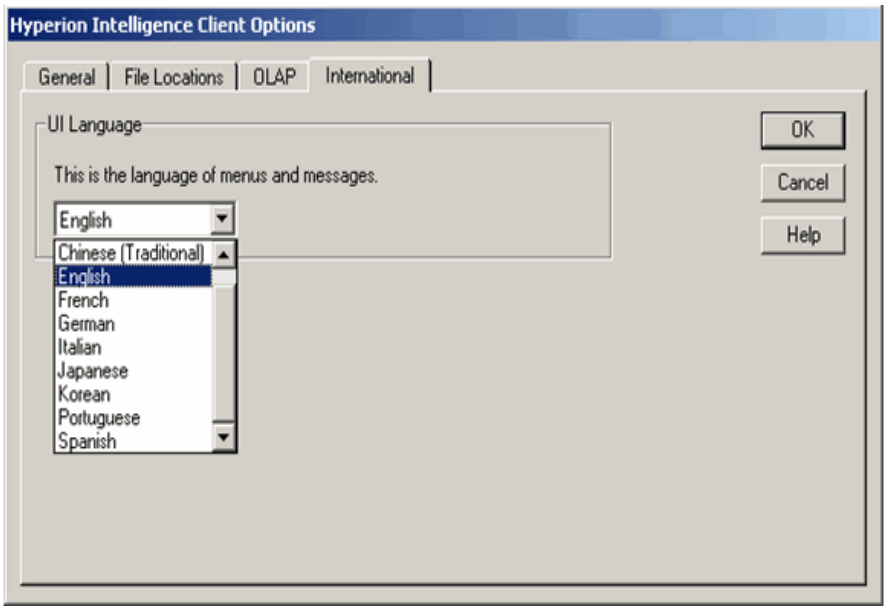
Setting the user interface language used for menus and messages is completed on the International tab of the Options dialog box.

➤ To change the user interface locale from either Designer or the Web Client:

1 Select **Tools > Program Options**.

If you are using the Designer product, the Hyperion - Designer Options dialog box is displayed. If you are using the Web Client product, the Hyperion Intelligence Client Options dialog box is displayed.





2 Select the **International** tab.

3 **Select a language from the UI Language drop-down list and click OK.**

The user interface language property is stored in the [Regional Settings] section of the config file with the following settings:

```
[Regional Settings]
UILanguage=English
```

By default, your browser locale is used to set the user language the first time you access the Release 8.5 programs. After you log on successfully, the Web Client uses the language selected from the Language drop-down list, or if no language is selected, the browser locale is used as the default. The *userlang* cookie is deposited that persists for all subsequent sessions unless the cookie is deleted.

When you open a BQY in the Web Client offline, you select the user interface language at that time. If you go on-line with the BQY, the user interface language must be the same as the one selected offline. If the user interface language of the BQY brought online differs, the new user interface language overrides the *userlang* cookie.

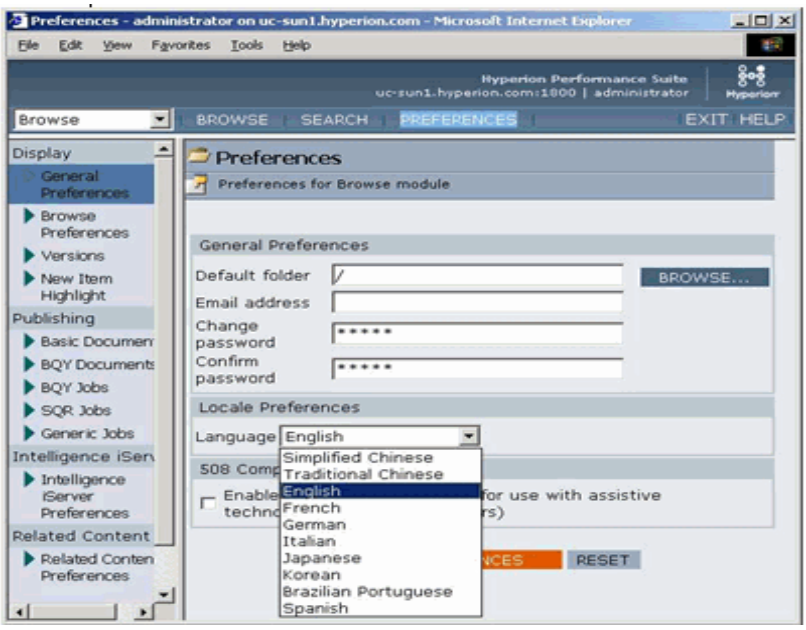
Hyperion Performance Suite Only

The localization language used for menus and messages is determined at installation for Hyperion Performance Suite.

➤ To set the user interface language from Preferences:

1 Select **Browse > Preferences.**

The Preferences window is displayed.

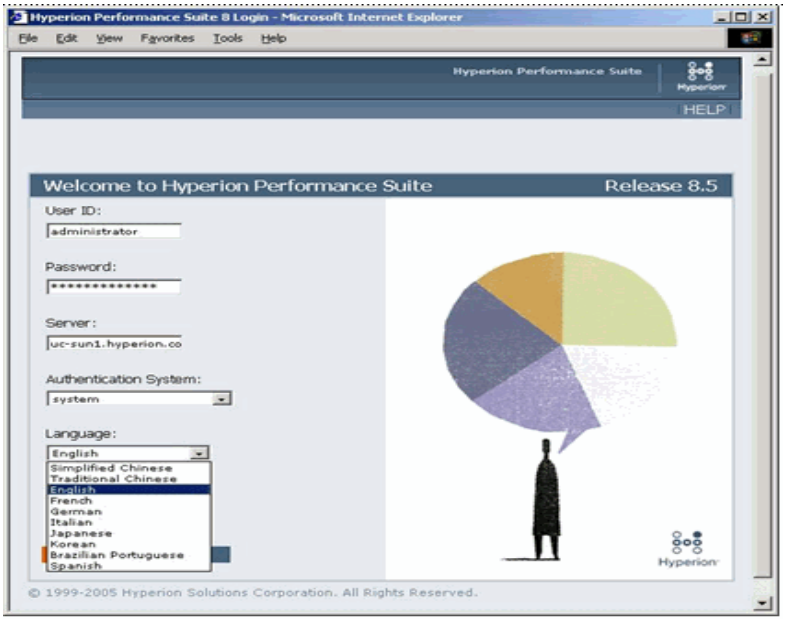


2 In the Locale Preferences section, select the user interface language from the Language drop-down list.

3 Click **Preferences.**

➤ To set the user interface language from the Advanced Logon Options, select the user interface language from the Language drop-down list and click **OK**.

By default, your browser locale is used to set the user language the first time you access the Release 8.5 programs. After you log on successfully, Hyperion Performance Suite uses the language selected from the Language drop-down list, or if no language is selected, the browser locale is used as the default. The *userlang* cookie is deposited that persists for all subsequent sessions unless the cookie is deleted.



➤ To set the user interface language when scheduling a job:

1 Select **Publish Job > Job Defaults**.

2 In the Select Locale section, select the default user interface language from the Language drop-down list.

3 Select the country associated with the default user interface language from the Country drop-down list and click **OK**.

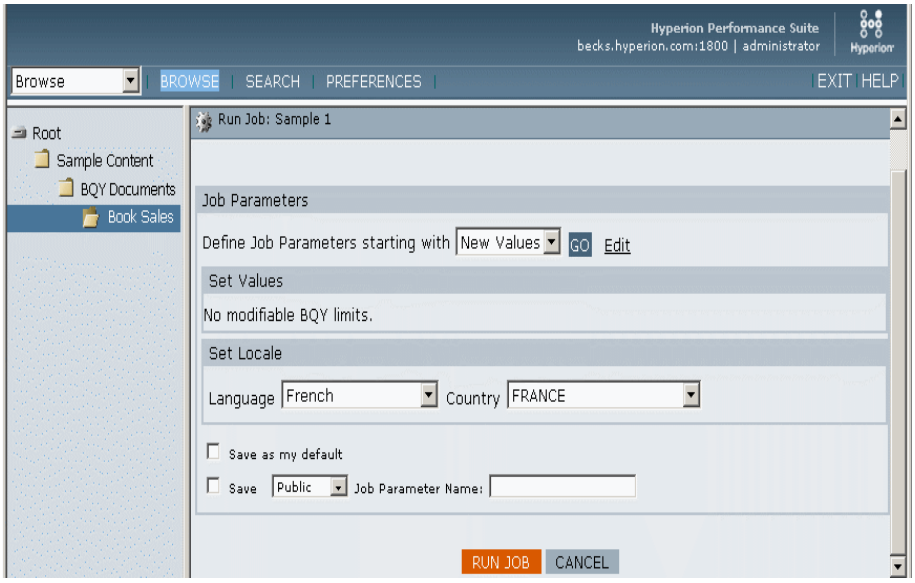
The country setting determines the data format, for example, the date/time format.

➤ To set the user interface language when running a job for each cycle:

1 From the Select Locale section of the Run Job dialog box, select the default user interface language from the Language drop-down list.

2 Select the country associated with the default user interface language from the Country drop-down list and click **Run Job**.

The country setting determines the data format, for example, the date/time format.



Unicode Enabling Software

Release 8.5 builds on the existing Hyperion Performance Suite and Hyperion Intelligence™ functionality by Unicode-enabling the software. By doing so, Hyperion Performance Suite supports deployment on all supported platforms in all supported languages. This removes previous restrictions that supported Windows-only deployments when requiring languages other than English. In addition, BQYs can be used on any system, not just those using a code page compatible with the system that created the document.

More importantly, these documents can contain data representing many languages. For example, a document created on a Japanese version of Windows can be opened through Hyperion Performance Suite running on a Korean UNIX system, displaying data simultaneously from the Turkish and Traditional Chinese character sets.

Although SQR, the SQR Developer component of Hyperion Performance Suite, already supports Unicode, it is now accessible by way of all supported platforms in any supported language.

Unicode related changes include:

- Support for localization and internationalization of UNIX products by using Unicode
- Support for connectivity to Unicode databases as well as non-Unicode databases
- Support for multi-lingual data in a BQY document
- Allowing existing BQY documents to be read in Unicode-enabled Hyperion Performance Suite. No migration effort is needed. Batch utilities are provided to convert existing BQYs to newer formats to gain performance.
- Support for multiple locales or languages within the Hyperion Intelligence Web Client™ including: Windows 2000 Server editions, Windows Server 2003, Solaris 8 and 9, AIX 5.x, HP-UX 11, and RedHat Linux 3.0
- Supported languages include: English, German, French, Portuguese (Brazilian), Spanish, Italian, Korean, Japanese, Traditional Chinese, and Simplified Chinese

This section includes these topics:

- Encoding OCE Files
- Hyperion Intelligence Application Configuration Files
- Unicode Migration Utility for BQYs and OCEs
- Unicode File Path Handling on UNIX
- Unicode Compliant Fonts
- File Import
- File Export

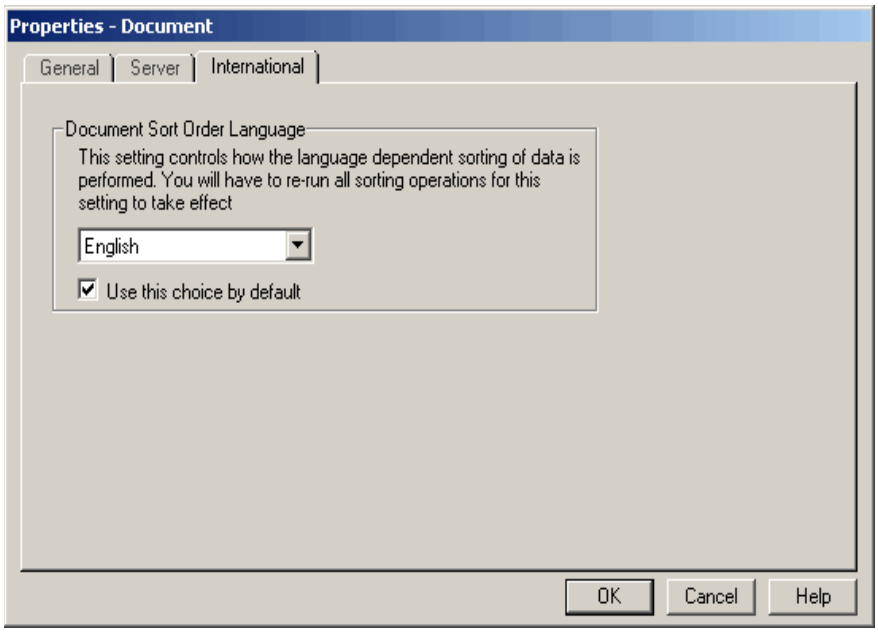
BQY Sort Order Language

When an existing BQY document is opened in Unicode-enabled Hyperion Intelligence, the document is converted from the old format to the new, and the language is determined based on the document code page. In certain locales (for example, most of the European languages) the language cannot be determined based on the code page from the existing BQY document. In this case, the user may be prompted to select the correct language for the document as shown below. When a new document is created, Hyperion Intelligence selects the language based on the user locale or language set in Windows unless the user selects another locale or language.

➤ To select the Document Sort Order Language manually:

1 **Select File > Properties.**

The Document Properties dialog box is displayed.



2 **Select the International tab.**

3 **Select the sort order language from the drop-down list.**

4 **Check the Use this choice by default box to use the sort order language as the default for new BQYs.**

5 **Click OK.**

The document sort order language property is stored in the config file with the settings:

```
[Regional Settings]
DefDocLanguage=English
```

These settings can be copied to the Hyperion Intelligence server config file.

BQYs should be converted to the new format by opening and saving them back to disk.

Encoding OCE Files

In addition to encoding your BQYs, it is recommended that you convert any existing connection file (.oce) in the Hyperion Intelligence Designer™.

➤ To convert an existing connection file:

1 **Create a new BQY using the connection file to be converted.**

The Convert OCE File dialog box is displayed.

2 **Select the language for textual data contained in the connection file from the drop-down list and click OK.**

3 **Save the connection by selecting Tools > Connection Manager > Logon.**

If a connection file is used in the Hyperion Intelligence Web Client, it is converted on the fly. In this case, the conversion language is assumed. If a connection file does not have any non-English textual data in it (which is common), the conversion is done successfully regardless of the language chosen.

Hyperion Intelligence Application Configuration Files

Hyperion Intelligence applications store some preferences in the system registry in order to allow different users to have their own personal settings. This is true for preferences contained in the Regional Settings and the Hyperion Intelligence sections which are located in registry for clients and in the config file for the Intelligence iserver.

Configuration File Locations in the registry are shown below:

Application	Registry Path
Hyperion Intelligence Designer	HKEY_CURRENT_USER\Software\Brio Software\brioqryEv
Hyperion Intelligence Web Client (Insight)	HKEY_CURRENT_USER\Software\Brio Software\BrioPluginEv

Application configuration file names have changed and are now suffixed with "_ev":

Application	Configuration File Name
Hyperion Intelligence Designer	brioqry_ev.ini
Hyperion Intelligence Web Client (Insight)	brioqplg_ev.ini
Intelligence iServer	intelligence_ev.ini

Hyperion Intelligence core configuration files have also changed:

File	Configuration File Name
Format template file	brioqry_ev.ini
Pre-defined metadata sources	briometa_ev0.ini
Control access to tables created in the Process to Table feature	bqtbls_ev5.ini
Custom Menus	bqtools_ev.ini

Unicode Migration Utility for BQYs and OCEs

You can convert all BQY and OCE files to Unicode format by using the Unicode Migration Utility. This utility enables you to convert multiple BQYs to Unicode format in batch mode. When pre-Release 8.5 BQYs are opened in the Release 8.5 Hyperion Intelligence product suite, they are converted automatically to the new Unicode format. When the documents are saved to disk, they are also stored in new format. BQYs in the Unicode format typically open faster because a conversion is unnecessary.

➤ To run the Unicode Migration Utility for BQYs:

1 **Install the Hyperion Migration Tools on a target platform.**

See *Hyperion Performance Suite, release 2.0.0 Migration Tool Installation and Usage Guide* for information on installing and running migration tools.

2 **Open a Command Prompt window.**

3 **Type: brioqry_s -cmd convert <language> -f <input files or directories> -outdir <output directories>**

- brioqry_s -cmd convert -lang — converts BQY and OCE files to unicode format. For BQYs, specify the language parameter used for the sort order stored in the output file. For OCEs, this parameter determines the encoding of the input file
- -f — specifies the name of the input files or directories to convert
- -outdir — specifies the name of the output file or directories to which the converted files are stored.

For example, if you type:

```
brioqry_s -cmd convert -lang English -f C:\My_BQYs D:\doc1.bqy -outdir
C:\Converted_Files
```

the D:\doc1.bqy file and all BQY and OCE files from the C:\My_BQYs folder are converted to Unicode format and stored in the C:\Converted_Files folder.

Unicode File Path Handling on UNIX

To ensure correct Unicode file path handling of the Intelligence iServer on UNIX, the en_US.UTF-8 locale must be available on the UNIX system.

Unicode Compliant Fonts

Release 8.5 is packaged with a set of Andale WT fonts, which facilitate the display and printing of multi-lingual texts. Both the Windows and UNIX Hyperion Performance Suite installers install these fonts automatically.

Note the behavior of the fonts:

- Andale WT font is the default product font.
- Andale WT font does not show the European Union currency symbol properly. To correct this issue, use the Andale WT J/K/CS/TC symbols.
- Andale WT font does not have a complete set of Arabic letters. Use one of the Andale WT J/K/CS/TC fonts.
- Arabic text is not shown properly with Andale WT in an internet browser. Arabic text with this font is not processed according to Arabic script rules. Use Andale WT J/K/CS/TC fonts instead.
- If you export Arabic text to a PDF file, the text is not processed according to Arabic script with any fonts.

Unicode Font Configuration

All font settings are available in the config file for easy visibility, customization, and portability across all platforms. The Default Fonts and BrioQuery settings of the config file are reserved for font configuration. The default user preference is set to Andale WT upon installation.

Unicode Compliant Font Preferences

By default Hyperion Performance Suite uses the Andale WT fonts for the display and printing of multi-lingual (Unicode compliant) text. To substitute this font with other fonts, use the new General and Dashboard tabs of the Default Fonts and Styles dialog box. Fonts definitions are set globally on the General tab. Font definitions on the Dashboard tab are set locally. When a new font is substituted, font changes are applied to new sections only and not the original sections. Also, you need to save and restart the application to apply the new settings.

Setting General Font Preferences

Use the General tab to select the text fonts displayed in a dialog box, fixed and variable widths.

➤ To apply a font substitute:

1 **Select Tools > Options > Default Options.**

The Default Fonts and Styles dialog box is displayed.

2 **Select a font category to change from the Settings list.**

- Dialog Font — Defines the font used for text in dialog boxes. Depending on the selected font, you can also select the font size from the Size drop-down box.
- Variable Width Font — Defines the font used for variable width fonts (such as Arial). Variable width fonts use characters of different widths.
- Fixed Width Font — Defines the font used for fixed with fonts (such as Courier). Fixed width fonts use characters of equal widths.

3 **Select the substitute font from the Font Name drop-down list.**

The following fonts are available for substitution:

4 **Click OK.**

To revert to the default settings for fonts defined at installation, click Defaults.

5 **Save the document and restart the application.**

Setting Dashboard Font Preferences

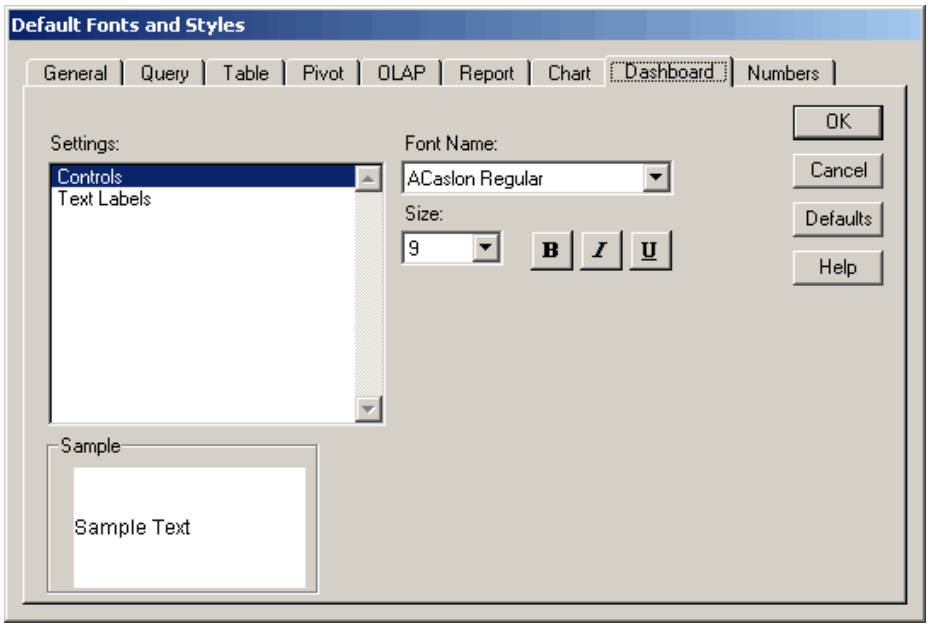
Use the Dashboard tab to select the text fonts properties to be substituted in controls and text labels. For example, if you are working with an older BQY that uses an Arial font for text labels and contains Chinese data, text may not show up correctly on a Korean operating system (Win2K). To fix the display, you can specify an Arial font substitute with more Unicode compliant characters on the Dashboard tab.

To revert to the default settings for controls and text labels, click “Defaults”.

➤ To set a font substitute for a control or text label:

1 **Select Tools > Options > Default Options.**

The Default Fonts and Styles dialog box is displayed.



2 Select Controls or Text Labels from the **Settings** list.

3 Select the substitute font from the **Font Name** drop-down list.

4 Select the font size from the Size drop-down list.

5 Select a font format.

Valid font format options are:

- B — bolds the font
- I — italicizes the font
- U — underlines the font.

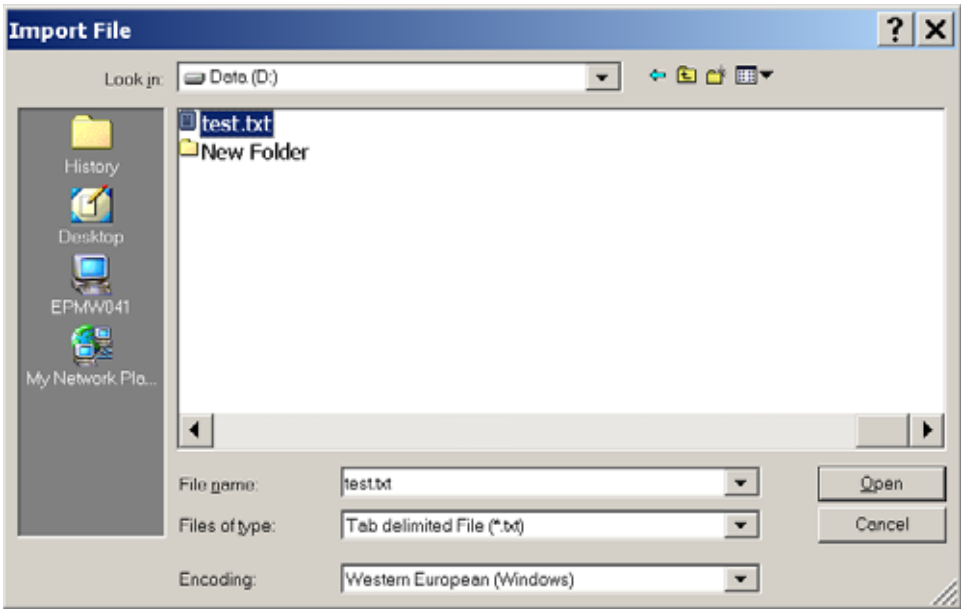
6 Click **OK**.

To revert to the default settings for fonts defined at installation, click Defaults.

7 Save the document and restart the application.

File Import

When importing a text file from disk, you can encode the file.



If the file has an initial special Unicode byte order mark (BOM), the file is recognized. Otherwise, the encoding can be set using the Encoding drop-down.

In addition, the following Hyperion Intelligence Object Model methods accept an additional parameter that sets the input file encoding:

```
ImportDataFile(String Filename, BqImportDataFileFormat Format, BqEncoding
Encoding)
ImportSQLFile(String Filename, int numColumns, BqEncoding Encoding)
LoadFromFile(String Filename, BqEncoding Encoding)
```

BqEncoding enumeration has the following values:

- bqEnc_Unicode_UTF8 — Unicode with UTF-8 encoding
- bqEnc_Unicode_UTF16 — 16-bit Unicode – operating system default endian
- bqEnc_Unicode_UTF16_BigEndian — 16-bit Unicode, most significant byte comes first
- bqEnc_Unicode_UTF16_LittleEndian — 16-bit Unicode, least significant byte comes first
- bqEnc_Unicode_UTF32 — 32-bit Unicode – operating system default endianess
- bqEnc_Unicode_UTF32_BigEndian — 32-bit Unicode, most significant byte comes first
- bqEnc_Unicode_UTF32_LittleEndian — 32-bit Unicode, least significant byte comes first
- bqEnc_Arabic_ISO — Arabic languages, ISO encoding (iso-8859-6)
- bqEnc_Arabic_Windows — Arabic languages, windows encoding (codepage 1256)
- bqEnc_Baltic_ISO — Baltic languages, ISO encoding (iso-8859-4)
- bqEnc_Baltic_Windows — Baltic languages, windows encoding (codepage 1257)

- bqEnc_CentralEuropean_ISO — Central European languages, ISO encoding (iso-8859-2)
- bqEnc_CentralEuropean_Windows — Central European languages, windows encoding (codepage 1250)
- bqEnc_Chinese_GBK — Chinese language, GBK encoding
- bqEnc_Chinese_GB2312 — Chinese language, GB2312 encoding
- bqEnc_Chinese_HZ — Chinese language, HZ-GB-2312 encoding
- bqEnc_Chinese_BIG5 — Chinese language, BIG5 encoding
- bqEnc_Cyrillic_DOS — Cyrillic languages, OEM encoding (windows codepage 866)
- bqEnc_Cyrillic_ISO — Cyrillic languages, ISO encoding (iso-8859-5)
- bqEnc_Cyrillic_KOI8R — Cyrillic languages, encoding KOI-8R
- bqEnc_Cyrillic_KOI8U — Cyrillic languages, encoding KOI-8U
- bqEnc_Cyrillic_Windows — Cyrillic languages, windows encoding (codepage 1251)
- bqEnc_Greek_ISO — Greek language, ISO encoding (iso-8859-7)
- bqEnc_Greek_Windows — Greek language, windows encoding (codepage 1253)
- bqEnc_Hebrew_Windows — Hebrew language, windows encoding (codepage 1255)
- bqEnc_Japanese_EUC — EUC - Japanese
- bqEnc_Japanese_JIS — Japanese, JIS
- bqEnc_Japanese_ShiftJIS — Japanese, Shift-JIS
- bqEnc_Korean — Korean (Unified Hangul Code)
- bqEnc_Thai_Windows, Thai language, windows encoding (codepage 874)
- bqEnc_Turkish_ISO — Turkish language, ISO encoding (iso-8859-2)
- bqEnc_Turkish_Windows — Turkish language, windows encoding (codepage 1254)
- bqEnc_WesternEuropean_ISO — Western European languages, ISO encoding (iso-8859-9)
- bqEnc_WesternEuropean_Windows — Western European languages, windows encoding (codepage 1252)
- bqEnc_Vietnamese_Windows — Vietnamese language, windows encoding (codepage 1258)

File Export

When exporting a BQY document or section to text or a Microsoft Excel (XLS file), the user can encode the file in the Save File dialog box, or supply optional encoding parameters in the Object Model as shown below. When the encoding parameter is not specified, the default encoding is used, which is determined from the context.

```
Export(Filename, BqExportFileFormat FileFormat, IncludeHeaders, Prompt,
BqEncoding Encoding);
ExportToStream([Filename, BqExportFileFormat FileFormat, IncludeHeaders,
DataStreaming, Prompt, BqEncoding Encoding);
```

Note: For BQY jobs, text files are exported in UTF-8 encoding. Microsoft Excel files are exported in default encoding that is determined from the context.

The TXT and CSV files can be exported using all available values of BqEncoding enumeration, but only these values are admitted in the Export to Excel (XLS) feature:

- bqEnc_Arabic_Windows
- bqEnc_Baltic_Windows
- bqEnc_CentralEuropean_Windows
- bqEnc_Chinese_GBK
- bqEnc_Chinese_BIG5
- bqEnc_Cyrillic_Windows
- bqEnc_Greek_Windows
- bqEnc_Hebrew_Windows
- bqEnc_Japanese_ShiftJIS
- bqEnc_Korean
- bqEnc_Thai_Windows
- bqEnc_Turkish_Windows
- bqEnc_WesternEuropean_Windows
- bqEnc_Vietnamese_Windows

Connectivity

Release 8.5 includes several connectivity changes.

DB Connectivity

Unicode enabling of the software affects the Oracle SQL* Net Connection version and DataDirect Oracle Driver.

Oracle SQL*Net Connection

The Oracle client must use a 9.2.0.5 or higher version.

DataDirect Oracle Driver

Hyperion Performance Suite installs DataDirect v5.1 drivers. For the DataDirect Oracle driver the Enable N-CHAR Support setting should be checked on the ODBC Oracle Wire Protocol Driver Setup.

Informix Connectivity

Informix connectivity is supported using Data Direct ODBC drivers and Data Direct ODBC driver manager.

Open Client Connectivity

Open Client connectivity for Sybase is not supported in Release 8.5.

Red Brick and Teradata

Redbrick and Teradata connectivity are not supported in Release 8.5.

Migration Tool

When migrating a BQY or OCE to Unicode format, the docLanguage and userDefLanguage parameters need to configured manually in the common. ini file.

docLanguage

The docLanguage=<Language_Name> sets the sort order language for BQY files that are converted to Unicode format during the migration process. For OCE files, this parameter sets the encoding for data conversions.

For example, to set the sort order language to English, type:

```
docLanguage=English
```

You can retrieve a list of language constants by running the Migration Tool utility:

```
brioqry_s -cmd languages
```

Below is the list of eligible sort order languages that can be retrieved from the utility:

- Afrikaans
- Albanian
- Arabic
- Basque
- Catalan
- Chinese (Simplified)
- Chinese (Traditional)
- Chinese (GB2312 Sort)
- Chinese (Stroke Sort)
- Croatian
- Czech
- Danish
- Dutch

- English
- English (UK)
- Estonian
- Finnish
- French
- German
- German (Phonebook Sort)
- Greek
- Hebrew
- Hungarian
- Icelandic
- Indonesian
- Italian
- Japanese
- Korean
- Latvian
- Lithuanian
- Macedonian
- Malay
- Norwegian (Bokmal)
- Norwegian (Nynorsk)
- Polish
- Portuguese
- Romanian
- Russian
- Serbian (Cyrillic)
- Serbian (Latin)
- Slovak
- Slovenian
- Spanish
- Spanish (Traditional Sort)
- Swedish
- Thai
- Turkish
- Ukrainian
- Vietnamese

userDefLanguage

The userDefLanguage=<Language_Code> parameter sets the default language for Hyperion Performance Suite portal users that are created during the user migration process. The language constants include:

- Simplified Chinese: zh_CN
- Traditional Chinese: zh_TW
- English: en_US
- French: fr_FR
- German: de_DE
- Italian: it_IT
- Japanese: ja_JP
- Korean: ko_KR
- Brazilian Portuguese: pt_BR
- Spanish: es_ES

For example, to set the userDefLanguage to English, type:

```
userDefLanguage=en_US
```

National Strings for MS SQL Server

National strings (for example, ntext, nchar, and nvarchar data types) used in SQL statements against a MS SQL server must be prefixed with N. For example, you would enter:

```
N'str丕東'
```

This rule applies for national strings used for Custom SQL, Computed Items, Stored Procedures input parameters, and Row Level Security.

Bidirectional Text

Bidirectional text is text containing both right-to-left and left-to-right directional runs and it is now supported in Hyperion Performance Suite. Bidirectional text support is available on all platforms, in the user interface and in all exported documents:

- Export to HTML
- Office HTML
- Export to PDF

Bidirectional text is rendered equally in all possible output modes, but with the following restrictions:

- Ligatures are not placed between Arabic letters if the text is exported to a PDF file. A ligature is a sequence of characters displayed as a single unit replacing characters that occur next to each other when common components are shared.

- In Office HTML Export, the bidirectional processing is performed by Microsoft Excel after the exported document is opened. Due to this behavior, text rendering may differ from what is presented in the Designer or Intelligence iServer.
- Release 8.5 does not change the system-defined bidirectional behavior of the following user interface controls: edit, drop-down, and list-box. This is true both for clients (Designer and Intelligence Client) and the Intelligence iServer.

Understanding Bidirectional Text

Each language has its writing system, or script. Script includes the set of characters which are used for writing. Most languages can be viewed in a left-to-right (LTR) direction of the text, which means that writing begins from left-hand side of page, and concludes at the right-hand side. However there are scripts which have a right-to-left (RTL) direction, for example, Arabic and Hebrew.

Since both RTL and LTR directional text can be displayed within the same paragraph, these paragraphs are referred to as "bidirectional." For example in the Hebrew text below, the text is written in the RTL direction, and numbers are LTR directional.

הרב שלמה אבינר נולד בשנת 1943 בעיר ליון בצרפת.

You read this sentence starting from the right side beginning with:

הרב שלמה אבינר נולד בשנת

next reading number 1943, then reading:

בעיר ליון בצרפת.

Complex structures occur when a paragraph of one direction has a citation of different directional text. This citation can contain other citations, which can have original text direction and so on. Consequently text of different directions can be enclosed several times.

The part of the text which has the same direction is called a run.

Bidirectional text can be ordered in visual or logical order. When characters are ordered in the same way in which they are displayed - from left to right, it is a visual order. Entering or reading bidirectional text is done in logical order, which is the sequencing of characters ordered on the way in which characters should be read. Some text scripts require that characters be combined into one ligature when output.

Bidirectional Configuration Settings

Bidirectional text properties can be set in the application configuration file. On UNIX, processing of bidirectional text is enabled by default. On Windows, bidirectional text support is performed when complex script support is enabled in the system. Additionally, bidirectional support is disabled by using the setting:

```
[Regional Settings]
BidiEnabled=0
```

All paragraphs are left-to-right directional by default, but this setting can be modified by using the setting:

```
[Regional Settings]
TextDirection=LTR|RTL|Context
```

All paragraphs have a left to right reading order if the setting is LTR, or a right to left reading order if the setting is RTL. Context refers to the paragraph direction based on the first letter of paragraph.

Controlling Document Versions in the Hyperion Intelligence Client

Automatic Document Refresh (ADR) enables you to merge and synchronize a locally saved BQY as soon as you connect to the repository with the latest version of the repository document. This feature is only available for documents opened for the first time.

When a locally saved document is opened, a connection dialog for the connection file (.oce) in the repository prompts the user to connect to the repository. The version information in the BQY is compared with the current version information for the document in the repository. If ADR has been enabled for the document, you are prompted to update to the latest version. At this point, you can proceed with the refresh or not. After the document has been refreshed, the refreshed data cannot be undone. If the flag *ADR* is disabled for the document, the document refresh feature is unavailable and the locally saved document is opened as is.

The ADR synchronizing procedure is controlled at the system and document levels where ADR always refreshes the whole document. All documents are published and stored in the repository with ADR control flags enabled or disabled. Unlike the Designer version, there is no concept of Model Type.

ADR Control Flags

Control flags determine if a BQY is eligible for ADR. These flags include:

- **ADR Global Flag**—This flag controls the availability of the ADR feature. For a new installation of Hyperion Intelligence, this flag defaults to enabled. For an upgrade installation, this flag is disabled. Your system administration can enable or disable this feature as needed.
- **ADR BQY Metadata**—This flag is enabled or disabled when a BQY is published to the repository. If the flag is enabled, this particular document is allowed only for ADR. For a simple ADR, this flag is always enabled. ADR for job output defaults to a disabled flag when a BQY is published by a job action. In this case, enable this flag by modifying the properties of the BQY. This flag is always disabled for a job output collection.

The following table shows how ADR behaves with documents in different scenarios:

BQY section in Repository version	Section in local BQY	Action in Merged BQY
Section does not exist	Section does not exist	No action
Section does not exist	Section exists	Add from local document
Section exists	Section does not exist	Add from Repository document
Section exists	Section exists	Write Repository version

Row Level Security Enhancements

Two new features have been added to Row Level Security. The first enhancement controls the type of logic used between groups when a user belongs to more than one. The second allows the security administrator to specify the type of join between tables.

OR Logic Between Groups

Each permission or restriction applied to the same group or user is separated by AND logic. For users who belong to multiple groups, these sets of permissions or restrictions can be separated optionally by OR conditions or by AND conditions. This option makes the logic disjunctive or conjunctive respectively. For example, a user who is in group 'A' is allowed to see sales data in the "truck" category for the Eastern Region. A user who is in group 'B' is allowed to see sales data in the 'minivan' category for the Western Region. A user who belongs to both group 'A' and group 'B' can see both 'truck' and 'minivan' data in their respective sales regions from one query using OR Logic. If conjunctive logic were used, that user would see no data because the category could not be 'truck' and 'minivan' simultaneously.

The type of logic applied is specified by a new table, BRIOSECP. This table contains one column, named BCONJUNC, and it can contain either the literal value "AND" or "OR", depending on the requirements of the security implementation.

JOINOPER Operator

Row Level Security rules now have the JOINOPER column in the BRIOSECR table, which is used when Row Level Security SQL is generated, and those restrictions specify a join table. The column can contain one of the following string values: =, <>, <, >, <=, >=, as well as the outer join options FULL, LEFT, and RIGHT. If no value is given, when the join table is added to the SQL, an equijoin (=) is chosen.

If the join table is already joined to the source table in the data model, the join type specified in the data model takes precedence.

Software Developer Kit Changes

Release 8.5 includes the following Software Developer Kit (SDK) enhancements:

- Support has been added to the SDK to log in to a CSS authentication system using CSS token and log in to Security Agent protected environment.
- A super user can retrieve the Get Scheduled Task by its name and owner even if the owner has changed. A non super user can retrieve the task if he or she is owner for the task. If a non super user tries to get a task in which the owner is another user, access is denied and an exception occurs.
- The SDK and the user interface have been synchronized to prevent the setting of an adaptive state role other than “View Only”.
- The Publisher Service has been corrected to prevent the change of the current active category during BQY document publication.

3 Hyperion Foundation Enhancements

This chapter lists the changes and enhancements made to Hyperion Foundation for Release 8.5.

In This Chapter

Alerting for Relational Queries and Reporting 46
Job Content (Cascade) Deletion 46
Job Queuing 46
Native Authentication Passwords 49
Creating a Mini-Administrator 49
Triggering Events 49

Alerting for Relational Queries and Reporting

If a BQY job generates an exception that alerts users to job results that fall outside scripted parameters, an exception can be written to the user's personal page. In this case, the BQY job generates an output.properties text file, an exception flag is set, and the stoplight icon turns red on the user's personal page. In addition, any exceptions in the system can be displayed in a Dashboard as a traffic light. The traffic light shows red when the exception is raised or green when it is not. To enable this option, check the "If exceptions are generated, allow users to add to their Exceptions Dashboard" field in the Advanced Option dialog box.

Job Content (Cascade) Deletion

Job content deletion is now streamlined. It is only necessary to delete the job, checking a box to delete any outputs along with the job itself. In the past, all outputs would have to be deleted prior to deleting the job.

BQY Job Log

A scheduled BQY job now generates a HTML log file rather than a text (.TXT) file.

Job Queuing

Administrators can now control the number of concurrent jobs processed by the Job Service by increasing or decreasing the limit on the number of concurrent jobs, without restarting the services. Administrators can also disable the Job Service from processing job for a defined time interval. To accommodate this change, enhancements have been made to:

- Scheduled Jobs
- Background Jobs
- Foreground Jobs
- Disabling the Job Service
- Job Limits
- Disabling the Job Service
- Acquiring Jobs
- Jobs Only Intelligence Service

Scheduled Jobs

Scheduled jobs are queued when all job services are processing the maximum concurrent jobs defined. The queue is maintained in the Event Service. Schedules in the queue are sorted based on the priority of the schedule and the order in which they are triggered.

When a schedule is ready to be processed, the Event Service builds the job and submits it to the Service Broker. The Service Broker gets the list of all the Job Services that can process the job and checks the availability of the Job Service based on the number of concurrent jobs being processed by that Job Service. This information is obtained dynamically from each Job Service.

If the Service Broker cannot find a Job Service to process this job, a Job Limit Reached exception is occurs. Queuing is now enabled in Event Service and the schedule is added to the queue and the job data required to pick the Job Service is cached. This job data includes the application and executable information for the job.

When the next schedule is ready to be processed, the job is built and it is determined if the type of job is already in the queue (based on the job data cached). If the job is present, it is added to the queue. If not, the job is submitted to the Service Broker for processing.

When queuing is enabled in Event Service, a new Job Service polling thread is initialized. This thread checks for available Job Services. If a Job Service is available, the first schedule that can be processed by this Job Service is processed. The job data cached in the Event Service is used to determine if a Job Service can process the schedule. The job data of a schedule is removed from the cache after the schedule is submitted to the Job Service. If the job properties are modified, the new properties are used only if the changes were made after the schedule is fired and is added to the queue.

Background Jobs

If a Job Service is not available to process a background job, a command is issued to the Event Server to create a schedule with a custom event to run at that time. This command persists the schedule information in the database. The schedule uses the job parameter that is associated with the background job, and it is processed as a normal schedule job in the Event Services.

Previously, if a job limit was reached when a background job was submitted directly to the Job Service, the job execution failed.

Foreground Jobs

If a Job Service is not available to process a foreground job, an exception occurs notifying the user that the Job Service is busy. The user is also prompted with an option of queuing the job for being processed with the next available Job Service. If the user adds it to the queue, a schedule is created with a custom event to run at the time. The schedule is processed as a normal schedule job in the Event Service. The schedule and the event are deleted after the job is submitted to the Job Service.

Disabling the Job Service

The HOLD property on the new Dynamic tab of the Properties of Job Service dialog box enables the user to place the Job Service on hold. When the Job Service is on hold, no new job is processed, but the Job Service continues to process any job that is already running. This property can be changed without restarting the Job Service.

Job Limits

The JOB_LIMIT property on the new Dynamic tab of the Properties of Job Service dialog box enables an administrator to modify the number of concurrent jobs that can be run by a Job Service. The job limit can be modified at runtime. When the job limit has been modified, Job Service properties are updated by way of the Name Service, which in turn notifies the Job Service about the change in the job limit. This property can be changed without restarting the Job Service.

Acquiring Jobs

The ACQUIRE_ONLY_JOB-ONLY_BI_SERVICES property has been added to the new Dynamic tab of the Properties of Job Service dialog box. When this property is enabled, the Job Service acquires Intelligence Services that are configured to process only jobs. If this property is disabled, the Job Service acquires Intelligence Services that processes both BQY jobs and interactive requests (including publishing BQY files and jobs, modifying BQY files and jobs). The default value for this property is disabled. When this property is enabled for Job Service, make sure that there is at least one Intelligence Service capable of handling interactive requests, and at least one capable of handling BQY Jobs only.

Jobs Only Intelligence Service

The Intelligence Service can be flagged to process only BQY jobs by adding the SERVICE_ROLE property with a value of JOB_REQUEST_PROCESSOR to the server.xml file in the <BrioPlatform>/etc folder. For example, type:

```
<property defid="0ad70321-0001-08aa-000000e738090110"
name="SERVICE_ROLE">JOB_REQUEST_PROCESSOR</property>
```

The Default value for SERVICE_ROLE is GENERAL_REQUEST_PROCESSOR.

The Intelligence Service can be configured to limit the number of concurrent BQY jobs when it is flagged to run only jobs. To limit the number of concurrent BQY jobs, add the MAX_CONCURRENT_JOB_REQUESTS property to the server.xml file in the <BrioPlatform>/etc folder.

In the example below, Intelligence Service has been set to run up to ten concurrent jobs:

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

The Default value for the MAX_CONCURRENT_JOBS property is unlimited.

At least one Intelligence Service must be configured with the default service role (GENERAL_REQUEST_PROCESSOR) to handle:

- Interactive requests including publishing and modifying BQY jobs and documents
- All requests from the .ihtml client

If you use the JOB_REQUEST_PROCESSOR role for the Intelligence Service, Job Services should be configured with the ACQUIRE_ONLY_JOB-ONLY_BI_SERVICES property.

Native Authentication Passwords

The Native Authentication feature only recognizes the first eight characters of the password selected on the New Native Authentication Password System dialog box and ignores any character greater than eight characters. This limitation is enforced despite the thirty-two character length of the Trusted Password field.

Creating a Mini-Administrator

A mini-administrator role must be given explicit Modify and Full Control access to each user name in a group to configure user accounts. If not, a mini-administrator has access to named objects only.

Triggering Events

TriggerExternalEvent.java has been added to the SDK sample.

TriggerExternalEvent.java Sample API

Use this sample to trigger an external event. When this event is triggered, it is set to run immediately. Externally triggered events are always PUBLIC events.

Input parameters: `acct pwd host port Ext Event Name`

Output: External event named in the control file is triggered.

4 Hyperion SQR Enhancements

This chapter lists the changes and enhancements made to Hyperion SQR for Release 8.5.

In This Chapter

Secure Report Bursting 52
Functional and Platform Improvements to SAP BW Reporting 53
Enhancements to SQR Developer 59
New Excel Output Type 78
New Charting Functionality 78
UFUNC Unicode Support 82
Support for New Encoding 82
Ability to Call C++ Libraries from an SQR Program 82
Ability to Add a Last Page Value to a Bar Code 84
New Paper Size in DECLARE-LAYOUT 85

Secure Report Bursting

SQR 8.5 provides secured report bursting to enable security and access control at the report section level of large enterprise reports. Secure report bursting personalizes content for each user, ensuring they see only the business information for which they are authorized.

For example, a section detailing the performance of one department may only be accessible to its manager while being restricted from other managers. Moreover, any table of contents, links, and other navigational aids are also automatically adjusted to only include the accessible sections of the report.

Release 8.5 includes support for secure report bursting across all SQR supported output types - including HTML, PDF, BQD, CSV, XLS, SPF, PS, LP, and HP – allowing the report user to view and print his/her report specific information.

The implementation of secured report bursting consists of two parts.

- First, inside SQR programs the SECURITY command is used to define the sections and their access control lists
- Second, when output from this SQR program or job is published, the Foundation Authorization engine manages the distribution of the secured sections of the reports according to user or group access controls.

An example of the SQR syntax for creating a secure report below:

```
Begin-Report
Security Mode 'On' Set=('Directors', 'Vice-Presidents')
! Only Directors and VPS can see this
Security Mode='On' Remove=('Directors')
! Only VPs can see this
Security Mode='Off'
! Anybody can see this
Security Mode='On' Append=('Managers')
! Only VPs and Managers can see this
Security Mode='On' Append=('Engineers')
! Only VPs, Managers, and Engineers can see this
End-Report
```

Note that secure bursting is useful only when used in conjunction with the Hyperion Performance Suite Foundation. The Security IDs refer to groups. To have the Security ID refer to a specific user, prefix it with U#. For example:

```
sales, marketing, u#King
```

refers to the sales group, the marketing group, and the user *King*.

SQR secure report bursting significantly reduces the administrative burden by minimizing the number of reports that need to be created and managed, with the added benefit of offering personalized views of information.

For more information refer to the *Hyperion Foundation User's Guide* and to the *Hyperion SQR Language Reference.*

Functional and Platform Improvements to SAP BW Reporting

Functional and platform improvements to SAP BW reporting include:

- Increased Platform Support of SAP BW to 3.0B, 3.1 and 3.5
- Certification for SAP NetWeaver (OLAP BAPIs)
- Data Cursoring for High Volumes of Data Processing
- Access to ODS Objects, InfoSets, and SAP Variables
- Currency Conversion
- Unit of Measure Conversion
- ANCESTOR and DESCENDANT Clauses
- Support for Key Figure Attributes – Currency, Formatted, Unit, Value Type

Increased Platform Support of SAP BW to 3.0B, 3.1 and 3.5

Release 8.5 supports SAP BW platforms up to and including 3.0B, 3.1, and 3.5.

Certification for SAP NetWeaver (OLAP BAPIs)

Hyperion SQR has been certified for integration with the OLAP BAPI interface as part of the SAP® NetWeaver Business Intelligence (SAP NetWeaver BI), which includes SAP Business Information Warehouse (SAP BW). This certification helps ensure that customers using the mySAP™ business Suite family of business applications and the SAP NetWeaver platform can further leverage Hyperion SQR to extend the value of their solutions.

Hyperion SQR is also certified by SAP for integration with the mySAP™ Business Suite.

Data Cursoring for High Volumes of Data Processing

In Release 8.5, the SAP interface provides the means to access selected rows from large data sources by allowing the request to include the start and end row numbers. You can also access restricted members of a dimension or hierarchy with the use of an input parameter. The `getChildren` function in the DDO SAP BW driver allows you to enter starting and ending row numbers. The `getChildrenContains` function allows you to enter specific values, low and high range values, or wildcard values.

The `getChildrenContains` function accepts the following input parameters:

- Cube—Cube name
- Dimension—Dimension member
- Parameter Array—Array of the following elements:
  - Low Value—Low member name value
  - High Value—High member name value
  - Option Value—Operator in select options

(including =, !=, <, >, <=, >=, BT (Between), CP(Contains))

- Sign—Inclusion or Exclusion field ('I' or 'E')

For example:

```
getChildrenContains (0APO_DS01,0APO_RTYPE,ParameterArray)
```

```
ParamterArray{D,,=,E}
```
returns all 0APO_RTYPE except D

```
ParamterArray{,D,H,=,I}
```
returns all 0APO_RTYPE between D and H

```
ParameterArray{D,H,<=,I}
```
returns all 0APO_RTYPE less than and equal to D

```
getChildrenContains (0APO_DS01,0APO_CUSTNO,3000,3999,BT,I)
```

```
ParameterArray{3000,3999,Between,I}
```
returns all 0APO_CUSTNO between 3000 and 3999

```
ParameterArray{3+00, ,Contains,I}
```
returns all 0APO_CUSTNO that begin with 3 and ends with 00

Note: See "Restricting the Members Retrieved" on page 68 for information on how to use SQR Developer to restrict the members retrieved from a query.

Access to ODS Objects, InfoSets, and SAP Variables

Figure 1 shows the hierarchical structure of the objects for an SAP BW data source. Note that you can now access ODS Objects, InfoSets, and SAP Variables.

Figure 1 Hierarchical Structure of Objects for an SAP BW Data Source

Information Providers
-----InfoCubes
-----QueryCubes
-----ODS Objects
-----InfoSets
---------Characteristics
----------CharacteristicLevels
----------CharacteristicLevelMembers
----------CharacteristicChildMembers
----------CharacteristicMembers
----------CharacteristicChildMembers
----------Properties (Optional)
----------Property
----------PropertyMembers
----------MandatoryProperty
----------Hierarchy
----------HierarchyLevels

```
-------------------------------------HierarchyLevelMembers
-----------------------------------------------HierarchyChildMembers
-------------------------HierarchyMembers
-------------------------------------HierarchyChildMembers
-------------------------Properties (Optional)
-------------------------------------Mandatory Properties
---------Key Figures
----------------Key Figure Attributes
---------SAP Variables (QueryCubes only)
----------------SAPVariableMembers
```

Note: See Step 1 under "Selecting an SAP BW Object" on page 64 for information on how to display ODS Objects and InfoSets in the SQR Developer Query Builder. See "Defining SAP Variables" on page 74 for information on how to define SAP Variables in the SQR Developer Query Builder.

Currency Conversion

Note: The syntax in this section refers to *key figures* as *measures*.

In Release 8.5, you can convert returned measures items to other currency types. To support this capability, the DDO SAP BW driver accesses the Currency Rates table maintained in an SAP BW data source. This table contains the `TO` and `FROM` currency types and the rate of conversion for each available rate type. Additionally, conversion ratio factors are maintained.

A new property value in the Data Source Property sheet specifies the Currency Conversion requested.

```
MDSelector.CurrencyConv.Name=MD Selector Currency Conversion Property
MDSelector. CurrencyConv.Description=Specify the Currency Conversion
requested.
MDSelector. CurrencyConv.ClassName=java.lang.String
```

The `CurrencyConv` property consists of a four part construct. Each construct is enclosed within a pair of braces ("{" and "}"). Construct elements include:

- Name of the SAP Key Figure (Measure)

  The name must be enclosed in quotes.
- `TO Currency` value
- Rate type
- (Optional) Effective date (`MM/DD/YYYY`)

For example:

The following converts `Measures.33L2S0QRHTBOKSWAQZAENHX2Z` to USD and `Measures.0HPW6SU8X9M7WBGRCD1PWI471` to EUR for rate type 'M' with an effective date of Jan 01, 2005.

```
MDSelector. CurrencyConv =
{"Measures.33L2S0QRHTBOKSWAQZAENHX2Z",USD,}
{"Measures.0HPW6SU8X9M7WBGRCD1PWI471",M,EUR,01/01/2005}
```

Note: See "Defining Currency Parameters for Key Figures" on page 69 for information on how to use SQR Developer to define currency conversion parameters for key figures in a query.

Unit of Measure Conversion

Note: The syntax in this section refers to *key figures* as *measures*.

In Release 8.5, you can convert returned measures items to other units of measure types. To support this capability, the DDO SAP BW driver accesses the Alternate Material Units Rates table maintained in an SAP BW data source. This table contains the `TO` and `FROM` material unit types and the rate of conversion.

The following formula determines the conversion factor from the base unit of measure to an alternate unit of measure.

```
Numerator (Base UOM)/Denominator (Base UOM)
Conversion Factor = ---------------------------
Numerator (Alternate UOM)/Denominator (Alternate UOM)
Alternate Quantity Value = Base Quantity Value * Conversion Factor
```

A new property value in the Data Source Property sheet specifies the Units of Measure conversion requested.

```
MDSelector.UOMConv.Name=MD Selector Units of Measure Conversion Property
MDSelector. UOMConv.Description=Specify the Units of Measure Conversion
requested.
MDSelector. UOMConv.ClassName=java.lang.String
```

The `UOMConv` property consists of a two part construct. Each construct is enclosed within a pair of braces ("{" and "}"). Construct elements include:

- Name of the SAP Key Figure (Measure)

  The name must be enclosed in quotes.

- `TO UOM` value

For example:

The following converts `Measures.99DJXUKK99CO5EQ4V69WAGS29` to `BOX`.

```
MDSelector. UOMConv =
{"Measures.99DJXUKK99CO5EQ4V69WAGS29", BOX}
```

ANCESTOR and DESCENDANT Clauses

In Release 8.5, a new property value in the Data Source Property sheet specifies the `Sets` function values requested. The `Sets` property is used whenever the Ancestors, Ascendants, or Descendants parameter is included in the select statement.

```
(MDX_Sets_Functions={Ancestors, Ascendants, Descendants})
```

```
MDSelector.Sets.Name=MD Selector SET Property
MDSelector.Sets.Description=Specify the SET value.
MDSelector.Sets.ClassName=java.lang.String
```

The `Sets` property consists of a four part construct. Each construct is enclosed within a pair of braces ("{" and "}"). Construct elements include:

- Column position in the result set (starts from 0)
- Numeric MDX `Sets` function code value

  The MDX `Sets` function code values are maintained with the `MDX_Sets_Functions` property.
- Dimension level or member fully qualified name
- Numeric `Sets` function processing flag code value.

  The `Sets` function processing flag code values are maintained with the `MDX_Sets_Function_Flags` property.

For example:

```
MDSelector.Sets={0,3,"0APO_LOCNO.CITY”,4}
```

Note: See *Type* in Table 4 on page 70 for information on how to specify ancestors or descendants when defining a filter in SQR Developer.

Support for Key Figure Attributes – Currency, Formatted, Unit, Value Type

Note: The syntax in this section refers to *key figures* as *measures*.

In Release 8.5, a set of retrievable attributes is available for each *measure* item returned from an MDX query. The set includes:

- `FORMATTED_VALUE`—Formatted value of a cell.
- `VALUE_TYPE`—Type of value.
- `CURRENCY`—Currency key.
- `UNIT`—Unit key.
- `NAME`—Measure’s descriptive name.

Measures in an InfoObject include an entry for each of the above items. You can use these items in the body of a Select statement.

For example:

```
begin-select
```

```
 !Measures.Open orders qty
 Measures.0HPW6SU8X9M7WBGRCD1PWI471 &Master_Query_.Open_ord=number
 (11,82)

 !Measures.Open orders qty Unit
 Measures.0HPW6SU8X9M7WBGRCD1PWI471.unit &Master_QueryOpen_ord_unit=char
 (11,180)

 !Measures.Incoming orders quantity
 Measures.33L2S0QRHTBOKSWAQZAENHX2Z &Master_Query_Incoming ord =number
 (11,191)

 !Measures.Incoming orders quantity Unit
 Measures.33L2S0QRHTBOKSWAQZAENHX2Z.unit & Master_Query_Incoming
 ord_unit=char  (11,290)

 !Measures.Incoming orders quantity Name

 Measures.33L2S0QRHTBOKSWAQZAENHX2Z.name & Master_Query_Incoming
 ord_name=char  (11,331)

 From  ! query_BW401

end-select
```

Enhancements to SQR Developer

Enhancements to SQR Developer include:

- Streamlined Query Builder
- Report Layout Explorers
- Editor Unicode Support
- User Interface Enhancements
- SAP BW Enhancements

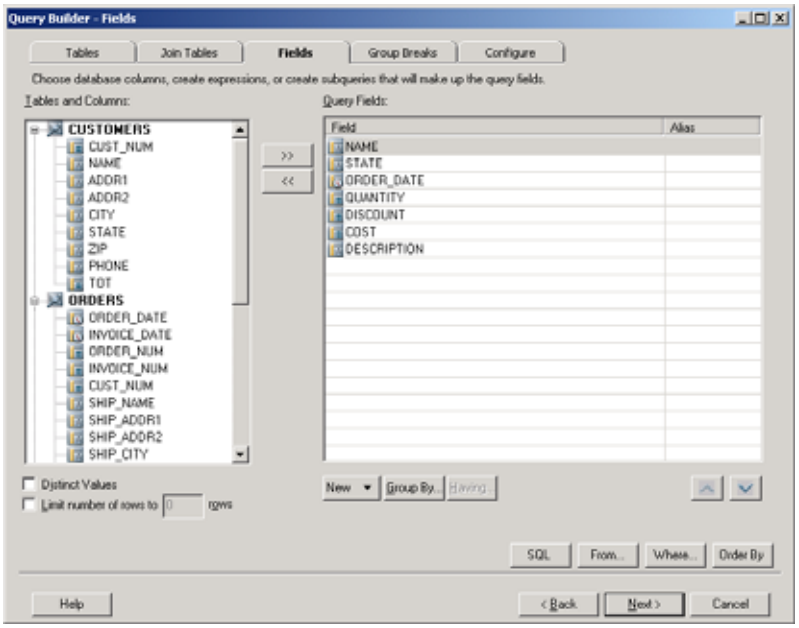
Streamlined Query Builder

Release 8.5 includes a reorganized Query Builder wizard with consistent conditions and selections regardless of the data source. In addition, the Query Builder now includes tabs to improve navigation.

➤ To move through the Query Builder, you can now:

- Click **Next** and **Back**.
- Click the tab on the top of a Query Builder page.

Figure 2 Sample Query Builder Page



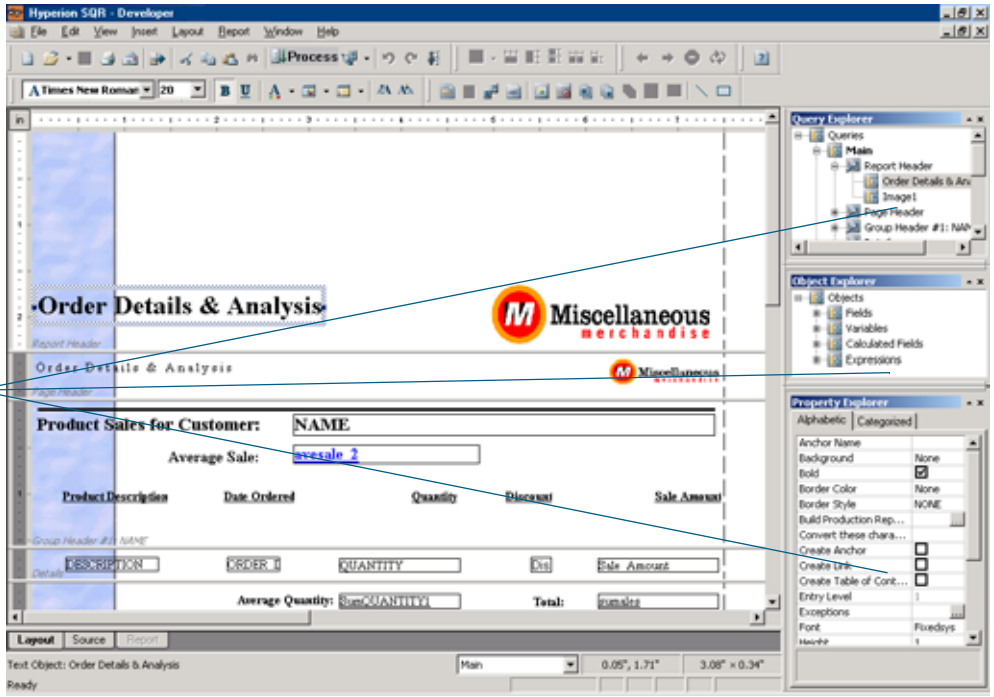
Report Layout Explorers

Users can access report control properties, queries, or objects from the menu bar, from the context menu, or from the SQR Developer Explorers. The new Explorers include:

- Query Explorer
- Object Explorer
- Property Explorer

When you initially access the SQR Developer Layout window, the Explorers appear on the right-hand side of the window. You can minimize, maximize, close, and re-open these Explorers as desired. You can also drag an Explorer to another part of the Layout window and resize the Explorer. To re-open an Explorer once it is closed, select **View** > **Explorers** and select the Explorer from the menu.

Figure 3 Sample Layout Window

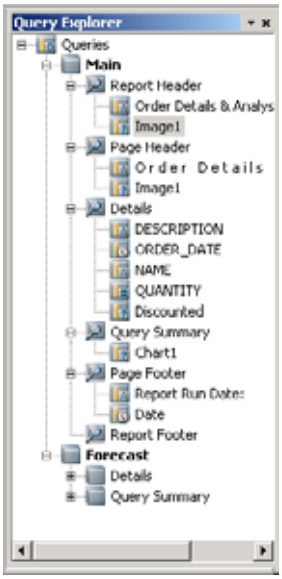


The Layout window Explorers help you format reports.

Query Explorer

The Query Explorer displays the query (or queries) used to create a report. The Query Explorer contains a “snapshot” of the report layout (it displays each Layout window section and the objects that appear in that section). When you select on object in the Query Explorer, the object is selected in the Layout window. Similarly, when you select an object in the Layout window, the object is also selected in the Query Explorer.

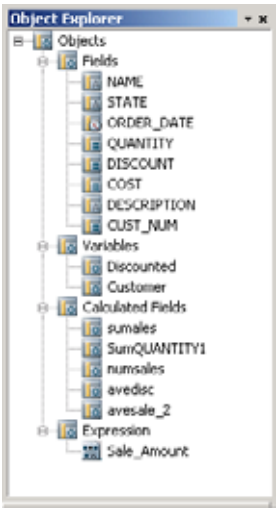
Figure 4 Sample Query Explorer



Object Explorer

The Object Explorer displays the fields, variables, calculated fields, and expressions defined in the Query Builder. As you format a report, you can drag the desired objects from the Object Explorer into the Layout window. To edit an object in the Object Explorer, right-click the object and select **Edit**.

Figure 5 Sample Object Explorer



Property Explorer

The Property Explorer displays the properties for the object selected in the report layout. You can display the properties alphabetically or by category. You can also edit the properties in the Property Explorer.

Figure 6 Sample Property Explorer

Editor Unicode Support

The Source editor tab now supports the editing of Unicode characters. This feature enables users to save SQR Developer files under either Unicode or platform encoding (Latin 1) formats.

User Interface Enhancements

Release 8.5 includes improved toolbars with new icons and expanded right-click options. Other enhancements include enabled tab order; enabled keyboard navigation; expanded alignment options; ability to minimize, maximize, and close wizards; enabled shift and crtl-click across wizards; and disabling of non-available features.

SAP BW Enhancements

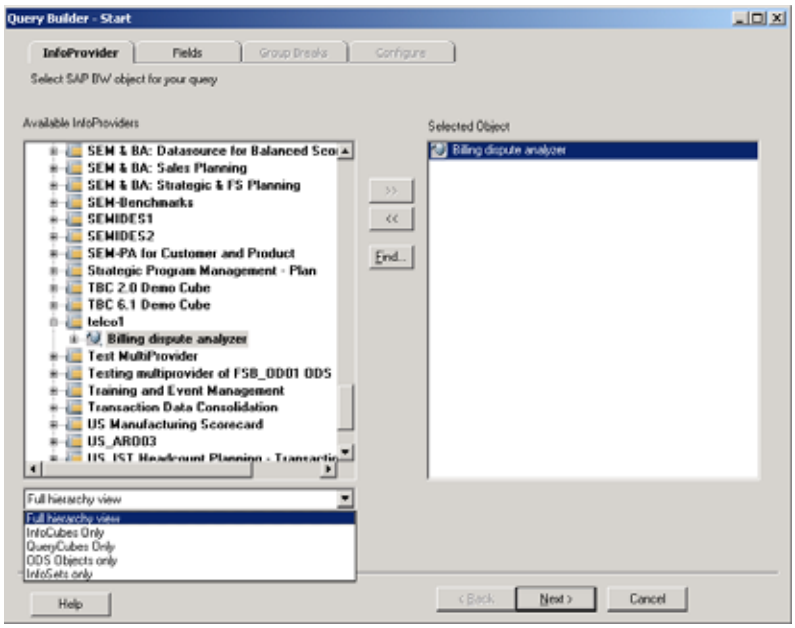
SAP BW enhancements in SQR Developer include:

- Access to ODS Objects and InfoSets (see "Selecting an SAP BW Object" on page 64)
- Support for defining SAP Variables (see "Defining SAP Variables" on page 74)
- Support for defining ancestors and descendants (see *type* in Table 4 on page 70)
- Support for defining currency and unit of measure parameters (see "Defining Currency Parameters for Key Figures" on page 69)
- Advanced filtering including the ability to filter with "and" and "or" conditions and the ability to limit the amount of metadata retrieved at one time (see "Defining Filters" on page 70)

These enhancements are included in the following sections that describe how to use the SQR Developer Query Builder to create reports using data from an SAP BW OLAP data warehouse.

Selecting an SAP BW Object

Begin building a query by selecting an SAP BW object. *You can only select one object per query.* Select an SAP BW object on the Query Builder - Start page.



➤ To select an SAP BW object:

1 **Select which SAP BW objects to display.**

To select which SAP BW objects to display, click the arrow to the right of the list box under Available InfoProviders and select an option:

- Full hierarchy view
- InfoCubes only
- QueryCubes only

- ODS Objects Only
- InfoSets Only

2 **Select the SAP BW object. To select an SAP BW object:**

- Drag the object from **Available InfoProviders** to **Selected Object**.
- Select the object and click the right arrow.

To remove an object, select the object and click the left arrow.

Tip: To display the SAP unique name for an SAP BW object, point at an object under Available InfoProviders. The SAP unique name displays as a tool tip in the metadata hierarchy.

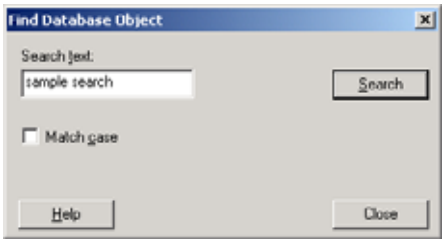
Finding an SAP BW Object

If you cannot find an SAP BW object, you can search for text in the object's name.

➤ To search for text in an SAP BW object:

1 **Click Find.**

The Find Database Object dialog box is displayed.



2 **Enter information in Search Text and click Search.**

You can search for an entire SAP BW object name, or you can search for any portion of the text in the name.

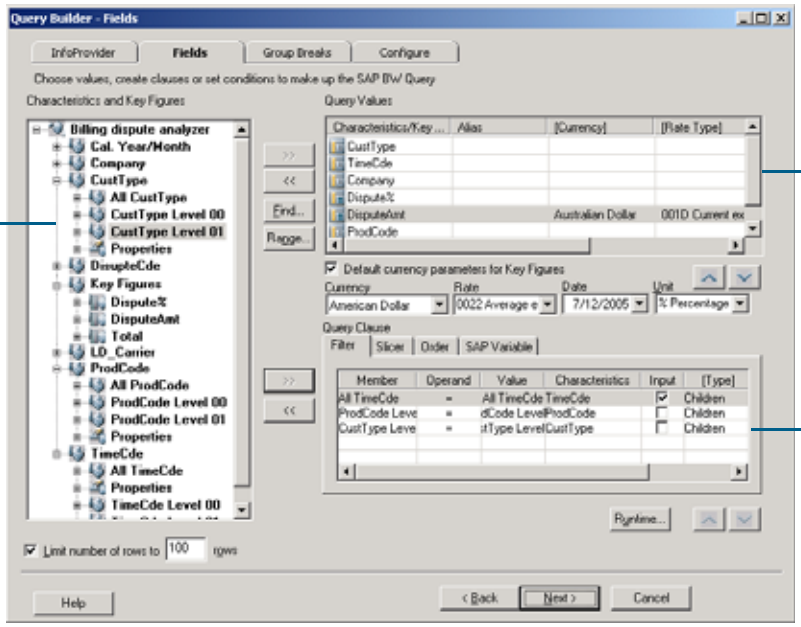
After you click **Search**, Hyperion Performance Suite highlights the first SAP BW object that contains the text.

3 **Select Match case to consider case when searching.**

Do not select this option if you want to search for all instances of the text string regardless of case. For example, *customer* as opposed to *Customer*.

Selecting Fields

After selecting an SAP BW object, the next step is to add query values. Use the Query Builder - Fields page to choose values, create clauses, and set conditions.



Browse the characteristics and key figures of the selected SAP BW object.

Select query values for the report.

Set the conditions.

The Characteristics and Key Figures pane displays the selected SAP BW object, its characteristics and key figures, and all its members. You can use this information to add query values or to set up the query clause.

➤ To add a query value:

- Drag a characteristic or key figure from **Characteristics and Key Figures** to **Query Values**.
- Select a characteristic or key figure and click the right arrow.
- Double-click a characteristic or key figure.

➤ To set up a query clause:

- Drag a member (dimension member, dimension child member, dimension level, or dimension level member) from **Characteristics and Key Figures** to **Query Clause**.
- Select a member and click the right arrow.
- Double-click a member.

Setting up a query clause allows you to add filters, slicers, and Order By clauses to the generated report. (See "Defining Filters" on page 70, "Defining Slicers" on page 72, and "Defining Order By Clauses" on page 73.)

Tip: To display the SAP unique name for an object under Characteristics and Key Figures, point at the object. The SAP unique name displays as a tool tip.

Limiting the Number of Query Rows

➤ To limit the number of query rows, select **Limit number of rows to** and enter the desired number of rows.

For example, assume your company has a large customer base. Assume further that you want a report on the first 100 customers to which your company made a sale. In this example, you would select **Limit number of rows to** and enter 100 as the number of rows. (This assumes that you sort your report by customer.)

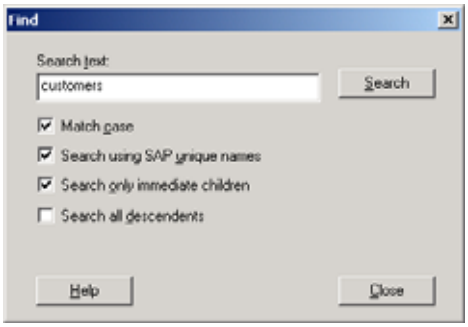
Searching for a Characteristic, Key Figure, or Member

If you cannot find a particular characteristic, key figure, or member you can perform a search.

➤ To search for a characteristic, key figure, or member:

1 **Click Find.**

The Find dialog box is displayed.



2 **Enter the text by which to search in Search Text, select the desired search options, and click Search.**

- **Match case**—Considers case when searching. Do not select this option to search for all instances of the text string regardless of case. For example, *customer* as opposed to *Customer*.
- **Search using SAP unique names**—Searches for the characteristic, key figure, or member using the SAP unique name.
- **Search only immediate children**—Searches for the immediate children of the selected characteristic, key figure, or member. Searching for immediate children can speed up the search process and improve search performance.
- **Search all descendents**—Searches for all the descendents of the selected characteristic, key figure, or member.

Restricting the Members Retrieved

When you build a query, you can restrict the members retrieved based on the required number of child nodes, level members, hierarchy levels, hierarchy members, hierarchy child members, or key figures. This provides greater flexibility and enhances performance - particularly for levels containing thousands of level members (common for InfoCubes).

➤ To restrict the members retrieved:

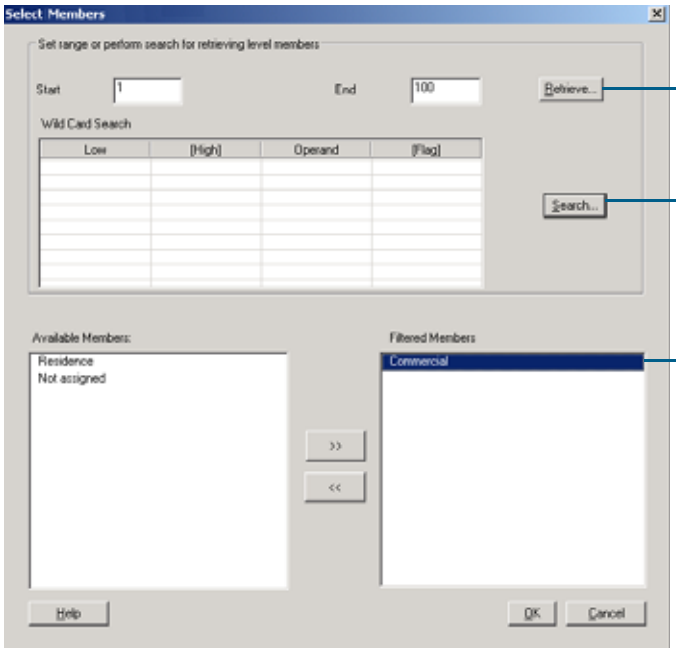
1 **On the Fields page, select a level member, hierarchy level, hierarchy member, hierarchy child member, or key figure and click Range.**

If you select a hierarchy, property, or property member, Range is disabled.

2 **In the Select Members dialog box, enter the start and end values for the range and click Retrieve.**

For example, to restrict the data to the first 100 members, you would enter 1 for the start value and 100 for the end value. If you leave the start and end values at their default value of 0, SQR Developer retrieves all of the records.

When you click **Retrieve**, the members in the defined range display under Available Members.



Click Retrieve to retrieve members according to the start and end values.

You can enter information to perform a wild card search.

You can further restrict the returned result set by moving a member to the Filtered Members pane.

3 **If desired, further restrict the members retrieved by entering information to perform a wild card search.**

- **Low**—Enter the low value for the search. For example, to restrict the search to members containing the letter "R", you would enter R*.
- **High**—Optionally, enter a high value for the search.
- **Operand**—Use the drop down list to select an operand. For example, to restrict the search to members containing the letter "R", you would select "contains" as the operand.
- **Flag**—Select either Include or Exclude.

In the example above, restricting the search to members containing the letter "R" would change the members retrieved from Commercial, Residence, and Not Assigned, to Residence.

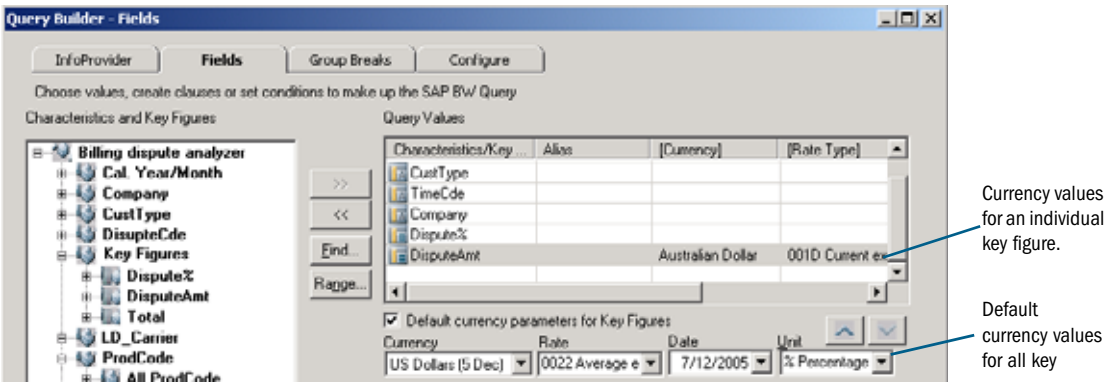
4 **If desired, further restrict the returned result set for a member under Available Members by selecting the member and clicking the right arrow to move the member to Filtered Members.**

If you move a member to Filtered Members, the filtered data appears in the Filter tab in the Query Clause section of the Fields page.

5 **Click OK to return to the Query Builder - Fields page.**

Defining Currency Parameters for Key Figures

SQR Developer allows you define currency conversion parameters for key figures in a query. Enter currency conversion information on the Query Builder - Fields page.



➤ To define the *default* currency parameters for all key figures in the query, select **Default Currency Parameters for Key Figures** and enter the information described in Table 3.

Table 3 Currency Conversion Parameters

Currency	The currency type.
Rate	The currency rate.
Date	The date on which the currency rate becomes effective.
Unit	A unit of measure based on the key figure selected. For example, if the key figure was Dispute%, the unit of measure could be *percentage*.

➤ To override the default parameters assigned to a key figure, define a currency, rate type, effective date, and unit of measure next to the key figure under Query Values.

In the example above, we defined individual currency parameters for the DisputeAmt key figure.

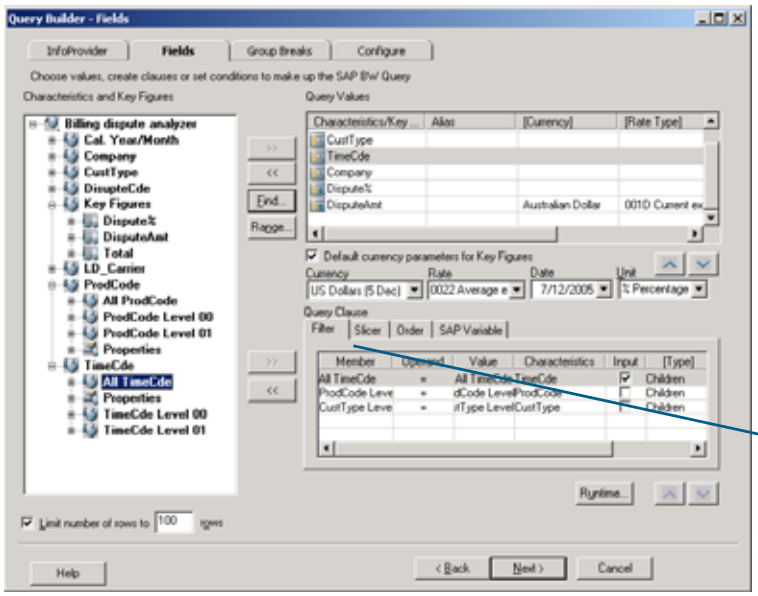
Defining Filters

Filters restrict the returned result set. Use the Filter tab in the Query Clause section of the Fields page to define a filter.

➤ To add a member to a filter:

- Drag a member from **Characteristics and Key Figures** to **Filter**.
- Select a member under **Characteristics and Key Figures** and click the right arrow.
- Double-click a member.

➤ To remove a member from a filter, click the member in the Filter tab and click the left arrow.



Use the Filter tab to define a filter.

Table 4 Filter Options

Member	The selected member.

Table 4 Filter Options *(Continued)*

Operand	The operands to use in the Filter clause. Use the following operands with key figures: = > < >= <= <> BottomCount BottomSum BottomPercent TopCount TopSum TopPercent Use the following operands with non-key figure members: = <>
Value	An integer numeric value for a key figure or the selected member of hierarchy for a non-key figure member.
Characteristics	The characteristic in the hierarchy for the selected member.
Input	Defines whether the value is an input parameter to be defined during runtime.
Type	*Optional:* The type of filter. Select a type of filter for a specific member in the output: ● Children ● Ancestor ● Descendent
Flag	*Optional:* Select a flag: ● After ● Before ● Before and After ● Self ● Self and After ● Self and Before ● Self Before After
Join	*Optional:* Defines the join type used for the filter. Select a join type: ● And ● Or The join type joins multiple filters on the same characteristic.
Runtime	Allows users to define filter values at runtime. (See "Defining Runtime Parameters" on page 74.) Note: The Runtime button is enabled only when Input is checked.
Up/Down Arrows	Not applicable for filters.

Tip: Except clauses provide additional filtering capability by limiting the set of values used for a selected characteristic. You can define an Except clause on a non-key figure member by using the < > operand in the Filter tab. This indicates that the selected characteristic should not be included in the required output.

Defining Slicers

Slicers (also called Where Clauses) specify the characteristics to use in the Query. Use the Slicer tab in the Query Clause section of the Fields page to define a slicer.

➤ To add a characteristic to a slicer:

- Drag the lowest child member for a characteristic from **Characteristics and Key Figures** to **Slicer.**
- Click the lowest child member for a characteristic under **Characteristics and Key Figures** and click the right arrow.
- Double-click the lowest child member for a characteristic.

➤ To remove a characteristic from a slicer, click the characteristic in the Slicer tab and click the left arrow.

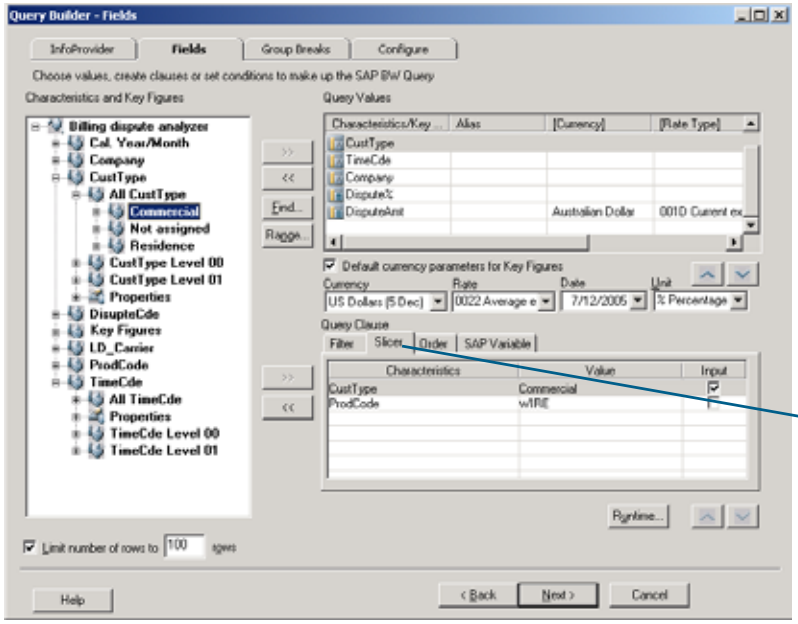


Table 5 Slicer Options

Characteristics	The selected characteristic
Value	The selected child member in the characteristic
Input	Specifies whether the value is an input parameter to be defined during runtime.

Table 5 Slicer Options *(Continued)*

Runtime	Allows users to define slicer values at runtime. (See "Defining Runtime Parameters" on page 74.) Note: Runtime is enabled only when Input is checked.
Up/Down Arrows	Not applicable for slicers.

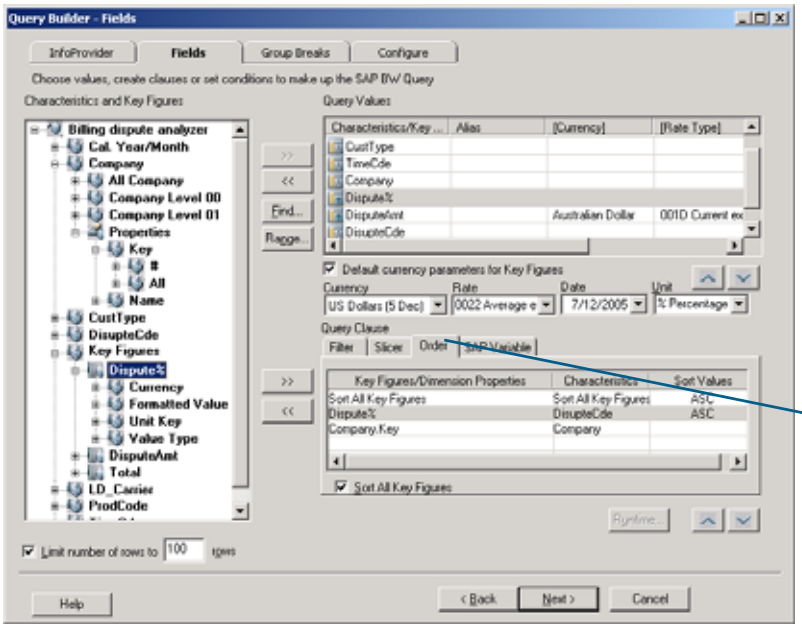
Defining Order By Clauses

Order By clauses specify the order in which the selected key figure or dimension property is returned. Use the Order tab in the Query Clause section of the Fields page to define an Order By clause.

➤ To add a key figure or dimension property to an Order By clause:

- Drag a key figure or dimension property from **Characteristics and Key Figures** to **Order**.
- Select a key figure or dimension property under **Characteristics and Key Figures** and click the right arrow.
- Double-click a key figure or dimension property.

➤ To remove a key figure or dimension property from an Order By clause, click the key figure or dimension property under Order and click the left arrow.



Use the Order tab to define an Order By clause.

Table 6 Order By Clause Options

Key Figures/ Dimension Properties	The selected key figure or dimension property.
Characteristics	The characteristic in the hierarchy for the key figure or dimension property. Use the drop down list to select a characteristic.

Table 6 Order By Clause Options *(Continued)*

Sort Values	Select an option: • ASC—Sort in ascending order and preserve the hierarchy. • BASC—Sort in ascending order and break the hierarchy. • DESC—Sort in descending order and preserve the hierarchy. • BDESC—Sort in descending order and break the hierarchy.
Sort All Key Figures	Sorts all the key figures or dimension properties for the selected SAP BW object.
Runtime	Not applicable for Order By clauses.
Up/Down Arrows	Changes the sequence of key figures or dimension properties in the Order by clause.

Defining SAP Variables

Use the SAP Variables tab in the Query Clause section of the Query Builder - Fields page to define an SAP Variable.

➤ To add a member to an SAP variable:

- Drag a member from **Characteristics and Key Figures** to **SAP Variable**.
- Select a member under **Characteristics and Key Figures** and click the right arrow.
- Double-click a member.

➤ To remove a measure member from an SAP variable, click the member under SAP Variable and click the left arrow.

Table 7 SAP Variable Options

Name	The name of the SAP variable.
Expression	The SAP variable expression.
Value	The SAP variable value.
High Value	(Optional) The high value for the SAP variable.

Defining Runtime Parameters

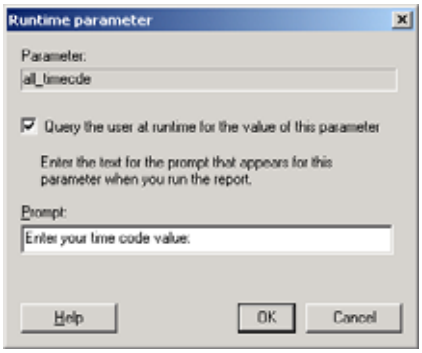
Runtime parameters allow you to enter values and generate new result sets each time you run a report. This is useful for reports with multiple parameters, such as start and stop dates, zip codes, and so on.

You can define runtime parameters for Slicers and Filters by clicking **Runtime** when you create the Slicer or Filter.

➤ To define a runtime parameter:

1 **Select a Slicer or Filter in the Query Clause section of the Fields page, check the Input box, and click Runtime.**

2 **The Runtime Parameter dialog box is displayed.**



3 **Select Query the user at runtime for the value of this parameter.**

4 **Enter a prompt to appear when the user runs the report.**

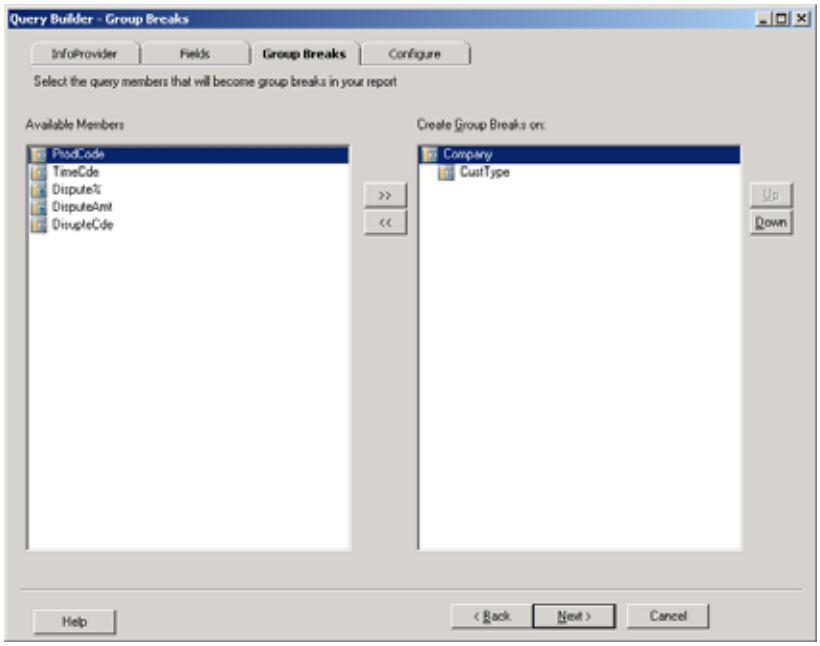
For example, a runtime parameter for *time code* could have a prompt that says: *Enter your time code value*.

5 **Click OK to return to the Fields page.**

Creating Group Breaks

Group breaks organize database information in tabular reports. Defining group breaks allows you to add white space to reports, avoid printing redundant data, perform conditional processing on variables that change, and print subtotals. When you define a group break, a column (or expression) prints only when the value of the column (or expression) changes.

Use the Query Builder - Group Breaks page to select the members that will become group breaks in the report.



➤ To select a query member to be a group break:

- Select a member and click the right arrow.
- Double-click a member.

➤ To remove a group break, click a member and click the left arrow.

Multiple group breaks must be arranged in a hierarchy. For example, if the breaks are geographical units, it is logical to arrange them according to size: first *state*, then *city*. To change group break order, select a break and click **Up** or **Down**.

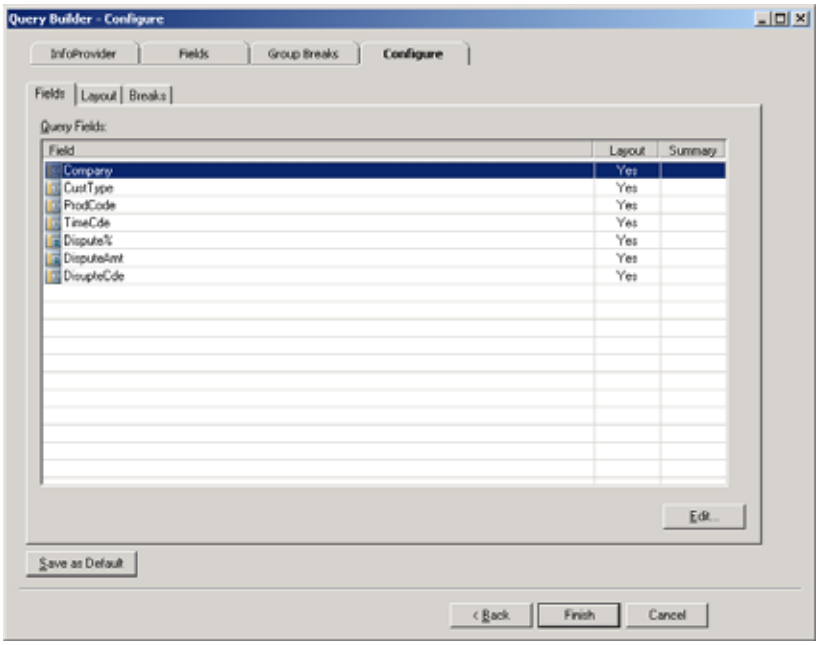
Note: SQR Developer uses the Group Breaks defined in the Query Builder to configure the default report layout. To change the group breaks while formatting a report, select **Report > Group Breaks** in the Layout window.

Configuring Layout Information

The default layout is the arrangement of data columns and headings that initially appears when you format your report layout. Use the Query Builder - Configure page to configure default layout information.

When you configure the default layout, you can:

- Configure query fields
- Configure layout information
- Configure breaks



New Excel Output Type

With SQR 8.5 you can produce a Microsoft Office Excel document as an SQR output type. The Excel output document preserves the data and formatting instructions of the original report (charts and formulas are not preserved). The integration is achieved using the Microsoft Office HTML (*.html) format.

The following printer command line flag (PRINTER:GD) has been created to generate Excel output: -PRINTER:GD -GD_DRIVER:EXCEL

For example, to create excel output from an SPF file named monthly.spf enter:

```
SQRP monthly.spf -printer:gd -gd_driver:excel
```

New Charting Functionality

Enhancements to chart creation include:

- New Attributes for the Number of Columns and Rows in a Legend
- New Attributes for Pie Chart Label Location and Pie Chart Percentage Precision
- Ability to Specify New Colors and Fonts
- Greater Control of the Name and Color for the "Other" Pie Chart Slice

New Attributes for the Number of Columns and Rows in a Legend

The `LEGEND-COLUMNS` and `LEGEND-ROWS` declaration keywords in the `ATTRIBUTES` argument for `DECLARE-CHART` and `PRINT-CHART` define the number of columns and rows to use when generating a chart legend. Table 8 describes these arguments.

Table 8 New Declaration Keywords

Declaration Keyword	Choices	Description
LEGEND-COLUMNS	Any numeric value.	Defines the number of columns to use when generating the chart legend. You can only have one LEGEND-COLUMNS value per chart. The default value is 0. (A value of 0 means the charting application will determine the proper value.) **Example**: `ATTRIBUTES=('All','LEGEND-COLUMNS',4)` Displays the chart legend with four columns.
LEGEND-ROWS	Any numeric value.	Defines the number of rows to use when generating the chart legend. You can only have one LEGEND-ROWS value per chart. The default value is 0. (A value of 0 means the charting application will determine the proper value.) **Example**: `ATTRIBUTES=('All','LEGEND-ROWS',2)` Displays the chart legend with two rows.

New Attributes for Pie Chart Label Location and Pie Chart Percentage Precision

The `LABEL-LOCATION` and `PERCENTAGE-PRECISION` declaration keywords in the `ATTRIBUTES` argument for `DECLARE-CHART` and `PRINT-CHART` define the location where labels are placed in a pie chart and the number of digits to the right of the decimal point. Table 9 describes these arguments.

Table 9 New Declaration Keywords

Declaration Keyword	Choices	Description
LABEL-LOCATION	INNER | OUTER | AUTO	Controls the location where labels are placed in a pie chart. INNER—Labels are placed within the chart area. OUTER—Labels are placed outside the chart area. AUTO—The label location is controlled by the charting application. The default value is AUTO. **Example**: `ATTRIBUTES=('All','LABEL-LOCATION’','OUTER')` Displays the pie chart labels outside of the chart area.
PERCENTAGE-PRECISION	Valid values are between 0 and 15.	Controls the number of digits to the right of the decimal point in a pie chart. The default precision is two digits to the right of the decimal point. **Example**: `ATTRIBUTES=('All','PERCENTAGE-PRECISION',5)` Displays the pie chart percentage with five digits to the right of the decimal point.

Ability to Specify New Colors and Fonts

The COLOR and FONT declaration keywords in the ATTRIBUTES argument for DECLARE-CHART and PRINT-CHART define the colors and fonts used in charts. Table 10 describes these arguments.

Table 10 New Declaration Keywords

Declaration Keyword	Choices	Description
COLOR	Use the list generated from CREATE-LIST, or enter an in-line list of values. When referencing a list, the keyword LIST: must prefix the name of the list. When using a named_color_palette, the keyword PALETTE: must prefix the name of the color palette.	Defines a single color or a group of colors for an individual chart element. COLOR values are used in presentation order and are reused once the current list is exhausted. The COLOR keyword overrides the default values established for the color property of a chart. An error occurs if the contents of the list do not match the data type expected. COLOR can reference a color palette or a single color. **Example**: `ATTRIBUTES=('CHART2.LINE','COLOR', LIST:(('red'),('yellow'),('maroon')))` Creates a group of line colors of 'red', 'yellow' and 'maroon' for chart2.
FONT	Fonts are specified in the [Fonts] section of the SQR.INI file. If this section is not defined or a font number is not found, Times New Roman is used. Each entry consists of a font number assigned to a named font. For example, 3 may represent Courier.	Defines the font for all text and/or for specific text areas within a chart image. The default font is Times New Roman. **Example**: `ATTRIBUTES=('ALL','Font',3,'Point-Size',12)` Sets the font typeface to the corresponding value from the SQR.INI file in the [Fonts] section for value 3 and the size to 12 point for all text items for all charts.

Greater Control of the Name and Color for the "Other" Pie Chart Slice

The COLOR declaration keyword in the ATTRIBUTES argument for DECLARE-CHART and PRINT-CHART defines the color for the "other" pie chart slice.

For example, the following sets the color for the "Other" pie chart slice to "red" for all pie charts.

```
ATTRIBUTES=('OTHER','COLOR',('red'))
```

UFUNC Unicode Support

Release 8.5 includes the ability to pass Unicode values with UFUNC.C

SQR provides the ability for report developers to write their own functions in C using the supplied source file UFUNC.C. You can pass any number of arguments to your function. Arguments and return values can either be numeric, single byte character strings, or UTF-8 encoded character strings. The specifics on how to specify the various argument and return types are explained in the UFUNC.C module. When using UTF-8 encoded strings, set UseUnicodeInternal=TRUE in the SQR.INI file.

Support for New Encoding

Release 8.5 supports the UTF8:BOM and UTF-8:BOM encoding as described in Table 11.

Table 11 SQR-Supported Encoding

Encoding	Character Set	Vendor / Standard Body	Acceptable SQR Names
UTF8	Unicode	Unicode	UTF8, UTF-8, UTF8:BOM, UTF-8:BOM

Ability to Call C++ Libraries from an SQR Program

In addition to using SQR's built-in functions, you can write your own functions in C using the supplied source file UFUNC.C (or EXTUFUNC.C on Windows). If the C routine accesses a C++ routine, rename the file to UFUNC.CPP. (This enables the C++ compiler to recognize that it is a C++ source file.) The supplied UFUNC.C source file is compatible with both C and C++ compilers.

You can pass any number of arguments to your function. Values can be returned by the function or passed back in variables. Arguments and return values can either be numeric, single byte character strings, or UTF-8 encoded character strings. The specifics on how to specify the various argument and return types are explained in the UFUNC.C module. When using UTF-8 encoded strings, set `UseUnicodeInternal=TRUE` in the SQR.INI file.

After editing and recompiling the UFUNC.C module, you must recreate the SQR executables. For UNIX, execute the `sqrmake` script located in the `lib` directory. For Windows, use the `sqrext.mak` make file located in the `lib` directory to recreate the DLL module.

When adding a new user-defined function to UFUNC.C, follow these rules:

- For routines that return a string value, define the routine to take the following arguments:
  - (int) Number of arguments
  - (char *) or (double *) Array of argument pointers, to either char[] or double.
  - (char *) Address of the result string. If unchanged, the function returns a NULL string.
  - (int) Maximum length of the result string, in bytes.

- For routines that return a numerical value, define the routine to take the following arguments:
  - (int) Number of arguments
  - (char *) or (double *) Array of argument pointers, to either char[] or double.
  - (double *) Address of the result value. If unchanged, the function returns a zero.

The following is an example of how to add a user-defined function to SQR so that is can be invoked using `LET`, `IF`, or `WHILE`. The example adds a new function called `rand`, which returns a random number. The function accepts a single parameter used as the seed to start a new sequence of numbers. If the seed is zero, the same sequence is used.

To add the `rand` function to the UFUNC.C module, make the following modifications:

1. Add the prototype for the function.

   ```
   LINKAGE void sqr_ufunc_rand CC_ARGS((int, double *[], double *));
   ```

2. Add the function name to the declaration table.
   - The name in the table must be in lower case; however, you can reference it in either upper case or lower case in your SQR program.
   - The name of the function called from SQR is `rand`.
   - The return type is n for numeric, the number of arguments is “1”, and the argument type is n for numeric.
   - The function name in the UFUNC.C module is `sqr_ufunc_rand`.
   - You must enter the characters “PVR” before the function name.

```
Name                     Return_type  Number of   Arg_Types        Function
                                      Arguments

{(char *)"max",          'n',         0,          (char *)"n",     PVR sqr_ufunc_max},

{(char *)"split",        'n',         0,          (char *)"C",     PVR sqr_ufunc_split},

{(char *)"printarray",   'n',         4,          (char *)"cnnc",  PVR sqr_ufunc_printarray},

{(char *)"system",       'n',         1,          (char *)"c",     PVR sqr_ufunc_sys},

{(char *)"sleep",        'n',         1,          (char *)"n",     PVR sqr_ufunc_sleep},

{(char *)"rand",         'n',         1,          (char *)"n",     PVR sqr_ufunc_rand},

{(char *)"unitest",      'u',         2,          (char *)"uU",    PVR sqr_ufunc_unitest},

/* Last entry must be NULL -- do not change */
 {(char *)"", '\0', 0, (char *)"", 0}
```

3. At the end of the UFUNC.C module add the `sqr_ufunc_rand` routine.

   ```
   /*
    * RandNumb function -- Get random number and optionally set seed
    *
    * Usage:  LET #Number = rand(#Seed)
    *
   ```

```
   */
   LINKAGE void sqr_ufunc_rand CC_ARGL((argc, argv, result))
   CC_ARG(int, argc)/* Number of actual arguments */
   CC_ARG(double *, argv[])/* Pointers to arguments */
   CC_LARG(double *, result)/* Where to store result */
      {
   #if defined(UNIX)
      if (*argv[0] > 0)/* If seed > 0 then set it */
           srand48((unsigned int)*argv[0]);
      *result = drand48();/* Get random number */
   #else
      if (*argv[0] > 0)
           srand((unsigned int)*argv[0]);
      *result = (double)rand()/(double)(RAND_MAX);
   #endif
      return;
      }
```

After you make these modifications, compile the UFUNC.C module and recreate the SQR executables.

The following is an example of a simple SQR program that uses the newly added function:

```
BEGIN-PROGRAM
     DO Get-Random-Number
     DO Process-Calculations
END-PROGRAM

BEGIN-PROCEDURE Get-Random-Number
     LET #Seed = 44
     LET #Ran = RAND(#Seed)
END-PROCEDURE

BEGIN-PROCEDURE Process-Calculations
     .
     .
     .
END-PROCEDURE
```

Ability to Add a Last Page Value to a Bar Code

In Release 8.5, you can add a last page value to a bar code as shown in the following example. When the report runs, the meta sequence `%LAST-PAGE%` will be replaced with the value of the last page of the report. This functionality is not available with bar code types 1,2,3,4,10, and 14.

```
begin-report
let $Caption = 'Page ' || Edit(#Page-Count, '88888') || ' of %LAST-PAGE%'
let $Text = Edit(#Page-Count, '88888') || ' %LAST-PAGE%'
print-bar-code (,30)
  type=5
  height=0.3
  text=$Text
  caption=$Caption
  checksum=Yes
end-report
```

New Paper Size in DECLARE-LAYOUT

Release 8.5 includes A3 as a valid paper size in the `DECLARE-LAYOUT` command. Table 12 lists the valid paper sizes.

Table 12 Valid Paper Names

Name	Width	Depth	Orientation
Letter	8.5 in	11 in	Portrait
Legal	8.5 in	14 in	Portrait
A4	8.27 in	11.69 in	Portrait
A3	11.69 in	16.54 in	Portrait
Executive	7.25 in	10.5 in	Portrait
B5	7.17 in	10.12 in	Portrait
Com-10	4.125 in	9.5 in	Landscape
Monarch	3.875 in	7.5 in	Landscape
DL	4.33 in	8.66 in	Landscape
C5	6.378 in	9.016 in	Landscape

5 Hyperion Intelligence Changes

This chapter describes the changes and enhancements made to Hyperion Intelligence Release 8.5.

In This Chapter

Exporting to Microsoft Office HTML File Formats 88
Customizing Chart Colors 102
Document Properties 104
Hyperion Intelligence Web Client Coexistence 105
Increasing the Idle Connection Time-outs for Hyperion Intelligence Web Client 107
Dashboards and Object Model Enhancements 107

Exporting to Microsoft Office HTML File Formats

You can export and deploy a Hyperion Intelligence section on an intranet or internet, and work back and forth between the HTML file and Microsoft Office 2000 Excel. This is achieved by using the Microsoft Office 2000 HTML (*.html) or Microsoft Office Web Archive (*.mhtml) file formats. These file formats preserve and recognize formatting attributes, formulas, and Visual Basic for Applications macros (VBA) created in the original section.

This feature uses Microsoft Office XML (extended markup language) tags in addition to the HTML. When the exported section is opened in Microsoft Office (specifically Excel), the formatting and formula from the original section are preserved and recognized. You can modify the file again and save it as an HTML file without the loss of any HMTL code. If you modify the file and save it in Office Excel file format, the HTML tags are not retained.

A section exported to a Microsoft Office 2000 HTML (.html) file format has the main web page saved in one folder and all the graphics and other related information (such as VBA macros) in another.

Microsoft Office 2000 Web Archive (.mhtml or .mhtml) refers to MS HTML, which is a standard for including objects in the same file as the HTML code (for example .gif or.jpeg files). Objects are encoded using the MIME HTML Internet standard. You might use this file format if you plan to e-mail the HTML as a single file.

Note: Any changes made to the section through the HTML Wizard are not recognized when it is exported to Microsoft Office 2000 HTML.

Types of HTML File Formats

Hyperion Intelligence supports three HTML file formats to which a section can be exported:

- Microsoft Office 2000 HTML (*.html)
- Microsoft Office Web Archive (*.mhtml)
- Standard HTML (*.html)

The export features are supported in the following sections:

- Tables and Results
- Chart
- Pivot
- Report

The Dashboard section cannot be exported.

➤ To export a section in Microsoft Office file format:

1. **Select File > Export > Section.**

   The Export Section dialog box is displayed.

2. **Specify the name of the section in the File name field.**

3 **Select the HTML file format for the section to be exported in the Save as type field.**

Valid HTML file formats are:

- Microsoft Office 2000 HTML (*.html)
- Microsoft Office Web Archive (*.mhtml)
- Standard HTML (*.html)

4 **Click Save.**

Comparison of HTML File Format Types

The following table shows which export attributes are supported by the HTML file format.

Table 13 Comparison of HTML File Formats

Section	Office HTML Export to Office Excel	Standard HTML Export to Excel
Results/Table	Formatting is preserved. Formulas are supported. (See Formula Mappings).	Formatting is not preserved. Formulas are not supported.
Pivot	Export preserves formatting. Cell merging is supported.	Export does preserve formatting. Export does not support formulas. Cell merging is supported.
Chart	The Chart section is written to two worksheets: the first worksheet contains the Chart data in a table, and the second worksheet contains the Chart image. Exported Chart data is aggregated in the same way as in the original Chart section. **Tip:** When a Chart section has been exported to Excel, you can use the data in the table as a data source and launch the Excel Chart Wizard to create the actual Chart. The rendered Chart looks the same as the one created in Designer.	The Chart data is written to a simple table.

Table 13 Comparison of HTML File Formats *(Continued)*

Section	Office HTML Export to Office Excel	Standard HTML Export to Excel
Report	The Report data is written to a simple table. If a report has only one table or one pivot (no matter how many pictures and or text labels are in the report), column widths are preserved. Otherwise column widths are not maintained. Column width, in this case, is resized to fit data from any table. Formatting is preserved. Report elements including: text, labels, images, tables, pivots, and charts are placed one after another vertically. A vertical gap of a certain fixed height exists between two elements. If elements overlap vertically by a small value, these elements are shown with no gap between them. This is to accommodate any design that includes stacked elements. The gap is one row high. Report headers/footers are output. Graphical elements such as lines, circles are not supported. Overlapping tables are not supported and are computed consecutively. Page headers/footers are not supported.	The Report data is written to a simple table.
Dashboard	Not supported except as an image in the export feature.	Not supported

Exporting a Section to Microsoft Office Excel Worksheets

Each exported section to Microsoft Office HTML creates one worksheet in the Excel workbook even when there are multiple sections. Exported sections of the same BQY do not reference each other in the Excel worksheet. For example if a Table section is created from a Results section, both exported sections are displayed as independent worksheets in Excel.

An Internet Explorer browser can show more than 65, 536 rows, and an Excel spreadsheet has 256 columns and a maximum of 65,536 rows. If an exported section has more rows than the Excel row maximum, all rows are exported to the source rows available, but only 65, 536 rows are processed for Excel to process any formulas. In this case, Excel truncates the exported data in the section.

Number and Date Formats

Number and date formats in Hyperion Intelligence sections are exported based on the following predicates:

- Any text within BI format strings is enclosed double-quotes (“”).
- Hyperion Intelligence fills the number formats for positive, negative, and zero numbers to achieve compatibility with Excel’s construction. Excel number format definition consists of the above number formats, delimited with a semi-colon.
- Date/time formats are exported as is except for the am/pm format, which receives a special format.
- Thousands and decimal separators are converted to Excel special symbols based on the default BI locale format. When the Hyperion Intelligence number format is evaluated, the format is converted so that the decimal separator becomes “.” (a period) and the thousand separator becomes “,” (a comma) in the target Excel format. Excel treats these special symbols as locale independent separators and replaces them with actual separators from the user’s locale at run-time.
- A value in an Excel cell has two aspects: the value formatted for displaying and the original value. Similarly when a value is placed in a HTML table, a special attribute x:num=”<actual value>” is used to preserve the original value. The pre-formatted value is placed in a cell of the HTML table, and it can be shown in the web browser. Formulas operate with the original values. The x:str attribute is used as a global designator in an html <table> in order to tell Excel that all the values which do not have x:num attribute specified should be treated as strings. This is useful when there is mixed string and numeric data in a string type column. The third component associated with an HTML table cell is a style class. It is used to specify the custom numeric format for a cell value. Date and time values are represented as numbers. These values are saved in the x:num attribute, and the style class definition specifies the format for date.

Styles

Exported sections support styles (CSS) that have been applied to the section including: font name and size, and bold/italic properties.

The Hyperion Intelligence overline and double overline properties are not supported in Excel, because Excel has no equivalent styles.

The simple overline style is supported in a browser; the double overline is not. In addition, the overline applies only to the text, not the whole cell as in Designer.

The Spotlighter feature is not supported in Excel. The appearance of a cell to which the spotlighter has been applied, retains the same look as in the exported spreadsheet, but without the real auto-formatting. Excel has an equivalent feature called Conditional Formatting. This feature can use a formula returning a Boolean result as a condition.

Colors

Excel supports a fifty-six color palette of which sixteen are used exclusively for Charts, and forty for cells in a spreadsheet. Colors can be defined by using RGB notation, and are shared among all worksheets in a workbook.

All colors displayed in the sections being exported are saved as a custom palette and saved in the HTML file (in a hidden xml section). By default Excel uses the color palette as a baseline for creating a custom palette. The colors that match Excel's colors retain their positions. Unused positions are filled by default Excel colors.

Excel automatically determines which index in the color palette to use for a particular color occurring in CSS.

If the total number of colors is more than forty, Excel determines the color placement.

Text Wrapping

If text wrapping is disabled in Designer and the section is exported, Excel sizes the column to display all values within a row. Words that do not fit into the cell are wrapped to the next line and are not visible. In the browser, the text is truncated.

Chart as Image

A worksheet containing a chart image shows no cell grid.

Charts larger than a single image (chart scrolling is enabled) are placed on a single worksheet, and page breaks are provided. Each image is printed on its own page.

Headers and Footers

Headers and footers are converted into appropriate Excel equivalents, and are displayed in a printed document. In addition, the format template is converted to the Excel format. Multiple headers and footers are supported, but they are converted into a single multi-line header or footer.

Headers and footers are not displayed when the exported section is opened in a browser.

Printing Improvements

Results, Table, and Pivot sections are exported so that the table headings and pivot top and side labels are marked. These marks allow headings and labels to be duplicated on each page when they are printed from Excel.

Suppress Duplicates for Results or Table

The Suppress Duplicate feature hides duplicate values in a selected column but includes them in calculations. Hidden values can be seen and changed only by clicking the cell in Excel. Typing any new value does not change the hidden status of the cell unless the user changes the cell format.

Formula Generation

An exported BQY table section can have the following kinds of JavaScript expressions:

- Grand/Break Total calculation (for example, Sum(Units, Breaks) + 10)
- Grand/Break Total labels (for example, "Total "+ ToChar(Store))
- Computed Item expressions (for example, Units*2 + 10)

When exporting a table with expressions, the following steps are performed:

- Values that are calculated using formulas are exported to HTML, and available to be shown in a web browser.
- Excel formulas are generated for each cell in the table and written to HTML. This file can be opened in Excel and all values are re-calculated. An end-user can change values in the cells and results are re-calculated again.

JavaScript Expressions

JavaScript expressions used with the Export to HTML feature can refer to columns in the table and invoke computed items. They can also refer to any JavaScript class like Math and String. Those computed items which do not have direct equivalent among the built-in Excel functions are simulated by custom VB functions. For information about how Hyperion Intelligence computed items map to Excel's built-in function, see "Formula Mappings" on page 94.

If the expression refers to a JavaScript class not supported by the Export feature, #NAME? is displayed in the Excel cell (and as a valid value in the web browser) and the Error Office html tag attribute is used. In addition, a comment is added to the cell explaining the error. In Excel, this comment is displayed near the cell where the error occurred. In the browser, the comment is displayed when you hover the mouse pointer over a cell with a small red-colored corner.

Mapping JavaScript to Excel

Hyperion Intelligence uses a translator to distinguish between the range types used to substitute a column reference in a computed item expression. There are three types of ranges:

- Continuous range corresponding to the whole column (for example, Units)
- Aggregate range – this is a continuous range that is broken by Break Totals (for example Units_Agg). The only purpose of this range is as a data source for aggregate functions. The translator uses Units in all kind of expressions that refer to the column, but it uses Units_Agg range for aggregate functions, for example, SUM(Units_Agg).
- Continuous sub-range used in break total computation.
- Individual cell references (for example, A1, B5)

The table below outlines how the JavaScript expression operations are mapped to Excel.

JavaScript	MS Excel
*, /, <=, =>, -, + (arithmetical plus operator)	Uses the same symbols.
%	MOD() built-in function
!=	<>
==	=
+ (concatenation operator)	&
Exp1? Exp2: Exp3 (ternary operator)	IF(Exp1, Exp2, Exp3)
Comma operator (e.g. exp1, exp2, ..., expN)	Replaced with the last expression. For example, (exp1, exp2, ..., expN) -> is replaced with expN in resulting Excel formulas. Expressions should be of the same type (expression or string)
!	NOT() built-in function
&&	AND() built-in function
||	OR() built-in function
Character literal (e.g. 'a')	Converts to string literal (for example, "a")
Hexadecimal and octal number (e.g. 0x10, 010)	Converts to a decimal number (for example, 16, 8)
Logical constants (true and false)	Converts to Excel's TRUE and FALSE
Expressions with 'null'	Expressions with 'null' have limited support. Column values can be compared with null. Computed Items should not produce null values.
\n\t\r\a\b and \xHH in string literals. Example: "a\na"	Replaced with CHAR(x) in the output string.For example: "a" & CHAR(10) & "a"

The JavaScript' '+' operator can be applied to strings and denotes a string concatenation. Excel's string concatenation operator is '&'. Hyperion Intelligence distinguishes between arithmetic '+' and string '+' by tracking the type of arguments in expressions. If one of the arguments is a string literal, a concatenation operator is used. The translator tracks the return type of Object Model functions and columns, so it can infer a final expression type. There are three supported types: numeric, string, and date.

Formula Mappings

Most of the computed items that comprise computed item expressions can be mapped directly to Excel functions. Certain functions have different numbers of arguments or argument order and require additional processing.

The following table shows which Hyperion Intelligence computed items map to Excel formulas.

Table 14 Computed Items to Excel Function Comparison of HTML File Formats

Hyperion Intelligence Computed Item	Microsoft Excel Function	Implementation	Aggregation
Decode	H_Decode	VBA	
Nvl	Limited support: Nvl(column, expr).	Excel	
AddMonth	H_AddMonth	VBA	
DayOfMonth	DAY	Excel	
LastDay	H_LastDay	VBA	
MonthsBetween	H_MonthsBetween	VBA	
NextDay	H_NextDay	VBA	
Sysdate	NOW	Excel	
ToChar(value, format)	Limited Support: Excel's built-in TEXT(val, format) function is used and the format argument is translated from the BI format into Excel's on the formula generation time. Consequently, the format should always be a constant string literal.	Excel	
ToChar(column)	When the argument for ToChar is a column reference, it is converted to TEXT(column, format), where format is one of real, int, or date_format depending on the column type.		
ToDate(x)	(DATEVALUE(x) + TIMEVALUE(x))	Excel	
ToMonth	H_ToMonth	VBA	
ToQtr	H_ToQtr	VBA	
ToYear	YEAR	Excel	
Abs	ABS	Excel	
Atn	ATAN	Excel	
Ceil	H_Ceil	VBA	
Cos	COS	Excel	

Table 14 Computed Items to Excel Function Comparison of HTML File Formats *(Continued)*

Hyperion Intelligence Computed Item	Microsoft Excel Function	Implementation	Aggregation
Cosh	COSH	Excel	
Exp	EXP	Excel	
Floor	H_Floor	Excel	
Ln	LN	Excel	
Log	LOG	Excel	
Log10	LOG10	Excel	
Max	MAX	Excel	
Min	MIN	Excel	
Mod	H_MOD	VBA	
Power	POWER	Excel	
Round	ROUND	Excel	
Sign	SIGN	Excel	
Sin	SIN	Excel	
Sinh	SINH	Excel	
Tan	TAN	Excel	
Tanh	TANH	Excel	
Trunc	TRUNC	Excel	
Avg	H_Avg	VBA	✔
AvgNonNull	AVERAGE	Excel	✔
Chr	H_Chr	VBA	
ColMax	H_ColMax	VBA	✔
ColMin	H_ColMin	VBA	✔
Count	H_Count	VBA	✔
CountDistinct	H_CountDistinct Limited Support: Can be used in a break or grand totals only. There may be performance considerations associated with this computed item because it uses heavy computations.	VBA/Excel	✔
CountNonNull	COUNTA	Excel	✔

Table 14 Computed Items to Excel Function Comparison of HTML File Formats *(Continued)*

Hyperion Intelligence Computed Item	Microsoft Excel Function	Implementation	Aggregation
CountNull	H_CountNull	VBA	✔
Cume	Computed Items with expressions that have exactly one call to Cume() and nothing else. Example: "Cume(Units)"	Excel/VBA	Aggregation and cross references in rows
Next	Function call is converted to appropriate cell reference. Limited support: Cannot be used in a Break or Grand Total expression.	Excel/VBA	Cross references in rows
Prior	Function call is converted to appropriate cell reference. Limited support: Cannot be used in a break or grand total expression.	Excel	Cross references in rows
Sum	Use Excel's SUM() for non-string columns. String columns exception: H_SumStr() is used. Can be used in break and grand totals only. There may be performance considerations associated with this computed item because it uses heavy computations.	Excel/VBA	✔
Median	MEDIAN	Excel	✔
Mode	H_Mode Mode maps to Excel's H_Mode because Excel's MODE does not support complex ranges. Mode can only be used in break or grand totals. There may be performance considerations associated with this computed item because it uses heavy computations.	VBA	✔
Percentile	H_Percentile	VBA	✔

Table 14 Computed Items to Excel Function Comparison of HTML File Formats *(Continued)*

Hyperion Intelligence Computed Item	Microsoft Excel Function	Implementation	Aggregation
Rank(column)	H-Rank Rank maps to Excel's H_Rank function. Rank can only be used in break or grand total expressions. There may be performance considerations associated with this computed item because it uses heavy computations.	VBA	✔
RankAsc(column)	H-RankAsc This function can only be used in break or grand totals expressions. There may be performance considerations associated with this computed item because it uses heavy computations.	VBA	✔
StdDev	H-StdDev	VBAI	✔
StdDevp	H_StdDevp	VBA	✔
Var	Var	VBA	✔
Varp	Varp	VBA	✔
Ascii	CODE	Excel	
Concat	CONCATENATE	Excel	
Initcap	H_Initcap	VBA	
Instr	H_Instr	VBA	
Length	LEN	Excel	
Lower	LOWER	Excel	
Ltrim	H_Ltrim	VBA	
Replace	H_Replace	VBA	
Rtrim	H_Rtrim	VBA	
Substr	MID	Excel	
Translate	H_Translate	VBA	
Upper	UPPER	Excel	
MovingAvg	H_MovingAvg	VBA	Cross references in rows
MovingDiff	H_MovingDiff	VBA	Cross references in rows

Table 14 Computed Items to Excel Function Comparison of HTML File Formats *(Continued)*

Hyperion Intelligence Computed Item	Microsoft Excel Function	Implementation	Aggregation
MovingMax	H_MovingMax	VBA	Cross references in rows
MovingMed	H_MovingMed	VBA	Cross references in rows
MovingMin	H_MovingMin	VBA	Cross references in rows
MovingSum	H_MovingSum	VBA	Cross references in rows
All aggregate functions with two and three arguments: Ex. Sum(col, break_col, break_value)	When two argument functions are used in break or grand total expressions and the second argument is 'Breaks', most of the functions are converted to their built-in Excel equivalents. Otherwise, the function is substituted with the appropriate H_xxx() VBA function.	Excel/VBA	

Custom Formulas

Hyperion Intelligence has a number of computed items that do not map directly to an equivalent Excel function. This section describes the methods the Export feature uses to construct a mapping from the Hyperion Intelligence computed item to an Excel function.

Adding VBA functions to the Exported Excel File

Hyperion Intelligence helper functions implemented in Visual Basic are saved to the exported file. In this instance, the Excel file becomes self-contained. It can be passed to any user as is and does not require installing additional components such as Excel add-ins. All the functions are implemented within this file.

Since VBA functions are treated in Excel as macros, Hyperion Intelligence signs them digitally in order to prevent a security warning dialog box in Excel if the macro security level is medium or high. When the macros are signed, you can observe certification information and choose to trust the sources. Once the certificate is marked as trusted, all subsequent openings of exported Excel files proceed silently.

Unsupported Formulas

The Excel formula content limit length is 1,024 characters.

Null Values

In Hyperion Intelligence, a null value is a cell absent of data, but which does not equal zero. In Hyperion Intelligence cells that are absent of data are represented by blank cells. Excel does not recognize null values, but has a similar concept. There are also built-in aggregation functions analogs which understand this concept.

Custom Cell Ranges

Named cell ranges are generated to construct Excel formulas for computed items in the spreadsheet. For example, if there is a Table section with Break Totals, then the following types of named ranges are generated:

- A continuous cell range that includes the whole column

  For example: Units = Results!$C$2:$C$998. This range can be used in all kinds of formula expressions, but cannot be passed to aggregate functions like SUM.

- A compound or broken cell range that includes all the cells with actual data excluding those occupied by Total information

  For example: Units_Agg = Results!$C$2:$C$4,Results!$C$6:$C$34,Results!$C$36:$C$164. This range can be passed to aggregate functions, which is reflected in its the name.

- A cell range used in break totals calculations

  For example: Sr_32_2=Results!$C$2:$C$4

Below is a sample spreadsheet generated from the Results section. The column Computed of the section contained Units*2 expression, the column Computed2 contained SUM(Units) expression. The generated Excel formulas are shown in the cells.

	A	B	C	D	E
1	Store	Item Name	Units	Computed	Computed2
2	Bayshore Electronics	1024 MB Drive	45	=Units*2	=SUM(Units_Agg)
3	Bayshore Electronics	EZ Fax Modem-I	30	60	21265
4	Bayshore Electronics	EZ Fax Modem-x	45	90	21265
5	Bayshore Electronics	One Button Mouse-	30	60	21265
6	Bayshore Electronics	The Quad	35	70	21265
7	Bayshore Electronics	One Button	30	60	21265
8	Bayshore Electronics	One Button Mouse-	40	80	21265
9	Bayshore Electronics	One Button Mouse-	35	70	21265
10	**Total Bayshore Electronics**		**475**	=SUM(Sr_13_2)	

For the Excel user, the Auto Outlining feature can be used to create row groups from the Break Totals that Hyperion Intelligence creates. Excel analyzes the formulas and creates the row groups and outline automatically. The user can then expand or collapse individual groups of rows.

Named cell ranges created in Hyperion Intelligence can be used for in-sheet formulas and in an external Excel file (worksheet linking). For example, an end user can create a new Excel file and enter a cell formula that refers to a data column of the workbook file exported from Hyperion Intelligence. For example, an end-user might enter: =SUM(BIExport.xls!Results_Units_Agg), where Results_Units_Agg is an auto-generated range that denotes the Units column of the Results table in the BIExport.xls file.

Worksheet Names and Range Names Generation

Worksheet names are based on the BQY section names. Before exporting a section, consider the following limitations that apply to worksheet names:

- The name of section to be exported should not be longer than thirty-one characters.
- The symbols / \ ? * are not allowed and are removed.
- Duplicate worksheet names are appended with a number to ensure the uniqueness of each worksheet.
- A worksheet name can include space symbols. When such a worksheet name is used in formulas, it should be surrounded with single quotation marks.

Generated range names that correspond to BQY table columns are based on a respective column name. The following restrictions apply to generated range names:

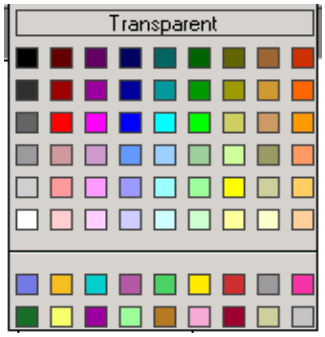
- The worksheet name is added to the beginning of the range name to ensure its uniqueness within the whole workbook.
- Spaces within the name are replaced by underscores.
- The symbols ~!@#$%^&*()_+|{}:”<>?`-=\[];’,./ are not allowed and are removed.
- Duplicate range names are appended with a number to ensure the uniqueness of each range name.
- Columns must have appropriate identifiers in JavaScript, for instance, an Order Date column can be referred to as Order_Date in a computed item expression. This identifier is used as the main part of a generated range name.
- Excel has a limitation on the length of the expression used as a named range. The total length of the references (not including worksheet names) cannot be more than 255 characters. When a named range is generated, its content is broken into smaller chunks, each comprising a smaller named range. The final range is a union of these smaller ranges.

Customizing Chart Colors

This section explains the color options available in the Chart section.

Default Color Palette and Legacy Color Palette

By default, Hyperion Intelligence bases the color scheme used in a chart on the default colors in the color palette. This palette consists of a legacy color palette (pre Release 8.4 color palette) and a default color palette as shown below:



The legacy color palette consists of the first six rows in the color palette. The default color palette consists of the last two rows. There are seventeen chart default colors and the chart background color is the last color on the bottom row.

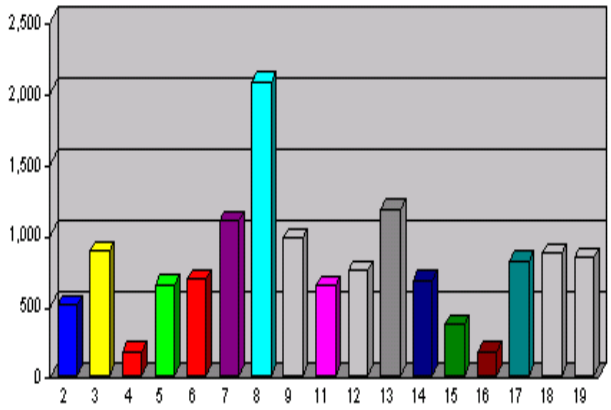
Using Legacy Chart Colors

When you open a chart created in Release 8.3 and earlier, the chart uses the default colors available in the legacy color palette. This feature is enabled automatically for all Release 8.3 and earlier charts in the "Use legacy colors for data points" check box on the Default Fonts and Styles dialog box. In addition, when this option is checked, all newly added charts use the default colors in the legacy color palette that was used in Release 8.3 and earlier. When this is unchecked (the default value), all newly added charts use the default colors in the default color palette. This value is saved with other application level parameters.

For an individual chart, you can use the default colors available in the legacy color palette by enabling the "Use legacy colors for data points" check box on the General tab of the Chart Properties dialog box.

If you modify the chart (for example, you perform a drill down, or change the chart type), the colors of the chart use the default color sets depending on the "Use legacy colors for data points" option. For example, if the check box is not enabled, the new default colors are used. Otherwise, the old default colors (pre Release 8.4) are used.

In the example below, the bar chart uses the default colors from the legacy color palette:



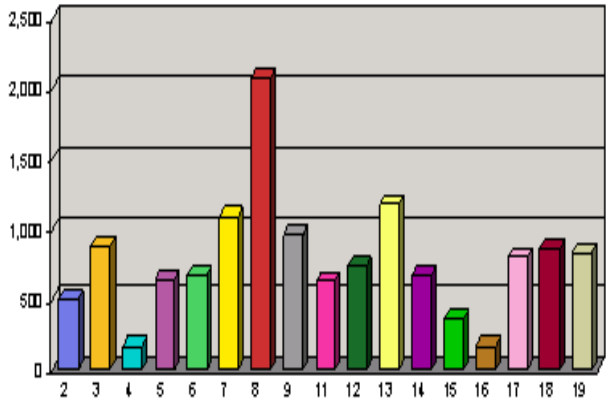
Using Default Colors

By default, charts created globally in Release 8.4 and later use the default colors available in the Default Color Palette. In this case, the "Use legacy colors for data points" check box is disabled on the Default Fonts and Styles dialog box.

For an individual chart, you can use the default colors available in the default color palette by disabling the "Use legacy colors for data points" check box on the General tab of the Chart Properties dialog box.

If you want to use the default colors from the legacy color palette, enable the "Use legacy colors for data points" check box on the General tab of the Chart Properties dialog box.

In the next example, the bar chart uses the colors from the default color palette:



General Chart Properties

The General tab of Chart Properties has been enhanced and includes the following options:

- **Do not remove text** –Toggles the Do Not Remove Text feature. When this feature is enabled, Smart Scaling does not remove any text (it does not clip or overlap text if an object is resized or repositioned to a minimum font size.). This feature is checked for new Chart sections, and unchecked for Release 8.3 and earlier Chart sections that have Smart Scaling enabled. Inserted text and legends are not affected by your selection in this field. Smart Scaling may shrink these objects, but they are never scaled to the point where they are eliminated entirely.
- **Use legacy colors for data points** – Enables an individual chart to use legacy colors from the Release 8.3 and earlier Color Palette. When this option is unchecked, the newly added chart uses the default colors from the 8.4 and later color palette. To enable all charts to use the legacy colors from the Release 8.3 and earlier color palette, check the "Use legacy colors for data points" option on the Default Fonts and Styles box. This option is saved with other options at the application level.

Computed Items

Conditional Scalar Function

The conditional scalar function Nvl assumes three values: c,exp1,exp2 (returns exp2 if null, and exp1 if not). However, if the function is included in a computed item expression, only two parameters are allowed: exp1 and exp2.

Document Properties

The Document Properties tabs are used to describe information about a BQY, such as in which application the document was created and also the server address associated with a document. Document properties consist of the General tab and the Server tab.

General Document Properties

Use the General Document Properties dialog box to display the creation and modification attributes of a document.

The following Created display fields for the document are shown on the General Document Properties dialog box:

- **Document Path** – Displays the path name to which the document has been saved.
- **Application Name** – Displays the application name in which the document was created. Application names include:
  - Hyperion Intelligence Client

- Hyperion Intelligence Web Client
- Intelligence iServer
- Hyperion Intelligence Job
- Hyperion SmartView for Office™

- **Application Version** – Displays the release in which the document was created.
- **Date** – Displays the date on which the document was created.

The Modified display fields are:

- **Document Path** – Displays the path name to which the modified document was saved.
- **Application Name** – Displays the application name in which the document was last modified. Application names include:
  - Hyperion Intelligence Client
  - Hyperion Intelligence Web Client
  - Intelligence iServer
  - Hyperion Intelligence Job
  - Hyperion SmartView for Office
- **Application Version** – Displays the release in which the document was last modified.
- **Date** – Displays the date on which the document was last modified.

Server Document Properties

Use the Server Document Properties dialog box to specify the BQY server address.

➤ To specify a server address for a document, enter the URL address in the Server Address field and click **OK**.

The URL address must be in the following format: http://<server>/workspace/dataaccess.

Hyperion Intelligence Web Client Coexistence

Hyperion Intelligence Web Client users can now work with different versions of the Hyperion Intelligence Web Client on the same client machine. To enable this feature, new Hyperion Intelligence Web Client installations include multiple versions of the application. Users can work with all servers older than Release 8.3 through Release 8.5. The feature was added because the Hyperion Intelligence Web Client installation is not backwards compatible, and only supports the Hyperion Intelligence Web Client Release 8.5. The correct version that gets used is determined at run-time when the BQY is opened. For example, if a user opens a BQY in a Release 8.3 server, the 8.3 version of the Hyperion Intelligence Web Client is launched to serve the request.

Two Hyperion Intelligence Web Client versions are installed in this release: an 8.3 version and the current 8.5 release. Each Hyperion Intelligence Web Client plug-in entity is separated on the disk and in the registry. Each version is installed in a separate folder under <IE Install Directory>\PLUGINS\Hyperion\<Insight Series Version> or under <Firefox Install Directory>\Plugins\Hyperion\<Insight Series Version>.

A mechanism has been implemented to find and start a particular Hyperion Intelligence Web Client executable based on its version. The Hyperion Intelligence Web Client Launcher that provides Release 6.x and 8.0 coexistence has been enhanced for Releases 8.3 through 8.5 and future coexistence. There is a single launcher in the system associated with the BQY mime-type and file extension.

Locally Saved BQYs

If a user is working with a locally saved BQY and saves it to disk from the Hyperion Intelligence Web Client, the file is appended with version information. If the user opens the file again, the appropriate version of the Hyperion Intelligence Web Client is launched. After this, Hyperion Intelligence Web Client checks the server version to connect to the server. If the server is newer than the current Hyperion Intelligence Web Client, the user is prompted to upgrade to the current version. In this case, the older version accesses the URL of the server, launches the zero administration page, and performs the upgrade. The Hyperion Intelligence Web Client is opened from the disk using file name in the upgraded version.

If a BQY was created in previous versions of the Hyperion Intelligence Web Client or in the Designer, the user is prompted for the version to use. The user can select from a list of versions of the Hyperion Intelligence Web Client, including the 6.x version.

If the user selects a 6.x version, no version record is attached to the document because there is no command on the Tools menu to open the document in another version of the Hyperion Intelligence Web Client.

After the Hyperion Intelligence Web Client is launched, the user can change the Hyperion Intelligence Web Client designated to open the document at any time by selecting the following options from the Tools menu:

- Change Web Client version to open this file
- Make this Web Client default for locally saved BQY files.

Hyperion Intelligence Web Client Mismatches

If there is a mismatch between the Hyperion Intelligence Web Client series required by the server and the user selects a compatible version, a file version record is saved and the document is reloaded.

Required Hyperion Intelligence Web Client series information is only provided by the Release 8.5 servers. If this information is unavailable, it is assumed that the user is connected to either an Release 8.3 or Release 6.x server. If the user opens a document in the Hyperion Intelligence Web Client from a release prior to the server release (for example, a document from a Release 6.x server is opened in a Release 8.3 server), the Save on Exit prompt is suppressed when the document is reloaded to prevent the user from upgrading accidentally to the newer series.

Increasing the Idle Connection Time-outs for Hyperion Intelligence Web Client

Using a configurable polling time, administrators can now control the frequency of messages that keep a Hyperion Intelligence Web Client connection active. This enhancement allows queries of varied lengths to process from Hyperion Intelligence Web Client when running behind a proxy or firewall with finite idle-connection time-outs. Previously, time-outs could not be increased on a proxy server and firewalls in many deployments. By default there is no polling and this feature needs to be explicitly enabled.

➤ To enable this feature follow the procedure below:

1. **Open the ws.conf file from the <SERVLETS DEPLOYMENT>/config folder in an editor such as Notepad or Wordpad.**
2. **In the ws.conf file, type: WebClient.Applications.DAServlet.PollingTimeSec=0.**
3. **Change the PollingTimeSec parameter value from 0 to the desired value.**
4. **Save the entry is in the ws.conf file.**

   Use the Servlet Configurator (<INSTALL_HOME>/servlets/bin, servletconfig.bat for Windows and servletconfig.sh for UNIX to manipulate the value.

   Note the value of the PollingTimeSec parameter value setting:

   - The Default value is 0 indicating that there is no polling mechanism in place to keep the connection active.
   - The PollingTimeSec value is in seconds.
   - The PollingTimeSec value must be less than proxy or firewall time-out value (otherwise this configuration change has no effect).
   - Setting the PollingTimeSec value too low increases network traffic. For example, if a firewall is set for a 15 min idle timeout, set the polling time to be 720 sec (== 12 min). There is no need to set the value for example to 10 seconds because an unnecessary network overhead results.

Dashboards and Object Model Enhancements

Object Model enhancements have been added in Release 8.5 which address the Unicode, Embedded Pivot Section Interactivity, and PrintOut features.

The Unicode Object Model enhancements include:

- "UnicodeEnabled (Property)" on page 108
- "SortOrderLang (Property)" on page 109
- "DefaultSortOrderLang (Property)" on page 111
- "UILanguage (Property)" on page 117
- "ResetDefaultSortOrderLang (Method)" on page 128

Object Model methods that have been modified as a result of the Unicode feature include:

- "Export (Method)" on page 129
- "ExportToStream (Method)" on page 132
- "ImportDataFile (Method)" on page 134
- "ImportSQLFile (Method)" on page 136

The Embedded Pivot Section Interactivity Object Model enhancements include:

- "Embedded Pivot Section Interactivity" on page 137
- "OnCellDoubleClick (Method)" on page 138
- "SideLabelValues (Array)" on page 139
- "TopLabelValues (Array)" on page 140

The new "Printout (Method)" on page 148 has also been added in Release 8.5.

The "Object Model Exclusion List" on page 148 has been updated in Release 8.5.

UnicodeEnabled (Property)

Applies To: Application object

Description: This property is set to true in Unicode enabled applications.

Action: Read only

Example: This example shows how to read and show the value of the UnicodeEnabled (Property):

```
//Clear all TextBoxes except TextBox5
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ActiveSection.Name
var ActionName = "UnicodeEnabled"
TextBox1.Text ="Start " + ActionName
if (TextBox5.Text == "")
{
TextBox1.Text ="Step 1"
try
{
TextBox3.Text =  "UnicodeEnabled is: " + Application.UnicodeEnabled
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}

}

else
```

```
{
TextBox1.Text ="Step 2"
try
{
Application.UnicodeEnabled = eval(TextBox5.Text)
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
TextBox1.Text ="Step 3"
try
{
TextBox3.Text =  "UnicodeEnabled is: " + Application.UnicodeEnabled
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
}
TextBox1.Text ="End " + ActionName
```

SortOrderLang (Property)

Applies To: Document object

Description: This is a property of a BQY document and controls how language dependent data sorting is performed within the document sections. Its possible value is a member of BqLanguage enumeration:

- bqLangAfrikaans
- bqLangAlbanian
- bqLangArabic
- bqLangBasque
- bqLangCatalan
- bqLangChinese_Simp (pinyin sort)
- bqLangChinese_Trad (big5han sort)
- bqLangChinese_GB2312_Sort
- bqLangChinese_Stroke_Sort
- bqLangCroatian
- bqLangCzech
- bqLangDanish
- bqLangDutch
- bqLangEnglish
- bqLangUKEnglish

- bqLangEstonian
- bqLangFinnish
- bqLangFrench
- bqLangGerman
- bqLangGerman_Phonebook_Sort
- bqLangGreek
- bqLangHebrew
- bqLangHungarian
- bqLangIcelandic
- bqLangIndonesian
- bqLangItalian
- bqLangJapanese
- bqLangKorean
- bqLangLatvian
- bqLangLithuanian
- bqLangMacedonian
- bqLangMalay
- bqLangNorwegian_Bokmal
- bqLangNorwegian_Nynorsk
- bqLangPolish
- bqLangPortuguese
- bqLangRomanian
- bqLangRussian
- bqLangSerbian_Cyrillic
- bqLangSerbian_Latin
- bqLangSlovak
- bqLangSlovenian
- bqLangSpanish
- bqLangSpanish_Trad_Sort
- bqLangSwedish
- bqLangThai
- bqLangTurkish
- bqLangUkrainian
- bqLangVietnamese

Action: Read/Write

DefaultSortOrderLang (Property)

Applies To: Application object

Description: The DefaultSortOrderLang (Property) sets the default sort order language format that is applied to an existing BQY document when a document is opened in a Unicode enabled application. If no default language is set (the check box is not enabled) in the user interface, this property returns 0, which does not correspond to any valid language. The sort order language of a BQY document may be changed in order to start sorting the data according to UI language preferences.

The Application .ini file stores this property in the DefDocLanguage property of [Regional Settings]. It is interchangeable between all applications including Services.

Action: Read/Write

Example: This example shows how to read and write the value of a default sort order language in a BQY document.

```
//Clear all TextBoxes except TextBox5
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ActiveSection.Name
var ActionName = "DefaultSortOrderLang"
TextBox1.Text ="Start " + ActionName
if (TextBox5.Text == "")
{
TextBox1.Text ="Step 1"
try
{
TextBox3.Text =  "DefaultSortOrderLang is: " +
Application.DefaultSortOrderLang
switch(Application.DefaultSortOrderLang)
{
case 0:
TextBox3.Text = "No Language specified"
break;
case 1:
TextBox3.Text = "Language is bqLangAfrikaans"
break;
case 2:
TextBox3.Text = "Language is bqLangAlbanian"
break;
case 3:
TextBox3.Text = "Language is bqLangArabic"
break;
case 4:
TextBox3.Text = "Language is bqLangBasque"
break;
case 5:
```

```
TextBox3.Text = "Language is bqLangCatalan"
break;
case 6:
TextBox3.Text = "Language is bqLangChinese_Simp"
break;
case 7:
TextBox3.Text = "bqLangChinese_Trad"
break;
case 8:
TextBox3.Text = "Language is bqLangChinese_GB2312_Sort"
break;
case 9:
TextBox3.Text = "Language is bqLangChinese_Trad"
break;
case 10:
TextBox3.Text = "Language is bqLangCroatian"
break;
case 11:
TextBox3.Text = "Language is bqLangCzech"
break;
case 12:
TextBox3.Text = "Language is bqLangDanish"
break;
case 13:
TextBox3.Text = "Language is bqLangDutch"
break;
case 14:
TextBox3.Text = "Language is bqLangEnglish"
break;
case 15:
TextBox3.Text = "Language is bqLangUKEnglish"
break;
case 16:
TextBox3.Text = "Language is bqLangEstonian"
break;
case 17:
TextBox3.Text = "Language is bqLangFinnish"
break;
case 18:
TextBox3.Text = "Language is bqLangFrench"
break;
case 19:
TextBox3.Text = "Language is bqLangGerman"
break;
case 20:
TextBox3.Text = "Language is bqLangGerman_Phonebook_Sort"
break;
case 21:
TextBox3.Text = "Language is bqLangGreek"
break;
case 22:
TextBox3.Text = "Language is bqLangHebrew"
break;
case 23:
TextBox3.Text = "Language is bqLangHungarian"
break;
case 24:
TextBox3.Text = "Language is bqLangIcelandic"
```

```
break;
case 25:
TextBox3.Text = "Language is bqLangIndonesian"
break;
case 26:
TextBox3.Text = "Language is bqLangItalian"
break;
case 27:
TextBox3.Text = "Language is bqLangJapanese"
break;
case 28:
TextBox3.Text = "Language is bqLangKorean"
break;
case 29:
TextBox3.Text = "Language is bqLangLatvian"
break;
case 30:
TextBox3.Text = "Language is bqLangLithuanian"
break;
case 31:
TextBox3.Text = "Language is bqLangMacedonian"
break;
case 32:
TextBox3.Text = "Language is bqLangMalay"
break;
case 33:
TextBox3.Text = "Language is bqLangNorwegian_Bokmal"
break;
case 34:
TextBox3.Text = "Language is bqLangNorwegian_Nynorsk"
break;
case 35:
TextBox3.Text = "Language is bqLangPolish"
break;
case 36:
TextBox3.Text = "Language is bqLangPortuguese"
break;
case 37:
TextBox3.Text = "Language is bqLangRomanian"
break;
case 38:
TextBox3.Text = "Language is bqLangRussian"
break;
case 39:
TextBox3.Text = "Language is bqLangSerbian_Cyrillic"
break;
case 40:
TextBox3.Text = "Language is bqLangSerbian_Latin"
break;
case 41:
TextBox3.Text = "Language is bqLangSlovak"
break;
case 42:
TextBox3.Text = "Language is bqLangSlovenian"
break;
case 43:
TextBox3.Text = "Language is bqLangSpanish"
break;
```

```
case 44:
TextBox3.Text = "Language is bqLangSpanish_Trad_Sort"
break;
case 45:
TextBox3.Text = "Language is bqLangSwedish"
break;
case 46:
TextBox3.Text = "Language is bqLangThai"
break;
case 47:
TextBox3.Text = "Language is bqLangTurkish"
break;
case 48:
TextBox3.Text = "Language is bqLangUkrainian"
break;
case 49:
TextBox3.Text = "Language is bqLangVietnamese"
break;
default:
TextBox3.Text = "Language Value is Not Available"
}
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
}
else
{
TextBox1.Text ="Step 2"
try
{
Application.DefaultSortOrderLang = eval(TextBox5.Text)
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
TextBox1.Text ="Step 3"
try
{
TextBox3.Text =  "DefaultSortOrderLang is: " +
Application.DefaultSortOrderLang
switch(Application.DefaultSortOrderLang)
{
case 0:
TextBox3.Text = "No Language specified"
break;
case 1:
TextBox3.Text = "Language is bqLangAfrikaans"
break;
case 2:
TextBox3.Text = "Language is bqLangAlbanian"
break;
case 3:
TextBox3.Text = "Language is bqLangArabic"
break;
case 4:
TextBox3.Text = "Language is bqLangBasque"
```

```
break;
case 5:
TextBox3.Text = "Language is bqLangCatalan"
break;
case 6:
TextBox3.Text = "Language is bqLangChinese_Simp"
break;
case 7:
TextBox3.Text = "bqLangChinese_Trad"
break;
case 8:
TextBox3.Text = "Language is bqLangChinese_GB2312_Sort"
break;
case 9:
TextBox3.Text = "Language is bqLangChinese_Trad"
break;
case 10:
TextBox3.Text = "Language is bqLangCroatian"
break;
case 11:
TextBox3.Text = "Language is bqLangCzech"
break;
case 12:
TextBox3.Text = "Language is bqLangDanish"
break;
case 13:
TextBox3.Text = "Language is bqLangDutch"
break;
case 14:
TextBox3.Text = "Language is bqLangEnglish"
break;
case 15:
TextBox3.Text = "Language is bqLangUKEnglish"
break;
case 16:
TextBox3.Text = "Language is bqLangEstonian"
break;
case 17:
TextBox3.Text = "Language is bqLangFinnish"
break;
case 18:
TextBox3.Text = "Language is bqLangFrench"
break;
case 19:
TextBox3.Text = "Language is bqLangGerman"
break;
case 20:
TextBox3.Text = "Language is bqLangGerman_Phonebook_Sort"
break;
case 21:
TextBox3.Text = "Language is bqLangGreek"
break;
case 22:
TextBox3.Text = "Language is bqLangHebrew"
break;
case 23:
TextBox3.Text = "Language is bqLangHungarian"
break;
case 24:
TextBox3.Text = "Language is bqLangIcelandic"
```

```
break;
case 25:
TextBox3.Text = "Language is bqLangIndonesian"
break;
case 26:
TextBox3.Text = "Language is bqLangItalian"
break;
case 27:
TextBox3.Text = "Language is bqLangJapanese"
break;
case 28:
TextBox3.Text = "Language is bqLangKorean"
break;
case 29:
TextBox3.Text = "Language is bqLangLatvian"
break;
case 30:
TextBox3.Text = "Language is bqLangLithuanian"
break;
case 31:
TextBox3.Text = "Language is bqLangMacedonian"
break;
case 32:
TextBox3.Text = "Language is bqLangMalay"
break;
case 33:
TextBox3.Text = "Language is bqLangNorwegian_Bokmal"
break;
case 34:
TextBox3.Text = "Language is bqLangNorwegian_Nynorsk"
break;
case 35:
TextBox3.Text = "Language is bqLangPolish"
break;
case 36:
TextBox3.Text = "Language is bqLangPortuguese"
break;
case 37:
TextBox3.Text = "Language is bqLangRomanian"
break;
case 38:
TextBox3.Text = "Language is bqLangRussian"
break;
case 39:
TextBox3.Text = "Language is bqLangSerbian_Cyrillic"
break;
case 40:
TextBox3.Text = "Language is bqLangSerbian_Latin"
break;
case 41:
TextBox3.Text = "Language is bqLangSlovak"
break;
case 42:
TextBox3.Text = "Language is bqLangSlovenian"
break;
case 43:
TextBox3.Text = "Language is bqLangSpanish"
break;
case 44:
TextBox3.Text = "Language is bqLangSpanish_Trad_Sort"
```

```
break;
case 45:
TextBox3.Text = "Language is bqLangSwedish"
break;
case 46:
TextBox3.Text = "Language is bqLangThai"
break;
case 47:
TextBox3.Text = "Language is bqLangTurkish"
break;
case 48:
TextBox3.Text = "Language is bqLangUkrainian"
break;
case 49:
TextBox3.Text = "Language is bqLangVietnamese"
break;
default:
TextBox3.Text = "Language Value is Not Available"
}
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
}
TextBox1.Text ="End " + ActionName
```

UILanguage (Property)

Applies To: Application object

Description: Defines the localization language of the application. This is the language of menus and messages. Its value is a member of the BqLanguage constant group.

Note: The UILanguage (Property) is not available in a BQY to be deployed in the Intelligence iServer, or in the Hyperion Intelligence Web Client (available offline only).

Action: Read/Write

Constants: The BqLanguage constant group consists of the following values:

- bqLangArabic
- bqLangBasque
- bqLangCatalan
- bqLangChinese_GB2312_Sort
- bqLangChinese_Simp

- bqLangChinese_Stroke_Sort
- bqLangChinese_Trad
- bqLangCroatian
- bqLangCzech
- bqLangDanish
- bqLangDutch
- bqLangEnglish
- bqLangEstonian
- bqLangFinnish
- bqLangFrench
- bqLangGerman
- bqLangGerman_Phonebook_Sort
- bqLangGreek
- bqLangHebrew
- bqLangHungarian
- bqLangIcelandic
- bqLangIndonesian
- bqLangItalian
- bqLangJapanese
- bqLangKorean
- bqLangLatvian
- bqLangLithuanian
- bqLangMacedonian
- bqLangMalay
- bqLangNorwegian_Bokmal
- bqLangNorwegian_Nynorsk
- bqLangPolish
- bqLangPortuguese
- bqLangRomanian
- bqLangRussian
- bqLangSerbian_Cyrillic
- bqLangSerbian_Latin
- bqLangSlovak
- bqLangSlovenian
- bqLangSpanish

- bqLangSpanish_Trad_Sort
- bqLangSwedish
- bqLangThai
- bqLangTurkish
- bqLangUKEnglish
- bqLangUkrainian
- bqLangVietnamese

Example: This example shows how to read and write the value of the user interface language in a BQY document.

```
//Clear all TextBoxes except TextBox5
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ActiveSection.Name

var ActionName = "UILanguage"

TextBox1.Text ="Start " + ActionName

if (TextBox5.Text == "")
{
TextBox1.Text ="Step 1"

try
{
TextBox3.Text =  "UILanguage is: " + Application.UILanguage

switch(Application.UILanguage)
{
case 0:
TextBox3.Text = "No Language specified"
break;
case 1:
TextBox3.Text = "Language is bqLangAfrikaans"
break;
case 2:
TextBox3.Text = "Language is bqLangAlbanian"
break;
```

```
case 3:
TextBox3.Text = "Language is bqLangArabic"
break;
case 4:
TextBox3.Text = "Language is bqLangBasque"
break;
case 5:
TextBox3.Text = "Language is bqLangCatalan"
break;
case 6:
TextBox3.Text = "Language is bqLangChinese_Simp"
break;
case 7:
TextBox3.Text = "bqLangChinese_Trad"
break;
case 8:
TextBox3.Text = "Language is bqLangChinese_GB2312_Sort"
break;
case 9:
TextBox3.Text = "Language is bqLangChinese_Trad"
break;
case 10:
TextBox3.Text = "Language is bqLangCroatian"
break;
case 11:
TextBox3.Text = "Language is bqLangCzech"
break;
case 12:
TextBox3.Text = "Language is bqLangDanish"
break;
case 13:
TextBox3.Text = "Language is bqLangDutch"
break;
case 14:
TextBox3.Text = "Language is bqLangEnglish"
break;
case 15:
TextBox3.Text = "Language is bqLangUKEnglish"
break;
```

```
case 16:
TextBox3.Text = "Language is bqLangEstonian"
break;
case 17:
TextBox3.Text = "Language is bqLangFinnish"
break;
case 18:
TextBox3.Text = "Language is bqLangFrench"
break;
case 19:
TextBox3.Text = "Language is bqLangGerman"
break;
case 20:
TextBox3.Text = "Language is bqLangGerman_Phonebook_Sort"
break;
case 21:
TextBox3.Text = "Language is bqLangGreek"
break;
case 22:
TextBox3.Text = "Language is bqLangHebrew"
break;
case 23:
TextBox3.Text = "Language is bqLangHungarian"
break;
case 24:
TextBox3.Text = "Language is bqLangIcelandic"
break;
case 25:
TextBox3.Text = "Language is bqLangIndonesian"
break;
case 26:
TextBox3.Text = "Language is bqLangItalian"
break;
case 27:
TextBox3.Text = "Language is bqLangJapanese"
break;
case 28:
TextBox3.Text = "Language is bqLangKorean"
break;
```

```
case 29:
TextBox3.Text = "Language is bqLangLatvian"
break;
case 30:
TextBox3.Text = "Language is bqLangLithuanian"
break;
case 31:
TextBox3.Text = "Language is bqLangMacedonian"
break;
case 32:
TextBox3.Text = "Language is bqLangMalay"
break;
case 33:
TextBox3.Text = "Language is bqLangNorwegian_Bokmal"
break;
case 34:
TextBox3.Text = "Language is bqLangNorwegian_Nynorsk"
break;
case 35:
TextBox3.Text = "Language is bqLangPolish"
break;
case 36:
TextBox3.Text = "Language is bqLangPortuguese"
break;
case 37:
TextBox3.Text = "Language is bqLangRomanian"
break;
case 38:
TextBox3.Text = "Language is bqLangRussian"
break;
case 39:
TextBox3.Text = "Language is bqLangSerbian_Cyrillic"
break;
case 40:
TextBox3.Text = "Language is bqLangSerbian_Latin"
break;
case 41:
TextBox3.Text = "Language is bqLangSlovak"
break;
```

```
case 42:
TextBox3.Text = "Language is bqLangSlovenian"
break;
case 43:
TextBox3.Text = "Language is bqLangSpanish"
break;
case 44:
TextBox3.Text = "Language is bqLangSpanish_Trad_Sort"
break;
case 45:
TextBox3.Text = "Language is bqLangSwedish"
break;
case 46:
TextBox3.Text = "Language is bqLangThai"
break;
case 47:
TextBox3.Text = "Language is bqLangTurkish"
break;
case 48:
TextBox3.Text = "Language is bqLangUkrainian"
break;
case 49:
TextBox3.Text = "Language is bqLangVietnamese"
break;
default:
TextBox3.Text = "Language Value is Not Available"
}

}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}

}

else
{
TextBox1.Text ="Step 2"
```

```
try
{
Application.UILanguage = eval(TextBox5.Text)
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}


TextBox1.Text ="Step 3"

try
{
TextBox3.Text =  "UILanguage is: " + Application.UILanguage

switch(Application.UILanguage)
{
case 0:
TextBox3.Text = "No Language specified"
break;
case 1:
TextBox3.Text = "Language is bqLangAfrikaans"
break;
case 2:
TextBox3.Text = "Language is bqLangAlbanian"
break;
case 3:
TextBox3.Text = "Language is bqLangArabic"
break;
case 4:
TextBox3.Text = "Language is bqLangBasque"
break;
case 5:
TextBox3.Text = "Language is bqLangCatalan"
break;
case 6:
TextBox3.Text = "Language is bqLangChinese_Simp"
break;
```

```
case 7:
TextBox3.Text = "bqLangChinese_Trad"
break;
case 8:
TextBox3.Text = "Language is bqLangChinese_GB2312_Sort"
break;
case 9:
TextBox3.Text = "Language is bqLangChinese_Trad"
break;
case 10:
TextBox3.Text = "Language is bqLangCroatian"
break;
case 11:
TextBox3.Text = "Language is bqLangCzech"
break;
case 12:
TextBox3.Text = "Language is bqLangDanish"
break;
case 13:
TextBox3.Text = "Language is bqLangDutch"
break;
case 14:
TextBox3.Text = "Language is bqLangEnglish"
break;
case 15:
TextBox3.Text = "Language is bqLangUKEnglish"
break;
case 16:
TextBox3.Text = "Language is bqLangEstonian"
break;
case 17:
TextBox3.Text = "Language is bqLangFinnish"
break;
case 18:
TextBox3.Text = "Language is bqLangFrench"
break;
case 19:
TextBox3.Text = "Language is bqLangGerman"
break;
```

```
case 20:
TextBox3.Text = "Language is bqLangGerman_Phonebook_Sort"
break;
case 21:
TextBox3.Text = "Language is bqLangGreek"
break;
case 22:
TextBox3.Text = "Language is bqLangHebrew"
break;
case 23:
TextBox3.Text = "Language is bqLangHungarian"
break;
case 24:
TextBox3.Text = "Language is bqLangIcelandic"
break;
case 25:
TextBox3.Text = "Language is bqLangIndonesian"
break;
case 26:
TextBox3.Text = "Language is bqLangItalian"
break;
case 27:
TextBox3.Text = "Language is bqLangJapanese"
break;
case 28:
TextBox3.Text = "Language is bqLangKorean"
break;
case 29:
TextBox3.Text = "Language is bqLangLatvian"
break;
case 30:
TextBox3.Text = "Language is bqLangLithuanian"
break;
case 31:
TextBox3.Text = "Language is bqLangMacedonian"
break;
case 32:
TextBox3.Text = "Language is bqLangMalay"
break;
```

```
case 33:
TextBox3.Text = "Language is bqLangNorwegian_Bokmal"
break;
case 34:
TextBox3.Text = "Language is bqLangNorwegian_Nynorsk"
break;
case 35:
TextBox3.Text = "Language is bqLangPolish"
break;
case 36:
TextBox3.Text = "Language is bqLangPortuguese"
break;
case 37:
TextBox3.Text = "Language is bqLangRomanian"
break;
case 38:
TextBox3.Text = "Language is bqLangRussian"
break;
case 39:
TextBox3.Text = "Language is bqLangSerbian_Cyrillic"
break;
case 40:
TextBox3.Text = "Language is bqLangSerbian_Latin"
break;
case 41:
TextBox3.Text = "Language is bqLangSlovak"
break;
case 42:
TextBox3.Text = "Language is bqLangSlovenian"
break;
case 43:
TextBox3.Text = "Language is bqLangSpanish"
break;
case 44:
TextBox3.Text = "Language is bqLangSpanish_Trad_Sort"
break;
case 45:
TextBox3.Text = "Language is bqLangSwedish"
break;
case 46:
```

```
TextBox3.Text = "Language is bqLangThai"
break;
case 47:
TextBox3.Text = "Language is bqLangTurkish"
break;
case 48:
TextBox3.Text = "Language is bqLangUkrainian"
break;
case 49:
TextBox3.Text = "Language is bqLangVietnamese"
break;
default:
TextBox3.Text = "Language Value is Not Available"
}




}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}

}

TextBox1.Text ="End " + ActionName
```

ResetDefaultSortOrderLang (Method)

Applies To: Application object

Description: Clears the current sort order language. This action is the equivalent of disabling the Use this choice by default on the International tab of the Properties dialog box.

Example: This example shows how to use the ResetDefaultSortOrderLang (Method) to clear the default sort order language.

//Clear all textboxes except TextBox5

TextBox1.Text = ""

```
TextBox1.Text = ""
TextBox1.Text = ""
TextBox4.Text = ActiveSection.Name
var ActionName = "ResetDefaultSortOrderLang new"
TextBox1.Text ="Start " + ActionName
try
{
Application.ResetDefaultSortOrderLang()
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}

TextBox1.Text ="End " + ActionName
```

Export (Method)

Applies To: ChartSection object, DataModelSection object, Document object, DashboardSection object, OLAPQuerySection object, PivotSection object, QuerySection object, ResultsSection object, Section object, TableSection object, ReportSection object

Description: Creates a new file with the information from a section object. Files can be created using the standard data formats from the BqExportFileFormat constant group.

Note: For UNIX users of the Export and ExportToStream methods, Hyperion Intelligence references the umask command of the user running the BIService to determine the three-digit octal code. This code defines the read-write-execute permissions to be turned off. Typically, umask is used in .login or .profile and in the Bourne and C shells. It is a built-in command.

Syntax: Expression.Export(Filename As String, [optional] FileFormat As BqExportFileFormat, [optional]IncludeHeaders As Boolean], [optional] Boolean Prompt, [optional] BqEncoding Encoding)

Expression Required: An expression that returns an object for any of the following sections:

- ChartSection
- DataModelSection

- DashboardSection
- OLAPQuerySection
- PivotSection
- QuerySection
- Section
- TableSection

Constants: The BqExportFileFormat constant group consists of the following values:

- bqExportFormatCSV
- bqExportFormatExcel2
- bqExportFormatExcel5
- bqExportFormatHTML
- bqExportFormatJPEG
- bqExportFormatLotus123
- bqExportFormatPDF
- bqExportFormatText
- bqExportFormatOfficeHTML
- bqExportFormatOfficeMHTML

The BQEncoding constant group values include:

- bqEnc_Arabic_ISO
- bqEnc_Arabic_Windows
- bqEnc_Baltic_ISO
- bqEnc_Baltic_ISO
- bqEnc_Baltic_Windows
- bqEnc_CentralEuropean_ISO
- bqEnc_CentralEuropean_Windows
- bqEnc_Chinese_BIG5
- bqEnc_Chinese_GB2312
- bqEnc_Chinese_GBK
- bqEnc_Chinese_HZ
- bqEnc_Cyrillic_DOS
- bqEnc_Cyrillic_ISO
- bqEnc_Cyrillic_KOI8R
- bqEnc_Cyrillic_KOI8U

- bqEnc_Cyrillic_Windows
- bqEnc_Greek_ISO
- bqEnc_Greek_Windows
- bqEnc_Hebrew_Windows
- bqEnc_Japanese_EUC
- bqEnc_Japanese_JIS
- bqEnc_Japanese_ShiftJIS
- bqEnc_Korean
- bqEnc_Thai_Windows
- bqEnc_Turkish_ISO
- bqEnc_Turkish_Windows
- bqEnc_Unicode
- bqEnc_Unicode_UTF16
- bqEnc_Unicode_UTF16_BigEndian
- bqEnc_Unicode_UTF16_LittleEndian
- bqEnc_Unicode_UTF32
- bqEnc_Unicode_UTF32_BigEndian
- bqEnc_Unicode_UTF32_LittleEndian
- bqEnc_Unicode_UTF8
- bqEnc_Vietnamese_Windows
- bqEnc_WesternEuropean_ISO
- bqEnc_WesternEuropean_Windows

For information on exporting a BQY to HTML format statically, see HTMLExportBreakRowCount (Property) and HTMLVerticalPageBreakUnits (Property) in the *Object Model Guide to Properties and Constants*.

For information on exporting a BQY to the HTML format dynamically (for use with the Hyperion System 9 BI + Workspace), see HTMLPregenerateMode (Property), HTMLPregenerateFirstLastNPages (Property), DHTMLExportBreakRowCount (Property), and DHTMLVerticalPageBreakUnits (Property) in the *Object Model Guide to Properties and Constants*.

Example: This example shows how to export a Results section to HTML. The first part of the script creates a computed column that displays the contents of the *URL* columns as HTML HREFs:

```
//Call the JavaScript link() method to convert the string to HREFs
var ComputedExpression = "URL.link()"
ActiveDocument.Sections["Results"].Columns.AddComputed("Clickable
URLS",ComputedExpression)
ActiveDocument.Sections["Results"].Export("C:\\HTML\\MyResults.htm",
bqExportFormatHTML,false)
```

ExportToStream (Method)

Applies To: ChartSection object, DataModelSection ojbect, Document object, DashboardSection object, OLAPQuerySection object, PivotSection object, QuerySection object, ResultsSection object, Section object, TableSection object, ReportSection object

Description: The ExportToStream (Method) enables you to use data streaming in Hyperion System 9 BI + Workspace. Streamed data displays before the entire file is exported. This feature improves performance.

If the the file name and associated path information are specified when this method, this information is ignored when streaming is enabled. In the case of the full clients (Designer/Explorer), the file name and associated path information are used for writing data to disk with no additional errors cited when streaming is enabled (as in the case of export without streaming).

Note: For UNIX users of the Export and ExportToStream methods, Hyperion Intelligence references the umask command of the user running the BIService to determine the three-digit octal code. This code defines the read-write-execute permissions to be turned off. Typically, umask is used in .login or .profile and in the Bourne and C shells. It is a built-in command.

Syntax:

```
Expression.Export( [optional]String Filename,
[optional]BqExportFileFormat FileFormat,
[optional]Boolean IncludeHeaders,
[optional]Boolean DataStreaming,
[optional]Boolean Prompt, [optional] BqEncoding Encoding)
```

Expression Required: An expression that returns an object for any of the following sections:

- ChartSection
- DataModelSection
- DashboardSection
- OLAPQuerySection
- PivotSection
- QuerySection
- Section
- TableSection

Constants: The ExportToStream (Method) uses the BqExportFileFormat constant group, which consists of the following values:

- bqExportFormatCSV
- bqExportFormatExcel2

- bqExportFormatExcel5
- bqExportFormatHTML
- bqExportFormatJPEG
- bqExportFormatLotus123
- bqExportFormatPDF
- bqExportFormatText
- bqExportFormatOfficeHTML
- bqExportFormatOfficeMHTML

The BQEncoding constant group values include:

- bqEnc_Arabic_ISO
- bqEnc_Arabic_Windows
- bqEnc_Baltic_ISO
- bqEnc_Baltic_Windows
- bqEnc_CentralEuropean_ISO
- bqEnc_CentralEuropean_Windows
- bqEnc_Chinese_BIG5
- bqEnc_Chinese_GB2312
- bqEnc_Chinese_GBK
- bqEnc_Chinese_HZ
- bqEnc_Cyrillic_DOS
- bqEnc_Cyrillic_ISO
- bqEnc_Cyrillic_KOI8R
- bqEnc_Cyrillic_KOI8U
- bqEnc_Cyrillic_Windows
- bqEnc_Greek_ISO
- bqEnc_Greek_Windows
- bqEnc_Hebrew_Windows
- bqEnc_Japanese_EUC
- bqEnc_Japanese_JIS
- bqEnc_Japanese_ShiftJIS
- bqEnc_Korean
- bqEnc_Thai_Windows
- bqEnc_Turkish_ISO
- bqEnc_Turkish_Windows
- bqEnc_Unicode
- bqEnc_Unicode_UTF16
- bqEnc_Unicode_UTF16_BigEndian

- bqEnc_Unicode_UTF16_LittleEndian
- bqEnc_Unicode_UTF32
- bqEnc_Unicode_UTF32_BigEndian
- bqEnc_Unicode_UTF32_LittleEndian
- bqEnc_Unicode_UTF8
- bqEnc_Vietnamese_Windows
- bqEnc_WesternEuropean_ISO
- bqEnc_WesternEuropean_Windows

Example: This example shows how to export a Results section in a data stream to HTML. Because the streaming is enabled, no file name and associated path information have been specified:

```
ActiveDocument.Sections["Results"].ExportToStream("", bqExportFormatHTML, false, true, false)
```

ImportDataFile (Method)

Applies To: Sections collection, WebClientDocument object

Description: Imports a data file into a Query section.

Syntax: Expression.ImportDataFile(Filename As String, FileType As BqImportDataFileFormat, [optional] BqEncoding Encoding)

Expression Required: An expression that returns a **Sections** object

Constants: The BqImportDataFileFormat constant group contains the following values:

- bqImportFormatCommaText
- bqImportFormatExcel
- bqImportFormatTabText

The BQEncoding constant group values include:

- bqEnc_Arabic_ISO
- bqEnc_Arabic_Windows
- bqEnc_Baltic_ISO
- bqEnc_Baltic_Windows
- bqEnc_CentralEuropean_ISO
- bqEnc_CentralEuropean_Windows
- bqEnc_Chinese_BIG5

- bqEnc_Chinese_GB2312
- bqEnc_Chinese_GBK
- bqEnc_Chinese_HZ
- bqEnc_Cyrillic_DOS
- bqEnc_Cyrillic_ISO
- bqEnc_Cyrillic_KOI8R
- bqEnc_Cyrillic_KOI8U
- bqEnc_Cyrillic_Windows
- bqEnc_Greek_ISO
- bqEnc_Greek_Windows
- bqEnc_Hebrew_Windows
- bqEnc_Japanese_EUC
- bqEnc_Japanese_JIS
- bqEnc_Japanese_ShiftJIS
- bqEnc_Korean
- bqEnc_Thai_Windows
- bqEnc_Turkish_ISO
- bqEnc_Turkish_Windows
- bqEnc_Unicode
- bqEnc_Unicode_UTF16
- bqEnc_Unicode_UTF16_BigEndian
- bqEnc_Unicode_UTF16_LittleEndian
- bqEnc_Unicode_UTF32
- bqEnc_Unicode_UTF32_BigEndian
- bqEnc_Unicode_UTF32_LittleEndian
- bqEnc_Unicode_UTF8
- bqEnc_Vietnamese_Windows
- bqEnc_WesternEuropean_ISO
- bqEnc_WesternEuropean_Windows

Example: This example shows how to import a comma separated data file:

```
var Filename = "C:\\Imports\\SalesData.csv"
var MySection = ActiveDocument.Sections.ImportDataFile(Filename,
bqImportFormatCommaText)
```

ImportSQLFile (Method)

Applies To: QuerySection

Description: Imports a complete SQL statement from a text file into an existing query, and retrieves the data set from the database server. When you import a file, it is scanned to determine the number of columns that will be returned by the SQL, with the request line becoming populated with a column indicator for each of the columns. Using this feature, you can take advantage of SQL statements that you have already written.

Before using this method, be sure that you are connected to a database server. The Query section to which you are importing the SQL must have no tables. In addition, the SQL file to be imported must begin with a SELECT statement and you should know the number of columns to be displayed in the Results section. When the SQL file has been imported into the query, you can drag items from the table to the Request line, use the Custom SQL feature, or display its properties.The imported SQL file cannot be edited, but you can specify a user-friendly name for the Request line item and identify its data type.

Syntax: Expression.ImportSQLFile(Filename As String,numColumns As Number, [optional] BqEncoding Encoding)

Expression Required: An expression that returns a Query object

Constants: The BQEncoding constant group values include:

- bqEnc_Arabic_ISO
- bqEnc_Arabic_Windows
- bqEnc_Baltic_ISO
- bqEnc_Baltic_Windows
- bqEnc_CentralEuropean_ISO
- bqEnc_CentralEuropean_Windows
- bqEnc_Chinese_BIG5
- bqEnc_Chinese_GB2312
- bqEnc_Chinese_GBK
- bqEnc_Chinese_HZ
- bqEnc_Cyrillic_DOS
- bqEnc_Cyrillic_ISO
- bqEnc_Cyrillic_KOI8R
- bqEnc_Cyrillic_KOI8U
- bqEnc_Cyrillic_Windows
- bqEnc_Greek_ISO
- bqEnc_Greek_Windows

- bqEnc_Hebrew_Windows
- bqEnc_Japanese_EUC
- bqEnc_Japanese_JIS
- bqEnc_Japanese_ShiftJIS
- bqEnc_Korean
- bqEnc_Thai_Windows
- bqEnc_Turkish_ISO
- bqEnc_Turkish_Windows
- bqEnc_Unicode
- bqEnc_Unicode_UTF16
- bqEnc_Unicode_UTF16_BigEndian
- bqEnc_Unicode_UTF16_LittleEndian
- bqEnc_Unicode_UTF32
- bqEnc_Unicode_UTF32_BigEndian
- bqEnc_Unicode_UTF32_LittleEndian
- bqEnc_Unicode_UTF8
- bqEnc_Vietnamese_Windows
- bqEnc_WesternEuropean_ISO
- bqEnc_WesternEuropean_Windows

Example: This example shows how to set the imported SQL file name, and process the query:

```
var Filename = "C:\\Program
Files\\Hyperion\\BIPlus\\docs\\Samples\\SQLLoad\\SalesData.sql"
var MySection = ActiveDocument.Sections["Query"].ImportSQLFile(Filename, 2)
ActiveDocument.Sections["Query"].Process()
```

Embedded Pivot Section Interactivity

Pivot tables embedded as active sections in a Dashboard can be scripted to enable users to double click on a cell and build interactivity around that cell. This feature is available on any individual fact cell in the pivot. When the fact cell is double-clicked, the OnCellDoubleClick event is fired and any JavaScript associated with the event can be executed. The event is associated with the object in the Dashboard, not with the Pivot section itself.

Embedded pivot section interactivity is available for all Hyperion Intelligence products, but not the Intelligence iServer.

Prior to executing the script associated with the OnCellDoubleClick event, certain properties of the object tree must be configured. Once set, these properties are valid from anywhere within the Object Model that the object is accessible, such as document startup and shutdown scripts. If the Pivot section is recalculated, these properties are reset to their initial, default state; as if no double click action had previously occurred.

To accommodate embedded pivot section interactivity, the following object, methods, and properties have been added:

- OnCellDoubleClick (Method)
- SideLabelValues (Array)
- TopLabelValues (Array)
- CellValue (Property)
- FactName (Property)

OnCellDoubleClick (Method)

Applies To: Dashboard Pivot Embedded Section Object (ESO) in Active Mode only

Description: Setting the OnCellDoubleClick (Method) is equivalent to repeating the action of clicking the last cell clicked in the user interface. This method performs an action that does not persist with (dirty) the document or application. This method does not allow the user to select and individual cell. If the user single clicks on cells, the selected rows are highlighted in black. If the user double clicks on cells, the selected rows are highlighted in white.

The OnCellDoubleClick (Method) is only available for the Hyperion Intelligence Client and Hyperion Intelligence Web Client. It is not supported in the Intelligence iServer and Jobs and Scheduler.

Syntax: `No arguments`

Example: This example shows how to execute an OnCellDoubleClick (Method) when a command button is clicked. The script includes a try-catch statement if the action is tried and then fails:

```
var ActionName = "OnCellDoubleClick"

TextBox1.Text ="Start " + ActionName

try

{

ActiveSection.Shapes["Pivot1"].OnCellDoubleClick()

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

TextBox1.Text ="End " + ActionName
```

SideLabelValues (Array)

Member of: Dashboard Pivot Embedded Section Object (ESO) in Active Mode only

Description: The SideLabelValues (Array) returns a JavaScript array (and not a Hyperion Intelligence Object Model array) of side label values. Each array element contains the value of the label at successively lower levels. For example, if there are two side labels in the Pivot, Year and Month, then the array element at index 1 in the array might contain "2001" and the next element at index 2 would contain a value like "November". Because JavaScript arrays are zero-based, the element at index 0 (zero) contains an empty string value.

The SideLabelValues (Array) is accessible for a Dashboard Embedded Section Object (ESO) only. It is available even if the Pivot contains no data. In addition, it is functional only when the pivot is in Active mode.

The SideLabelValues (Array) is refreshed each time a Pivot ESO cell is clicked in the user interface, or the OnCellDoubleClick() method is used. An array is used in this case due to the uncertainty of the SideLabel hierarchy at any given time. The array forces the user to iterate through and get the value each time to ensure it is up to date. It avoids purposefully the standard x.y.z.name structure of an object, which could become ambiguous and/or incorrect with any change of the Pivot side labels.

The array is read-only. An attempt to change the values in the array has no effect, scripts continue and this action does not change any of the pivot side label values.

The value returned by the array is a string which reflects the names of the pivot side labels in sequence. The number of values is returned using the length accessor. JavaScript arrays are normally 0-based, but for purposes of consistency with other Object Model objects, the [0] value always returns an empty string. If there are three side labels, the array returns …SideLabelValues[0] =<empty string>, …SideLabelValues[1] =<First Side Label Name>, …SideLabelValues[2] =<Second Side Label Name>, …SideLabelValues[3] =<Last Side Label Name>.

When no pivot side labels exist, the SideLabelValues[1] returns the value "undefined"; however no exception is generated and scripts continue to execute. The same behavior occurs when a value exceeding the length of the array is called. The array functions the same whether or not data exists at the selected intersection.

Any value returned by the array is not formatted. Any formatting that you want applied to the value should be scripted additionally.

Iterating through the SideLabelValues (Array) itself does not persists in (dirty) the document, nor does retrieving the value. The value in the array reflects the names of the Pivot side labels in sequence. It does not persist separately from the Pivot values in the document, nor does it persist within the application.

The SideLabelValues (Array) is available in the Hyperion Intelligence Client and Hyperion Intelligence Web Client. It is not supported in the Intelligence iServer and Jobs and Scheduler.

Note: There is no graphic to identify an individual selected cell using this array.

Example: This example shows how to access top label array data:

```
var ActionName = "SideLabelValues Array"

TextBox1.Text ="Start " + ActionName

try

{

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nSideLabelValues Default is: " +
ActiveSection.Shapes["Pivot1"].SideLabelValues

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nSideLabelValues length is: " +
ActiveSection.Shapes["Pivot1"].SideLabelValues.length

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nSideLabelValues[0] is: " +
ActiveSection.Shapes["Pivot1"].SideLabelValues[0]

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nSideLabelValues[1] is: " +
ActiveSection.Shapes["Pivot1"].SideLabelValues[1]

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nSideLabelValues[2] is: " +
ActiveSection.Shapes["Pivot1"].SideLabelValues[2]

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nSideLabelValues[3] is: " +
ActiveSection.Shapes["Pivot1"].SideLabelValues[3]

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

TextBox1.Text ="End " + ActionName
```

The results of the above script are displayed as follows:

```
SideLabelValues Default is: ,4
SideLabelValues length is: 2
SideLabelValues[0] is:
SideLabelValues[1] is: 4
SideLabelValues[2] is: undefined
SideLabelValues[3] is: undefined
```

TopLabelValues (Array)

Member of: Dashboard Pivot Embedded Section Object (ESO) in Active Mode only

Description: The TopLabelValues (Array) returns a JavaScript array (and not an Hyperion Intelligence Object Model array) of top label values. Each array element contains the value of the label at successively lower levels. For example, if there are two top labels in the Pivot, Year and Month, then the array element at index 1 in the array might contain “2001” and the next element at index 2 would contain a value like “November”. Because JavaScript arrays are zero-based, the element at index 0 (zero) contains an empty string value.

The TopLabelValues (Array) is accessible for a Dashboard Embedded Section Object (ESO) only. It is available even if the pivot contains no data. In addition, it is only functional when the pivot is in Active mode.

The TopLabelValues (Array) is refreshed each time a Pivot ESO cell is clicked in the user interface, or the OnCellDoubleClick() method is used. An array is used in this case due to the uncertainty of the TopLabel hierarchy at any given time. The array forces the user to iterate through and get the value each time to ensure it is up to date. It purposefully avoids the standard x.y.z.name structure of an object which could become ambiguous and/or incorrect with any change of the Pivot Top Labels.

The array is read-only. An attempt to change the values in the array has no effect, scripts continue and this action does not change any of the pivot top label values.

The value returned by the array should be a string which reflects the names of the Pivot Top Labels in sequence. The number of values is returned using the length accessor. JavaScript arrays are normally 0-based, but for purposes of consistency with other Object Model objects, the [0] value always returns an empty string. If there are three Top labels, the array returns …TopLabelValues[0] =<empty string>, …TopLabelValues[1] =<First Top Label Name>, …TopLabelValues[2] =<Second Top Label Name>, …TopLabelValues[3] =<Last Top Label Name>.

When no pivot top label sexist, the TopLabelValues[1] returns the value “undefined”; however no exception is generated and scripts continue to execute. The same behavior occurs when a value exceeding the length of the array is called. The array functions the same whether or not data exists at the selected intersection.

Any value returned by the array is not formatted. Any formatting that you want applied to the value should be scripted additionally.

Iterating through the TopLabelValues (Array) itself does not persist in (dirty) he document, nor does retrieving the value. The value in the array reflects the names of the pivot top labels in sequence. It does not persist separately from the pivot values in the document, nor does it persist within the application.

The TopLabelValues (Array) is available in the Hyperion Intelligence Client and Hyperion Intelligence Web Client. It is not supported in the Intelligence iServer and Jobs and Scheduler.

Note: There is no graphic to identify an individual selected cell using this array.

Example: This example shows how to access top label array data:

var ActionName = "TopLabelValues Array"

```
TextBox1.Text ="Start " + ActionName

try

{

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nTopLabelValues Default is: " +
ActiveSection.Shapes["Pivot1"].TopLabelValues

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nTopLabelValues length is: " +
ActiveSection.Shapes["Pivot1"].TopLabelValues.length

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nTopLabelValues[0] is: " +
ActiveSection.Shapes["Pivot1"].TopLabelValues[0]

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nTopLabelValues[1] is: " +
ActiveSection.Shapes["Pivot1"].TopLabelValues[1]

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nTopLabelValues[2] is: " +
ActiveSection.Shapes["Pivot1"].TopLabelValues[2]

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nTopLabelValues[3] is: " +
ActiveSection.Shapes["Pivot1"].TopLabelValues[3]

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

TextBox1.Text ="End " + ActionName
```

The results of the script are displayed as follows:

```
TopLabelValues Default is: ,15
TopLabelValues length is: 2
TopLabelValues[0] is:
TopLabelValues[1] is: 15
TopLabelValues[2] is: undefined
TopLabelValues[3] is: undefined
```

CellValue (Property)

Applies To: Embedded Pivot Objects

Description: The CellValue (Property) returns the value in the cell which was double-clicked. It is accessible for the Dashboard Pivot Embedded Sections Objects (ESOs). This property is only available when a Dashboard section containing a Pivot ESO is included in a BQY. It is only functional when the Pivot is in Active mode.

Reading the CellValue property is equivalent to noting the value when clicking an individual cell in a Pivot Embedded Section Object. The CellValue Property is read-only. The property returns a Value of type VARIANT representing the unformatted JavaScript content of the cell. The user must format it using JavaScript. The default value of the property is the value of the cell selected.

When no data exists at the intersection, a catchable “no such property” exception is returned, and the script halts without a “try/catch” statement being executed. The scripts continue only when the catch is employed.

The CellValue (Property) is available for the Hyperion Intelligence Client and Hyperion Intelligence Web Client. Is is not available for the Intelligence iServer or the Jobs and Scheduler.

Action: Read only, undefined

Example: This example shows how to return the value of a cell that has been double-clicked. It includes a try-catch statement and the default and tostring values.

```
//Clear all TextBoxes except TextBox5

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

TextBox4.Text = ActiveSection.Name

var ActionName = "CellValue"

TextBox1.Text ="Start " + ActionName

if (TextBox5.Text == "")

{

TextBox1.Text ="Step 1a"

try

{
```

```
PivotESOTextBox.Text = "CellValueDefault is: " +
ActiveSection.Shapes["Pivot1"].CellValue

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

TextBox1.Text ="Step 1b"

try

{

PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nCellValuetoString is:
" + ActiveSection.Shapes["Pivot1"].CellValue.toString()

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

}

else

{

TextBox1.Text ="Step 2"

try

{

ActiveDocument.Sections["Pivot"].Facts.CellValue = TextBox5.Text

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

TextBox1.Text ="Step 3a"

try

{
```

```
PivotESOTextBox.Text = "CellValueDefault is: " +
ActiveSection.Shapes["Pivot1"].CellValue

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

TextBox1.Text ="Step 3b"

try

{

PivotESOTextBox.Text = "CellValuetoString() is: " + PivotESOTextBox.Text
+ "\r\n" + ActiveSection.Shapes["Pivot1"].CellValue.toString()

}

catch(e)

{

TextBox2.Text = "Caught: " + e.toString()

}

}

TextBox1.Text ="End " + ActionName
```

In this example, the following values are returned:

CellValueDefault is: 5361.3

CellValuetoString is: 5361

FactName (Property)

Applies To: Embedded Pivot Objects

Description: The FactName (Property) returns the name of the fact to which the value applies. For example, if the value in CellValue corresponds to the name "Amount Sales" in the Facts section of the Pivot Outliner, then FactName is set to "Amount Sales". When no Fact exists and the property value is retrieved, the property returns an empty string.

The FactName property is accessible for the Dashboard Pivot Embedded Sections Objects (ESOs). This property is only available when a Dashboard section containing a Pivot ESO is included in the BI document. It is only functional when the Pivot is in Active mode.

The FactName Property is read-only. The value returned are a string representing the unformatted JavaScript value of the cell. The default value of the property is the name of the fact associated with the cell selected.

The FactName (Property) is available for the Hyperion Intelligence Client and Hyperion Intelligence Web Client. Is is not available for the Workspace or the Jobs and Scheduler.

Action: Read only, String

Example: This example shows how to return the value of the fact that has been double-clicked. It includes a “try-catch” statements, and a call to the tostring value. In this example, “FactName is: Amount” is returned.

```
//Clear all TextBoxes except TextBox5
TextBox1.Text = ""
TextBox2.Text = ""
TextBox3.Text = ""
TextBox4.Text = ActiveSection.Name
var ActionName = "FactName"
TextBox1.Text ="Start " + ActionName
if (TextBox5.Text == "")
{
TextBox1.Text ="Step 1"
try
{
PivotESOTextBox.Text = PivotESOTextBox.Text + "\r\nFactName is: " +
ActiveSection.Shapes["Pivot1"].FactName
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
}
else
{
 TextBox1.Text ="Step 2"
try
{
ActiveDocument.Sections["Pivot"].Facts.FactName = TextBox5.Text
```

```
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
TextBox1.Text ="Step 3"
try
{
TextBox3.Text =  "Facts is: " + ActiveDocument.Sections["Pivot"].Facts.FactName
}
catch(e)
{
TextBox2.Text = "Caught: " + e.toString()
}
}
TextBox1.Text ="End " + ActionName
```

Printout (Method)

Applies To: Intelligence iServer object

Description: The Printout (Method) enables the data sent back to the client browser to include a PDF file as part of the data stream when the server finishes processing a script. This PDF data opens in a new window using the Adobe Acrobat Reader (version 7.0.1 or later required), and the Adobe Print dialog box is displayed. Users must reset the Print Range options in Adobe Acrobat Reader before clicking OK to initiate the printing.

Note that the script is run in its entirety before the PDF data is sent to the client browser. This may cause some unexpected behavior with existing documents, such as those with a script that displays an Alert dialog box saying "Print Complete", when in fact it has not even started. Script authors should be aware of these factors when designing the application.

Use of the Alert (Method) after the Printout (Method) prevents the Printout (Method) from being executed in the Intelligence iServer.

When used in the Intelligence iServer, the optional parameters of the method, such as pages to print, number of copies, etc., are ignored. These options can be specified in the Adobe Acrobat Reader's Print dialog.

The Printout() method is not supported in scripts that run as the result of Document events (such as OnStartup). It is also not supported for section-level activation and deactivation events. The Printout() method has no effect on jobs and does not cause a printout to occur. It is also not supported when used as a method of a section of type Query or Datamodel.

Action: Read-Write

Object Model Exclusion List

This section provides a list of those objects, methods and properties that you cannot reference when you design a BQY for deployment in the Intelligence iServer in Release 8.5. When a method or property on this list is encountered in a script, a warning is entered in an error log file and execution of the script continues if possible. If you need to include the syntax shown below, be sure to include it in a BQY designed for desktop viewing

- ActiveDocument.Close()
- ActiveDocument.SetODSPassword()
- ActiveDocument.ODSUsername (Set only)
- ActiveDocument.Save()
- ActiveDocument.SaveAs()
- ActiveDocument.Sections["SectionName"].Copy()
- ActiveDocument.Sections["Chart"].XLabels.DrillInto()
- ActiveDocument.Sections["Dashboard"].Shapes["TextBox1"].OnChange()
- ActiveDocument.Sections["Dashboard"].Shapes["TextBox1"].OnEnter()

- ActiveDocument.Sections["Dashboard"].Shapes["ListBox1"].OnDoubleClick()
- ActiveDocument.Sections["Dashboard"].Shapes["Results1"].OnRowDoubleClick()
- ActiveDocument.Sections["OLAPQuery"].Slicers.Add()– Only the case where the last argument VariableSlicer=TRUE. Ignores the
- last argument, always defaulting to VariableSlicer=FALSE ActiveDocument.Sections["Query"].DataModel.Connection.DBLibAllowChangeDatabase
- ActiveDocument.Sections["Query"].DataModel.Connection.DBLibApiSeverity
- ActiveDocument.Sections["Query"].DataModel.Connection.DBLibDatabaseCancel
- ActiveDocument.Sections["Query"].DataModel.Connection.DBLibPacketSize
- ActiveDocument.Sections["Query"].DataModel.Connection.DBLibServerSeverity
- ActiveDocument.Sections["Query"].DataModel.Connection.DBLibUseQuotedIdentifiers
- ActiveDocument.Sections["Query"].DataModel.Connection.DBLibUseSQLTable
- ActiveDocument.Sections["Query"].DataModel.Connection.SaveWithoutUsername
- Application.CreateConnection()
- Application.DoEvents()
- Application.LoadSharedLibrary()
- Application.Shell()
- Application.ShowMenuBar()
- Application.Quit()
- Application.UILanguage() (set only)
- Console.Write()
- Console.WriteLn()
- Document.Close()
- Documents.Open()
- JOOLE Objects

Dashboard Export to HTML (Dashboards only exports .JPG in other client applications)

The following properties do not have any user interfaceeffect in the Intelligence iServer, but the values set must be retained as part of the BQY document. In the event that the BQY document is saved locally for use with other Intelligence clients, any changes made from the Intelligence iServer should be reflected. When encountered in a script, the methods and properties on this list do not cause an entry in an error log file and execution of the script continues.

- ActiveDocument.ShowCatalog
- ActiveDocument.ShowSectionTitleBar
- ActiveDocument.Sections["Query"].DataModel.Connection.SaveWithoutUsername
- ActiveDocument.Sections["Query"].DataModel.Connection.ShowMetadata
- ActiveDocument.Sections["Query"].DataModel.Limits["Year"].VariableLimit

- ActiveDocument.Sections["Query"].DataModel.MetaDataConnection.SaveWithoutUsernam e
- ActiveDocument.Sections["Query"].DataModel.MetaDataConnection.ShowMetadata
- ActiveDocument.Sections["Query"].DataModel.ShowIconJoins
- ActiveDocument.Sections["Query"].Limits["Quarter"].VariableLimit
- ActiveDocument.Sections["Query"].SaveResults
- ActiveDocument.Sections["DataModel"].DataModel.AutoJoin
- ActiveDocument.Sections["DataModel"].DataModel.Connection.SaveWithoutUsername
- ActiveDocument.Sections["DataModel"].DataModel.Connection.ShowMetadata
- ActiveDocument.Sections["DataModel"].DataModel.Limits["Year"].VariableLimit
- ActiveDocument.Sections["DataModel"].DataModel.MetaDataConnection.SaveWithoutUser name
- ActiveDocument.Sections["DataModel"].DataModel.MetaDataConnection.ShowMetadata
- ActiveDocument.Sections["DataModel"].DataModel.ShowIconJoins
- ActiveDocument.Sections["SectionName"].ShowOutliner
- ActiveDocument.Sections["Dashboard"].Shapes["EmbeddedSection"].ScrollbarsAlwaysShow n
- ActiveDocument.Sections["Dashboard2"].Shapes["EmbeddedSectionResults"].ShowOutliner
- ActiveDocument.Sections["Dashboard2"].Shapes["Embedded"].ShowRowNumbers
- Application.ShowMenuBar
- Application.ShowStatusBar
- Application.StatusText
- Application.Visible
- Application.WindowState
- BqFontEffectOverDouble
- BqFontEffectOverLine
- BqFontEffectStrikeThru
- BqFontEffectSuperScript
- BqFontEffectSubScript
- Toolbars.Count
- Toolbars["Formatting"].Name
- Toolbars["Formatting"].Type
- Toolbars["Formatting"].Visible
- Toolbars["Sections"].Name
- Toolbars["Sections"].Type
- Toolbars["Sections"].Visible
- Toolbars["Navigation"].Name

- Toolbars["Navigation"].Type
- Toolbars["Navigation"].Visible

Part II

Release 8.3.1 Addendum

6 Hyperion Performance Suite Installation Guide Changes

This chapter describes the changes to the *Hyperion Performance Suite Release 8.3 Installation Guide*.

In This Chapter

Apache Tomcat Port Conflict with Oracle 9.2.x 156
Apache Tomcat Precompile Script 156
BEA WebLogic 6.x with WebLogic SP3 or Later Manual Configuration Instructions. 157
BEA WebLogic 7.x Manual Configuration Instructions 157
IBM WebSphere 5.0.1, 5.0.2, and 5.1 Manual Configuration Instructions 157
Oracle Configuration 162
Cessation of Services Before Upgrading 162
Configuring SunONE_iPlanet 6.1 on Solaris 163
UNIX Temporary Directories 168

Apache Tomcat Port Conflict with Oracle 9.2.x

If you are using Apache Tomcat with Hyperion Performance Suite, and your system has an Oracle 9.2.x installation, a service named "OracleOraHome92TNSListerner" tends to use the same port as the default Apache Tomcat port, which is 8080. To resolve this conflict, you need to do one of the following:

- Configure the Oracle service to use a different port
- After Hyperion Performance Suite installation, manually configure Apache Tomcat to use a different port.

To configure the Oracle service to use a different port, consult the documentation supplied by the vendor.

➤ To manually configure Apache Tomcat to use a different port:

1 **Open the** *INSTALL_HOME*`/Tomcat/conf/server.xml` **file.**

2 **Replace** `8080` **(see the following code snippet) to a valid non-conflicting port (for example, 8081):**

```
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on port 8080 -->
    <Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
        port="8080" minProcessors="5" maxProcessors="75"
        enableLookups="true" redirectPort="8443"
        acceptCount="100" debug="0" connectionTimeout="20000"
        useURIValidationHack="false" disableUploadTimeout="true"  />
```

Apache Tomcat Precompile Script

The following sample `precompile.sh` script replaces the one shown on page 160 of the *Hyperion Performance Suite Release 8.3 Installation Guide*.

Sample Apache Tomcat precompile.sh

```
#!/bin/sh
JAVA_HOME=<Java home directory>
TOMCAT_HOME=<Tomcat directory>
SERVLETS_HOME=<Servlets home>
CLASSPATH=./ant-contrib-
0.4.jar:./ant.jar:./optional.jar:"$JAVA_HOME"/lib/tools.jar:"$CLASSPATH"
echo "$JAVA_HOME"/bin/java -classpath "$CLASSPATH"
org.apache.tools.ant.Main -Dtomcat.home="$TOMCAT_HOME" -
Dwebapp.name=Hyperion -Dwebapp.path="$SERVLETS_HOME" -buildfile
tomcat-precompile-jsp.xml
"$JAVA_HOME"/bin/java -classpath "$CLASSPATH"
org.apache.tools.ant.Main -Dtomcat.home="$TOMCAT_HOME"
-Dwebapp.name=Hyperion -Dwebapp.path="$SERVLETS_HOME" -buildfile
tomcat-precompile-jsp.xml
```

BEA WebLogic 6.*x* with WebLogic SP3 or Later Manual Configuration Instructions

Because BEA WebLogic 6.*x* with WebLogic SP 3 or later uses JDK 1.3.*x*, you must consult the "Using JRE 1.3.x" on page 147 of the *Hyperion Performance Suite Release 8.3 Installation Guide* for additional configuration steps.

BEA WebLogic 7.*x* Manual Configuration Instructions

BEA WebLogic 7.*x* should be configured to use Sun JDK 1.4.1; otherwise, you must consult the "Using JRE 1.3.*x*" on page 147 of the *Hyperion Performance Suite Release 8.3 Installation Guide* for additional configuration steps.

IBM WebSphere 5.0.1, 5.0.2, and 5.1 Manual Configuration Instructions

The instructions for manually configuring IBM WebSphere 5.0.1 and 5.0.2 as presented in Chapter 9 of the *Hyperion Performance Suite Release 8.3 Installation Guide* have changed. To configure IBM WebSphere 5.0.1 and 5.0.2 manually, use the instructions in the following sections.

Configuring IBM WebSphere 5.0.1, 5.0.2, or 5.1

To configure and test the servlets on IBM WebSphere 5.0.1, 5.0.2, or 5.1, you must perform the tasks described in the following sections:

1. "Configuring IBM WebSphere 5.0.1, 5.0.2, or 5.1 Manually" on page 158
2. "Copying Required Files" on page 159
3. "Configuring a Local IBM HTTP Server" on page 160
4. "Configuring a Remote IBM HTTP Server" on page 161
5. "Testing the Servlets" on page 162

Considerations

Notice the following items:

- Hyperion does not support WebSphere Application Server Global Security, so you are not prompted for a username or password. You must disable WebSphere Application Server Global Security (if enabled) before installation if you are using the Hyperion Performance Suite installer.
- The Hyperion Performance Suite servlets do not start properly after installation if Java 2 Security is enabled.

- Before installing the Hyperion Performance Suites servlets, verify that the WebSphere Application Server and the IBM HTTP Server are configured and working correctly by testing one of the following URLS:

  `http://server/snoop` or `https://server/snoop`

  If access to one of these URLS was successful, continue with the WebSphere Application Server and the IBM HTTP Server manual configuration steps. If you could not access one of these URLs, refer to the IBM documentation on how to configure WebSphere Application Server and the IBM HTTP Server.

The following steps assume that WebSphere Application Server and the IBM HTTP Server are installed correctly.

Configuring IBM WebSphere 5.0.1, 5.0.2, or 5.1 Manually

➤ To configure IBM WebSphere:

1. **Start your WebSphere Server.**

   For example, on Windows, run `/IBM/AppServer/bin/startServer.bat server1`. On UNIX, run `/usr/WebSphere/AppServer5/bin/startServer.sh server1`.

2. **Enter the User ID, then click OK.**

3. **In the left Navigation frame, select Applications→Install New Application.**

4. **Select Server path, enter the path to the `Brio.ear` file in the `\misc` directory on the Servers CD Disk 3, then click Next.**

5. **Select Generate Default Bindings, then click Next.**

6. **Select the Pre-compile JSP (if you want to precompile JSPs) and Deploy EJBs AppDeployment options, then click Next.**

7. **Select Web Module and select a name from the Virtual Host drop-down list, click Next twice, then click Finish.**

   Wait while WebSphere deploys the application.

8. **Select Save in the Messages area, then select Save in the Save to Master Configuration area.**

9. **Open the `\WebSphere\AppServer\installedApps\SERVERNAME\Brio.ear\Brio.war\config\ws.conf` file and make these changes:**

   - Modify the `WebClient.Diagnostics.Log.Configuration` parameter to include the path to `servletLog4JConfig.xml`, which is `/WebSphere/AppServer/installedApps/SERVERNAME/Brio.ear/Brio.war/config/`.

Caution! Notice that when specifying the path, you must use slashes (/), not backslashes (\).

- Modify the `WebClient.Diagnostics.Log.LoggingService` parameter to GSM_computer_name:GSM_port_number value.
- Modify the `WebClient.UserInterface.Login.RMServer` parameter to GSM_computer_name:GSM_port_number value.

For example:

```
WebClient.Diagnostics.Log.Configuration=file://localhost/C:/Brio/servlet
s/deployment/config/servletLog4JConfig.xml

WebClient.Diagnostics.Log.LoggingService=venice:1800

WebClient.UserInterface.Login.RMServer= venice:1800
```

10 **Open the** `\WebSphere\AppServer\installedApps\`*`SERVERNAME`*`\Brio.ear\Brio.war\config\servletLog4JConfig.xml` **file and set the path for the** `appenders.xml` **and** `debugLoggers.xml` **files to the directory in which these files are located, which is** `/WebSphere/AppServer/installedApps/`***SERVERNAME***`/Brio.ear/Brio.war/config/`.

11 **Open the** `/WebSphere/AppServer/installedApps/`***SERVERNAME***`/Brio.ear/Brio.war/config/appenders.xml` **file and set the path for the log files to the directory in which this file is located.**

Note that when specifying the path, you must use slashes (/), not backslashes (\).

12 **Change the DEPLOYMENT_LIB and DEPLOYMENT_PATH variables in the** `/servlets/bin/servletconfig.bat` **file to point to the deployment directory.**

Copying Required Files

➤ To copy required files:

1 **Copy the** `Brio.Insight.en.cab`, `Brio.InsightNoHelp.en.cab`, `Brio.Insight.en.xpi`, **and** `Brio.InsightNoHelp.en.xpi` **files from the Servers CD Disk 3** `\misc\zeroadmin` **directory to the** `\WebSphere\AppServer\installedApps\`*`SERVERNAME`*`\Brio.ear\Brio.war\zeroadmin\component` **directory.**

2 **Extract the help files from the Servers CD Disk 3** `\misc\help` **directory to the** `\WebSphere\AppServer\installedApps\`*`SERVERNAME`*`\Brio.ear\Brio.war\help` **directory.**

Note: To simplify the process of unzipping help, the installation includes a file in the *`INSTALL_HOME`*`/install` directory called `unzipWebComponentHelp.bat` or `unzipWebComponentHelp.sh`. In this file, fill in the location of the help files (for example, `mountedDrive\serverCD\disk3\misc\help`), the location of a JDK on your system, and the target deployment location (for example, `/home/websphere/installedApps/Brio.ear/Brio.war/`). When you run the script, the help is unzipped.

3 **If you changed the default context root, in the** `ws.conf` **file, replace** `/Hyperion` **in the** `WebClient.Applications.URL.*` **parameters with your custom context path.**

4 **Restart WebSphere.**

Configuring a Local IBM HTTP Server

If you are using the IBM HTTP server on the same computer on which the WebSphere application server is installed, you must verify that the WebSphere plug-in is correctly configured. To do so, enter the following line in the Address bar of your Web browser:

`http//http_server_host[:http_server_port]/snoop URI`

If you receive a 404 error, you must manually configure the local IBM HTTP Server.

If your IBM HTTP server is on a different computer from the WebSphere application, see "Configuring a Remote IBM HTTP Server" on page -161 for instructions.

➤ To configure a local IBM HTTP server manually:

1. **Log in to the WebSphere Administrative Console.**
2. **Verify that the IBM HTTP Server host is served by WebSphere as a virtual host:**
   a. Navigate to **Environment >Virtual Hosts > *hostname (default_host by default)* > Host**.

      Aliases should contain the following entries:

      - **Host Name**— *
      - **Port**—your HTTP Server port

      If the proper host alias is absent, add it:

      i. Click **New**.

      ii. Supply the new alias information with the following values:

      - **Host Name**— *
      - **Port**—your HTTP Server port

      iii. Click **OK**.

      iv. Click the **Save** link, then click the **Save** button.

   b. Go to **Environment** > **Update Web Server Plugin** and click **OK** to generate the `plugin-cfg.xml` file.
3. **Add the path to the `plugin-cfg.xml` file to the *`IBM_HTTP_SERVER_HOME`*`/conf/httpd.conf` file; for example:**

   `WebSpherePluginConfig WAS_HOME/config/cells/plugin-cfg.xml`
4. **Add the path to the WebSphere plug-in (`mod_ibm_app_server_http.dll` on Windows; `mod_ibm_app_server_http.so` on Linux, Solaris, and AIX) to the *`IBM_HTTP_SERVER_HOME`*`/conf/httpd.conf` file. For example, on UNIX:**

```
LoadModule ibm_app_server_http_module WAS_HOME/bin/mod_ibm_app_server_http.so
```

5. **Restart WebSphere (if you added host aliases).**
6. **Restart the IBM HTTP Server.**

Configuring a Remote IBM HTTP Server

If you are using the IBM HTTP server on a different computer from the WebSphere application server, you must manually configure the remote IBM HTTP server after you have installed Hyperion Performance Suite.

➤ To configure a remote IBM HTTP Server manually:

1. **Log on to the WebSphere Administrative Console.**
2. **Go to Environment→Update Web Server Plugin and click OK to generate the** `plugin-cfg.xml` **file.**
3. **Create a directory in** *IBM_HTTP_SERVER_HOME*, **for example:**

   IBM_HTTP_SERVER_HOME/brio

4. **Move the regenerated** `plugin-cfg.xml` **file from** *WAS_HOME*/config/cells/plugin-cfg.xml **to the directory that you created in Step 3.**
5. **Add the path to the moved** `plugin-cfg.xml` **file to the** *IBM_HTTP_SERVER_HOME*/conf/httpd.conf **file; for example:**

   WebSpherePluginConfig *IBM_HTTP_SERVER_HOME*/brio/plugin-cfg.xml

6. **Move the** *WAS_HOME*/bin/mod_ibm_app_server_http.so **(UNIX) or the** *WAS_HOME*/bin/mod_ibm_app_server_http.dll **(Windows) library file to the directory that you created in Step 3.**
7. **Add the path to the moved library file to the** *IBM_HTTP_SERVER_HOME*/conf/httpd.conf **file. For example, on UNIX:**

   LoadModule ibm_app_server_http_module *IBM_HTTP_SERVER_HOME* /brio/mod_ibm_app_server_http.so

8. **Copy the WebSphere** /wsmedia **directory on the remote computer to the local directory that you created in Step 3, for example:**

   IBM_HTTP_SERVER_HOME/brio/wsmedia

9. **Add this line to the** `httpd.conf` **file to create an alias for** /wsmedia**:**

   Alias /wsmedia "*IBM_HTTP_SERVER_HOME*/brio/wsmedia"

10. **Create an alias for the Hyperion application (which is the real path on the remote computer with WebSphere):**

    Alias Brio "*WAS_HOME*/installedApps/*NODE_NAME*/Brio.ear/Brio.war"

11. **Correct the path to the log file in the** *IBM_HTTP_SERVER_HOME*/brio /plugin-cfg.xml **file:**

    <Log LogLevel="Error" Name="*IBM_HTTP_SERVER_HOME*/brio /http_plugin.log"/>

12 **Correct the transport host name in the** *IBM_HTTP_SERVER_HOME*/brio/plugin-cfg.xml **file:**

```
<Transport Hostname="WebSphere_Machine_IP_Address" Port="9080" Protocol="http"/>
<Transport Hostname="WebSphere_Machine_IP_Address" Port="9443" Protocol="https">
```

13 **Copy these files from the remote WebSphere to the** *IBM_HTTP_SERVER_HOME*/brio/etc **directory:**

- plugin-key.kdb
- plugin-key.sth

14 **Correct the path to these files in the** *IBM_HTTP_SERVER_HOME*/brio/plugin-cfg.xml **file:**

```
<Property name="keyring" value="IBM_HTTP_SERVER_HOME/brio/etc/plugin-key.kdb"/>
<Property name="stashfile" value="IBM_HTTP_SERVER_HOME/brio/etc/plugin-key.sth"/>
```

15 **Start the IBM HTTP Server.**

Testing the Servlets

Test that the servlets run in the Web server.

➤ To test the servlets, in a Web browser, go to the following URL:

```
http://localhost:7001/Hyperion/browse/login
```

Oracle Configuration

When creating an Oracle user account for Hyperion Performance Suite, you do not need to specify a temporary tablespace, as stated on page 45 of the *Hyperion Performance Suite Release 8.3 Installation Guide*.

To create the Hyperion Performance Suite user, enter the following commands from within the Oracle SQL*Plus utility:

```
create user username identified by password
default tablespace tablespace;
```

Cessation of Services Before Upgrading

In addition to the general upgrade recommendations listed on page 57 of the *Hyperion Performance Suite Release 8.3 Installation Guide*, you must also stop all services prior to upgrading Hyperion Performance Suite. If you do not stop all services, the upgrade will fail, even though the installer gives you a message saying that the upgrade was successful. This is true for both single and distributed installations.

Configuring SunONE_iPlanet 6.1 on Solaris

Instructions for configuring SunONE iPlanet 6.1 on Solaris are included in the following sections:

- "Deploying the Hyperion Web Applications on SunONE iPlanet 6.1 (Solaris)" on page 163
- "Precompiling SunONE iPlanet 6.1" on page 166
- "Removing the Deployed Hyperion Web Applications" on page 167

Deploying the Hyperion Web Applications on SunONE iPlanet 6.1 (Solaris)

To configure SunONE iPlanet 6.1 Web server with its native servlet engine manually, you must complete the tasks described in the following sections:

- "Unpacking the Web Application Contents" on page 163
- "Updating the ws.conf and Logging Parameters" on page 164
- "Replacing the English Language Files (Optional)" on page 164
- "Setting the JDK 1.4.2 Path" on page 165
- "Deploying the Servlets" on page 165
- "Adding the wsmedia Virtual Directory" on page 166

Unpacking the Web Application Contents

Note: This task is required for the application installed with the "Do not configure servlet engine at this time" option. Skip this step if the application was installed with "Install servlet files in preparation for manual deployment" option. If you are uncertain which option was selected during the installation, check whether *HYPERION_HOME*`/servlets/deployment` directory exists. If it does, you do not need to perform this task.

1. **Unjar the `Brio.ear` file.**

   Copy `Brio.ear` (this file is located in `disc3/misc` folder) to a temporary directory and unjar it with using *JAVA_HOME*`/bin/jar -xvf Brio.ear`.

   This extracts the `Brio.war` file and `META-INF` directory from `Brio.ear`.

2. **Unjar the `Brio.war` file.**

   Move Brio.war from the temporary directory to `HYPERION_HOME/servlets/deployment` directory and unjar it using *JAVA_HOME*`/bin/jar -xvf Brio.war`.

   This extracts `Brio.war` files and subdirectories.

3. **Remove `Brio.war` file from the `HYPERION_HOME/servlets/deployment` directory.**

Updating the ws.conf and Logging Parameters

➤ To update the ws.conf and logging parameters:

1 **In the** *WEB_APPLICATION_DIRECTORY*`\config\ws.conf` **file:**

   a. Locate WebClient.Internal.Redirect.RedirectPolicy=HttpRedirect and change the value to JavaScriptRedirect.

   b. Modify the WebClient.Diagnostics.Log.LoggingService and WebClient.UserInterface.Login.RMServer parameters for next values (GSM_computer_name:GSM_port_number).

   c. Set the servletLog4JConfig.xml path to the directory in which the file is located.

2 **Open the** *WEB_APPLICATION_DIRECTORY*`\config\servletLog4JConfig.xml` **file and set the path to the appenders.xml and debugLoggers.xml files to the directory in which these files are located.**

3 **Open the** *WEB_APPLICATION_DIRECTORY*`\config\appenders.xml` **file and set the path for the log files to the directory in which this file is located.**

4 **Change the DEPLOYMENT_LIB and DEPLOYMENT_PATH variables in the** `/servlets/bin/servletconfig.bat` **file to point to the deployment directory.**

Replacing the English Language Files (Optional)

➤ To replace the English language files:

1 **Copy the template language files.**

   For example, for French, copy;

   INSTALL_HOME`\servlets\deployment\_templates\cctemplates_`

   to:

   `fr_FR to` *INSTALL_HOME*`\servlets\deployment\cctemplates.`

2 **Copy the Zero administration language files.**

   For example, for French, copy these files from the Hyperion Performance Suite Servers CD Disk 3 `\misc\zeroadmin` directory to the *INSTALL_HOME*`\servlets\deployment\zeroadmin\component` directory:

```
Brio.Insight.fr.cab
Brio.Insight.fr.xpi
Brio.InsightNoHelp.fr.cab
Brio.InsightNoHelp.fr.xpi
```

3 **Copy the** `licenses.txt` **file from** *INSTALL_HOME*`\lib` **to** *INSTALL_HOME*`\servlets\deployment\WEB-INF\classes`.

Setting the JDK 1.4.2 Path

➤ To set the path to JDK 1.4.2:

1. In the **Administration Console**, select a virtual server to manage from the **Manage Virtual Servers** drop-down list, then click **Manage**.
2. Select the **Java** tab.
3. Select **JVM General**.
4. Add the path to JDK 1.4.2 to the Java Home path.
5. Click **OK**.
6. Click **Apply** (at the top right).

Deploying the Servlets

➤ To deploy the servlets:

1. In the **Administration Console**, click the **Servers** tab.
2. Select **Manage Virtual Servers**.
3. Select a virtual server to manage from the **Manage Virtual Servers** drop-down list, then click **Manage**.
4. Click **Class Manager Apply** (at the top right).
5. Select the server name entry under **Tree View of the Server** to manage that server.
6. Select the **Web Applications** tab.
7. On the **Web Applications** page, click **Deploy Web Applications**.
8. Enter the following information:
   - **WAR File On**—Server/Local Server
   - **WAR File Path**—Location of the WAR file (`Brio.war`) This should be the root directory of the Web application.
   - **Application URL**—`/Hyperion` (the context where your deployment is located)
   - **Installation Directory**—Location where you want to place the expanded deployment directory of the Web application. Should be the same as the WAR file path.
9. Click **OK** to deploy the application.

Adding the wsmedia Virtual Directory

➤ To add the `wsmedia` virtual directory:

1. In the **Administration Console**, click the **Servers** tab.
2. Select **Manage Virtual Servers**.
3. From the **Manage Virtual Servers** drop-down list, select a virtual server to manage, then click **Manage**.
4. Click **Class Manager** (at the top right).
5. In the **Administration Console**, click the **Content Management** tab, then click **Additional Document Directories**.
6. Specify these parameters, then click **OK**:
   - URL Prefix—`/wsmedia`
   - Map To Directory—Path to the `/wsmedia` directory
7. Click **Apply** (at the top right).

Precompiling SunONE iPlanet 6.1

Precompiling greatly improves the performance of the Hyperion Performance Suite servlets. Use the following sample script to precompile Sun Java System Web Server.

iplanet61_precompile_solaris.sh Sample Script

```
#!/bin/sh
# precompile servlet for iPlanet/Solaris
IWS_SERVER_HOME=/export/home/brcore/iPlanet61
BRIOHOME=/export/home/brcore/hps83
URIBASE=$BRIOHOME/servlets/deployment/WEB-INF/jsp
SERVER_NAME=https-epmwu10.qa.minsk.epam.com
APP_NAME=Hyperion
APP_ROOT=$BRIOHOME/servlets/deployment
OUTPUT_DIR=$IWS_SERVER_HOME/$SERVER_NAME/ClassCache/$SERVER_NAME/$APP_NA
ME
$IWS_SERVER_HOME/bin/https/bin/jspc -genclass -v2 -uribase $URIBASE -d
$OUTPUT_DIR -webapp $APP_ROOT
```

Precompilation is an optional step. It has a significant effect on performance, but you do not need to precompile for a development or test system.

Removing the Deployed Hyperion Web Applications

Follow the tasks outlined in the following sections to remove the Hyperion Web Applications:

- "Removing the Deployment" on page 167
- "Removing the wsmedia Virtual Directory" on page 167
- "Resetting the JDK to the Default" on page 168
- "Optional: Removing the Extracted Hyperion Deployment Files" on page 168

Removing the Deployment

➤ To remove the deployed Hyperion Web applications:

1. In the **Administration Console**, click the **Servers** tab.
2. Select **Manage Virtual Servers**.
3. Select a virtual server to manage from the **Manage Virtual Servers** drop-down list, then click **Manage**.
4. Click **Class Manager** (at the top right).
5. Select the server name entry under **Tree View of the Server** to manage that server.
6. Select the **Web Applications** tab.
7. On the **Web Applications** page, click **Edit Web Applications**.
8. Select **Delete** from the **Action** drop down list on the Hyperion URI.
9. Click **OK**.
10. Click **OK** on the confirmation box to remove the deployment.
11. Click **OK** on the **Web Applications Saved** box.
12. Click **Apply** (at the top right).
13. Click **Apply Changes**.
14. Click **OK** at the success message.

Removing the wsmedia Virtual Directory

➤ To remove the `wsmedia` virtual directory:

1. In the **Administration Console**, click the **Servers** tab.
2. Select **Manage Virtual Servers**.
3. Select a virtual server to manage from the **Manage Virtual Servers** drop-down list, then click **Manage**.
4. Click **Class Manager** (at the top right).
5. Click the **Content Management** tab, then click **Additional Document Directories**.
6. Click **Remove** next to the wsmedia entry, then click **OK** to accept the remove and **OK** on the success.

Resetting the JDK to the Default

➤ If you changed the default JDK, follow these steps to reset the JDK to the default:

1. In the **Administration Console**, select a virtual server to manage from the **Manage Virtual Servers** drop-down list, then click **Manage**.
2. Select the **Java** Tab.
3. Select **JVM General**.
4. Place in the Java home the path to the Sun JDK 1.4.2 (`$SUNONEDIR/bin/https/jdk`, replace `$SUNONEDIR` with the actual directory tree of the install)
5. Click **OK**.
6. Click **Apply** (at the top right).

Optional: Removing the Extracted Hyperion Deployment Files

➤ Remove the directory and contents of *`HYPERION_HOME`*`/servlets/deployment`.

After you have removed the deployed Hyperion Web application, you can uninstall Hyperion Performance Suite.

UNIX Temporary Directories

The information on UNIX temporary directories included on pages 51, 64, 78, and 223 of the *Hyperion Performance Suite Release 8.3 Installation Guide* is updated as follows:

> On UNIX systems, to avoid conflicts with `/tmp` directories created by other users, or access problems to the `/tmp` directory, use this syntax to specify the temporary directory that you use when running the installer:
>
> ```
> setupSolarisPlatform.bin -is:tempdir /home/user/hpsu_install_tmp
> ```
>
> where `hpsu_install_tmp` is a pre-created directory with at least 1 GB of space available.
>
> Calling this private temporary directory something other than `tmp` avoids confusion with the system `/tmp` directory.
>
> Depending on the Hyperion Performance Suite components that you install, some support files may be left in this directory after the installation is completed. You should also make sure that this temporary directory does not contain any important files that are not Hyperion Performance Suite files, because running the Hyperion Performance Suite uninstaller may purge the entire directory.
>
> InstallShield also uses the system `/tmp` directory for some temporary files, so make sure that `/tmp` is not full.

7 Hyperion Performance Suite Enhancements

This chapter describes the general changes and enhancements made to Hyperion Performance Suite Release 8.3.1.

In This Chapter

Accessibility Changes 170
Backward Compatibility 170
Default Settings for Services 172
Default Settings for Servlets. 173
Tips on Distribution and Replication of Services 174
Supported Web Servers 174

Accessibility Changes

When using MAGic screen magnification software for Microsoft Windows with Hyperion Performance Suite, there is no indicator of which frame is selected within a Pivot, Results, or Chart section of a BQY document in the Hyperion Intelligence iServer. This issue has been addressed as follows:

- To display the speed menu content and focus on the first item in the speed menu frame, press **[Shift]+[F10]**.

  After you use the [Shift]+[F10] keystroke combination to display the speed menu frame (from any Results, Pivot, or Chart section), you will be placed at the top of the speed menu frame's list of selections.

- When tabbing through the toolbar buttons, when the last toolbar button is reached (which is the help button) the next [Tab] will select the frame image. Pressing [Tab} again will select the first label of the section (Pivot, Results, or Chart).

Backward Compatibility

This section describes the backward compatibility between Hyperion Intelligence products. Compatibility issues are described in the following topics:

- "Compatibility Between Different Client and Server Releases" on page 170

Note: Compatibility between different versions of server components, including servlets, is never supported. For example, a Release 8.2.1 Intelligence Service cannot process queries using a Release 8.3 Data Access Service.

- "Compatibility Within BQY Documents Across Versions" on page 171
- "Storage of Content Created by One Software Release in a Repository Managed by a Different Release" on page 172

Compatibility Between Different Client and Server Releases

Within a major release series, an earlier Hyperion Intelligence Web Client (formerly Insight) can connect to and interact with a later release of the server.

Note: The major release number is the leftmost digit in the version number. Releases 8.1.2, 8.2.0, and 8.3.0 all belong to the major release series 8.

For example, a Release 8.1.2 Hyperion Intelligence Web Client can open BQY documents stored in a Release 8.3 server and process queries through the Release 8.3 server.

Note: For Release 8.3, because of security enhancements in the product, a configuration change must be made to enable backward compatibility. Hyperion does not recommend that you make a configuration change, because the change disables the security features of Release 8.3. The documentation explains how to make the change.

Because this capability does not work, it is considered a product defect and Hyperion will support the customer in these cases.

Later clients will not work with earlier servers, and Hyperion does not support such configurations. For example, a Release 8.3 Hyperion Intelligence Web Client cannot open content stored on a Release 8.2.1 server.

Compatibility Within BQY Documents Across Versions

Note: This discussion applies to all software components used to open BQY documents: Hyperion Intelligence Designer, Hyperion Intelligence Explorer, the Hyperion Intelligence Web Client, and the Hyperion Intelligence Service (BIService). BIService is used to support both Intelligence iServer (iHTML or thin client) use of BQY documents and to process BQY jobs.

Hyperion Solutions makes every effort to ensure that BQY files created by one release of the software are usable in other releases. In most cases, earlier software releases, while opening the document, simply discard content created by a later release of the software that the earlier releases do not understand. In other cases, errors may occur when a document is opened, but it is still generally usable, such as when new functions are introduced and used in a document that is then opened in an earlier release, before that function existed.

When new features are added to new releases or critical fixes are applied to a patch of an existing release, it may not be possible to maintain the usability of documents created or modified by these later software releases or patches. For example, new functions for computed items were added in Release 8.2. If a document created in Release 8.2 using these functions is opened by Release 8.1.2, an error will occur on those computed columns because Release 8.1.2 does not recognize them. The document could still be opened, however, and would otherwise be functional.

When customers require compatibility between users of different releases of the products, those customers should ensure that only the earliest release in use creates and modifies the content to be accessed by all users. All later releases can access BQY files created and modified by this earlier release. As stated previously, however, Hyperion cannot guarantee that the reverse is the case, that is, that an earlier release can work with a later release.

Hyperion reserves the right to obsolete features. This may mean that some earlier documents will no longer open in later releases. When this happens, Hyperion will announce it and provide a means to convert documents using the obsolete features. For example, Release 8 software cannot open documents using the Detail section of documents created by Release 5 software. Release 6, however, provided the means to convert the Detail sections to the later Report section.

Storage of Content Created by One Software Release in a Repository Managed by a Different Release

When publishing BQY documents, the rules discussed previously are in effect, because the Intelligence Service opens documents while publishing. Therefore:, these are the existing conditions:

- A BQY file created by a later release of Hyperion Intelligence Designer could possibly be published in an earlier release of Hyperion Performance Suite. The Intelligence Service should be able to open the document and ignore content that it does not recognize. Mixed software releases accessing the same BQY file, however, is not guaranteed to work.

Note: Use of Hyperion Intelligence Designer here and elsewhere in this Addendum is only as an example. Any Hyperion Intelligence Client has the means to create content, and the same rules apply as in the examples.

- Publishing a BQY file created by an earlier release of Hyperion Intelligence Designer in a later release of Hyperion Performance Suite is a fully supported operation.
- Upgrading an earlier release of Hyperion Performance Suite to a later release does not cause any modifications or conversions to the Intelligence documents already stored in the repository. Such a conversion is not necessary; newer clients, as well as older clients, will be able to access the documents within the same release number, to the extent that they can handle the content.

  For example, if you upgrade an 8.2.0 server to 8.3.0, and BQY documents create by 8.2.0 are in the repository, then both 8.2.0 and 8.3.0 clients are able to open and use that content.

  An 8.1.2 client would also be able to connect to the server to access the document, given that earlier clients can communicate with later servers. Whether an 8.1.2 client can successfully use the 8.2.0-created document, however, depends on the features of 8.2.0 that were used when the document was created.

Default Settings for Services

You can customize the names and port numbers used by the Hyperion Performance Suite services. Table 1 list the default port numbers and service names for the Hyperion Performance Suite services. The default service names use the format *service_host*, where *host* is the name of the computer on which the service is installed.

Table 1 Default Service Names and Port Numbers

Service	Default Name	Default Port Number
Authentication Service	AN1_host	Not applicable
Authorization Service	AZ1_host	Not applicable
Common Services	CORE1_host	1801
Data Access Service	DAS1_host	1803*

Table 1 Default Service Names and Port Numbers *(Continued)*

Service	Default Name	Default Port Number
Event Service	ES1_host	1494
Global Service Manager (GSM)	Host name	1800
Intelligence Service (also known as BrioQuery Service)	BI1_host	1802*
Job Service	JF1_host	1496
Logging Service	LS1_host	Not applicable
Name Service	NS1_host	1498
Publisher Service	PUB1_host	Not applicable
Repository Service	RM1_host	1495
Service Broker	SB1_host	1497
Session Manager	SM1_host	Not applicable
Usage Service	UT1_host	Not applicable

* For the Intelligence Service and the Data Access Services, the default ports are firewall ports. For second and subsequent HYPERION_HOME installations on a computer, additional ports are opened for these two services.

Default Settings for Servlets

You can customize the application and Web component names. By customizing the names of the applications and the Hyperion Performance Suite servlets, you customize the URLs used to access your system data. Each servlet uses a URL of the following form:

`http://webserver:port/applicationname/componentname`

The default application name is *Hyperion*.

Table 2 Default Servlet Names

Servlet or Web Component	Default Web Service Name
Administration module	administration
Browse module	browse
Dashboard	viewmanager
Data Access	dataaccess
Hyperion Intelligence iServer	ihtml
Job Manager	jobmanager
Personal Pages module	personalpages

Tips on Distribution and Replication of Services

Normally, all common services (RSC and LSC services) processes share the same process space or Java Virtual Machine (JVM). Some of the common services can be distributed. When distributing them, they each execute in their own process space. Some of these services cannot run in their own process space:

- The Name Service has to be in the same process space as the Global Service Manager (GSM)
- The Repository Service must be in the same process space as the Publisher Service.

The only RSC services that can be distributed are the Event Service, Service Broker, and Job Service.

To distribute services, you must reinstall the replicable component using the installer. The installer sets up a Local Service Manager in association with the new HYPERION_HOME on the server, which enables the distributed services to work with the Global Service Manager. Furthermore, the installer updates the database, XML files, and configuration files. The installation process configures services on the different computers.

When starting common services, the set of services actually started is determined by two files: `server.xml` and `config.dat`. Each of the LSC services have an entry defined in the `server.xml` file and have a `<runtype>` line indicating hold or start status. Anything marked *start* will be started. For the RSC services, there is a file in `/BrioPlatform/etc` typically named `BrioPlatform1_host_gsmhost_gsmport.dat`. This file contains one line for each service to run; if you remove a service line in this file, the service will not get started.

Supported Web Servers

Hyperion Performance Suite Release 8.3.1 introduces support for IBM WebSphere 5.1 (on all platforms) and SunONE iPlanet 6.1 (on Solaris). To configure these Web servers manually, follow the instructions in the following sections:

- "IBM WebSphere 5.0.1, 5.0.2, and 5.1 Manual Configuration Instructions" on page 157
- "Configuring IBM WebSphere 5.0.1, 5.0.2, or 5.1" on page 157

8 Hyperion Foundation Enhancements

This chapter lists the changes and enhancements made to Hyperion Foundation for Release 8.3.1.

In This Chapter

New Look 176
Browse Module Enhancements 176
External Authentication Nested Group Support 179
Viewing or Modifying Group Memberships 179
Publish Stand-alone BQY Document Job Action 180
Replace BQY Document or Job 180

New Look

Hyperion Performance Suite has been updated with new colors and a new look. Figure 1 shows the new login page.

Figure 1 Login Page

Browse Module Enhancements

The following sections describe the Browse module enhancements:

- "Browse Preference to Hide Job Output Listings" on page 177
- "SQR Jobs Output Format" on page 177
- "SQR and Generic Job Output Folder" on page 178
- "Automatically Deleting Job Outputs for Schedules" on page 178
- "New Mime Type for BQY Documents Generated by Generic Jobs" on page 178

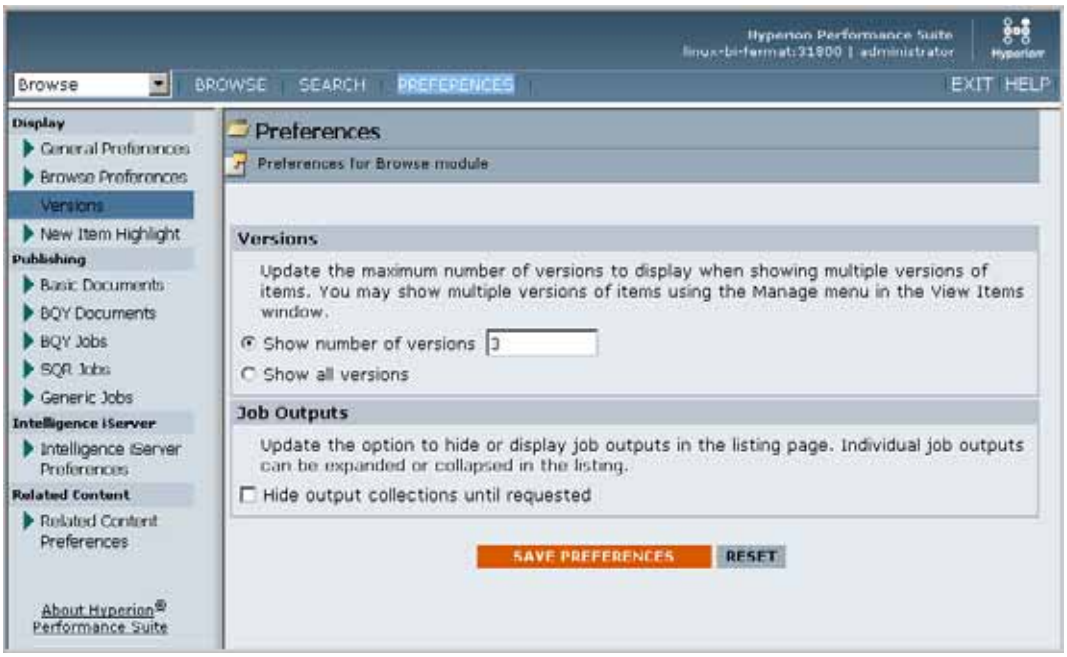
Browse Preference to Hide Job Output Listings

Job output listings can be hidden on the Browse page by enabling the following preference under Versions Preferences:

- **Hide output collections until requested**—Enable to hide job output listings on the Browse page. Under the job title a plus (+) sign displays when job output listings are hidden. Select the plus sign to see hidden job output listings.

Figure 2 shows the new Job Outputs preference on the Browse Preferences page.

Figure 2 Versions Preferences



For more information see the "Browse Preferences" section of the *Hyperion Foundation User's Guide*.

SQR Jobs Output Format

HTML format is automatically selected for an output format unless you deselect it. If you deselect HTML and do not select any other output formats, a warning message is displayed to inform you that you are running a job with no output format selected.

To display SQR job output on a Personal Page, the HTML option must be selected. If the HTML option is not selected, nothing will be displayed on the Personal Page. Only HTML job output can be displayed on a Personal Page.

Secure SQR jobs will continue to display only HTML output.

For more information, see the "SQR and Generic Jobs" chapter of the *Hyperion Foundation User's Guide*.

SQR and Generic Job Output Folder

Job output from SQR and generic jobs are by default stored in the same folder as the job. For jobs with job parameters, you can specify a different folder for job output. For foreground jobs, you can either enter or browse for a job output folder immediately before running the job. For scheduled jobs, you can enter or browse for a folder when modifying job parameters through the Job Manager module.

➤ To specify a job output folder for a foreground SQR or generic job:

1. **Choose the run job icon for the job.**
2. **Enter or Browse for the correct folder.**
3. **Run the job.**

➤ To specify a job output folder for an SQR or generic scheduled job:

1. **In the Job Manager module, select the job from the Scheduled Job list.**
2. **Select modify for the job parameters in which you want to change the job output folder.**
3. **Enter or Browse for the correct folder.**
4. **Choose OK to save the changes.**

Automatically Deleting Job Outputs for Schedules

You can automatically delete job outputs associated with a schedule by enabling the Job Output property when adding or modifying a schedule:.

You indicate when to delete by entering a number and selecting a time frame (minutes, hours, days, weeks, years) after the job is run. Default is days.

For more information on automatic deletion of jobs, see the "Managing Jobs" chapter of the *Hyperion Foundation User's Guide.*

New Mime Type for BQY Documents Generated by Generic Jobs

BQY documents generated by generic jobs will no longer be automatically associated with Intelligence Clients products. No check for adaptive state will be made.

Use an Intelligence Client to open all BQY documents generated by generic jobs. Then work on the BQY documents as before.

External Authentication Nested Group Support

When using external authentication, nested group information is imported for ADS and LDAP drivers. Nested group information is not imported for NT, NIS, or JDBC drivers.

For more information on external authentication, see the *Hyperion Foundation Administrator's Guide*.

Note: If you do not want nested group information, set the `disableImportSubgroupinfo` Java system property in the `startCommonServices` script to TRUE, to maintain the behavior of the old driver.

Viewing or Modifying Group Memberships

In the Administration module, you can view members of groups and modify members of groups to which you have ASSIGN or FULL CONTROL access, and groups that are not managed by an external authentication system. Members of groups can be either users or groups.

If an external authentication system is set to not import external groups, then you can create native groups for this authentication system and you can modify the membership of these groups.

In the Members of Group list, groups are preceded by two asterisks (**).

➤ To view or modify members (users or groups) of a group:

1. Display the member association page by choosing **Manage Groups/Modify/Members**.
2. Select **Groups/Users** and choose **Update** List.
3. View members of this group in the **Members of Group** list.
4. To add members, select **Groups/Users** to add from the **Available Users/Groups** list and choose the greater-than symbols (>>).
5. To delete members, select **Groups/Users** from the **Members of Group** list and choose the less-than symbols (<<).
6. Choose **Apply**.

For more information on managing groups, see the *Hyperion Foundation Administrator's Guide*.

Publish Stand-alone BQY Document Job Action

The publish stand-alone BQY job action creates a stand-alone BQY document as the job output. This job-generated BQY document can be locally saved, changed, and processed like any other BQY document. It inherits all its properties from the job that created it, but it is not attached to the job after it is published. You will be prompted for a destination folder for the stand-alone BQY document.

Note: You cannot add this job action to a cycle in which the action "Save as Job Output" has already been selected.

➤ To create job output as a stand-alone BQY document:

1. **In the Action wizard, select Publish Document from the Action drop-down list.**
2. **Enter the information requested:**
   - **Document Name**—The name can contain letters, numbers, the space character, and the underscore character (-). The default name is the same as the job name.
   - **Description**—A meaningful description of the stand-alone BQY document. The default is the description of the job.
   - **Destination Folder**—Browse for a folder. The default folder is the folder of the job.
3. **Run the job.**

For additional information on BQY job actions, see the *Hyperion Foundation User's Guide.*

Replace BQY Document or Job

The following sections describe the Replace BQY enhancements for both end users and administrators.

- "End-User Changes for Replace BQY" on page 180
- "Administrator Changes for Replace BQY" on page 181

End-User Changes for Replace BQY

You can publish modified BQY documents or jobs as versions of existing BQY documents or jobs. You can choose to replace or add a later version of a modified BQY document, and you can replace a BQY job. E-mail notifications are automatically sent to owners of job schedules (if they set their e-mail addresses in preferences) that are affected by replaced jobs.

You must set open catalog extensions (OCEs) for new sections and reset the sections that you want to pregenerate when you replace BQY documents or jobs. The user interface also gives you the option of creating a message of up to 4,000 characters to send in the e-mail notification.

➤ To replace a BQY document or job:

1 **Choose Replace on the version page for BQY documents or on the general properties page for BQY jobs.**

2 **Select a file.**

3 **Fill in the properties in the Replace wizard.**

For detailed property information, see "Query Properties," "BQY Job Properties," and "Job Defaults" in the *Hyperion Foundation User's Guide*.

4 **Choose Return to Job.**

5 **Job schedule owners may need to modify their job parameters to reflect the changes in the job.**

Each new or replaced BQY document version has its own properties.

➤ To modify version properties:

1 **In Browse, navigate to the BQY document whose version you want to edit.**

2 **Choose Modify.**

3 **Modify the properties.**

For detailed property information, see "Query Properties" and "Properties of Versions" in the *Hyperion Foundation User's Guide*.

4 **Save the changes.**

Administrator Changes for Replace BQY

An error message is logged and the job continues to run when the Intelligence Service encounters job actions that no longer apply.

Replacing BQY jobs can cause Smartcuts associated with the job to fail if job parameters are changed. You must update the Smartcuts.

9 Hyperion Intelligence Enhancements

This chapter lists the changes and enhancements made to Hyperion Intelligence for Release 8.3.1.

In This Chapter

New Icons 184
Null Values in Pivot Section 184
Object Model Enhancements 185

New Icons

Hyperion Intelligence Release 8.3.1 includes new launch and document icons (see Table 3) to update the look and feel of the interface and provide a user experience more consistent with other Hyperion Products. The new icons are available in all Hyperion Intelligence Client products including these:

- Hyperion Intelligence Designer
- Hyperion Intelligence Explorer

The new icons replace the earlier icons on all Intelligence documents, dialog boxes, grids, lists, and trees.

Table 3 Hyperion Intelligence Icons

Icon Type	Old Icon	New Icon
Document		
Launch		

Null Values in Pivot Section

The default option to evaluate and show null values for moving functions in a new Pivot section is now disabled. Prior to Release 8.3.1, the default option was enabled. If you want to evaluate and show null values for a new Pivot section, then you must select Enable Null Facts In Computed Items on the General tab of the Pivot Options dialog box.

Object Model Enhancements

Object Model enhancements have been added in Release 8.3.1 as described in the following sections:

- "Width (Property)" on page 185
- "Session (Object)" on page 186

Width (Property)

Width (Property) has been extended to TableFacts and TableDimensions objects in the Report section. Before this release, Width (Property) was only available for Line objects.

Now in the Report section, Width (Property) enables you to set the width of a table fact or dimension column.

Width (Property) represents a pixel value, which is readable and writable. The minimum value is 7 screen pixels and the maximum value is 7560 screen pixels.

When the width is set to a new value, the column width of a table object is resized accordingly. If the width value is positive but exceeds the width of a window or content frame, a catchable error is thrown.

If the width is set to a negative number, then the following catchable error is thrown:

```
Invalid width specified in column's name's width.
```

The following example shows how to set the width of a table fact column to three hundred pixels:

```
ActiveDocument.Sections["Report"].Body.Tables["Table"].Facts["Amount"].Width = 300
```

Session (Object)

The Item methods of the variable collections for Session (Object) have been enhanced to accept index values. The Forms, Cookies, and URL collections now accept a (string) value name or an index value for key value pairs.

If a string value is passed, the Item method accepts a string parameter (the key) and a string output value (the value).

If an integer is passed, then it is assumed to be an index, and the *nth* parameter key and value pair is returned. For example, the following value pair may be returned:

```
bqtype=Insight
```

where the key and value are separated by an equal sign (=).

If an invalid index value is passed, the method throws a catchable error and no value is returned.

The Forms, Cookies, and URL collections now can use Count (Property). This property returns a count of the total number of parameter key and value pairs in the URL, cookie, or form.

Additionally, Session (Object) has been removed from the exclusion list of the Hyperion Intelligence iServer Object Model.

10 Hyperion SQR Enhancements

This chapter discusses the features and enhancements offered by Hyperion SQR Release 8.3.1.

In This Chapter

SAP BW Data Source Enhancements 188
Support for BINARY Data 193
Specification of a Separator Value for CSV File Generation 195
Deprecated Thin-Client Analysis Options in SQR Developer 196

SAP BW Data Source Enhancements

Hyperion SQR Release 8.3 includes the following SAP BW data source enhancements:

- "New Syntax to Sort by Dimension Properties or Selected Measures" on page 188
- "New Ranking Operators for Filtering" on page 190
- "Hierarchies Under Characteristics" on page 191
- "Restricting Data Retrieved" on page 191

New Syntax to Sort by Dimension Properties or Selected Measures

In Hyperion SQR Release 8.3.1, additional syntax for the `ORDER BY` clause enables you to sort a dimension by a dimension property or by a selected measure. This sorting process specifies the order in which selected dimension items are returned.

Syntax

ORDER BY [(DIM, EXPRESSION)] {SO} [(DIM, EXPRESSION) {SO}]…

Arguments

`DIM`
A string variable, column, or literal that represents a legal dimension or hierarchy.

`EXPRESSION` is of this form:

```
(MEASURE[.VALUE])
```

or

```
(Dimension[.Dimension Property])
```

`SO`
A literal value of one of the following:

- **ASC**—Sort in ascending order and preserve the hierarchy.
- **BASC**—Sort in ascending order and break the hierarchy.
- **DESC**—Sort in descending order and preserve the hierarchy.
- **BDESC**—Sort in descending order and break the hierarchy.

SO cannot be a bind variable.

Example The following sorts the selected measure by a specific dimension property (`0APO_RTYPE, 0APO_RTYPE.20APO_RTYPE`) in ascending order, breaking the hierarchy (`BASC`).

```
Begin-Select
  0APO_RTYPE
  Measures.D5XK1R2CAGKTXDWNCTIDVK8RI
  From SAP BW
  ORDER BY (0APO_RTYPE, 0APO_RTYPE.20APO_RTYPE) BASC
end-select
```

Example The following code sorts the selected measure by two dimension properties (`ZN_STATE, ZN_STATE.1ZN_STATE`) and (`ZN_SKU, ZN_SKU.1ZN_SKU`) in ascending order, breaking the hierarchy (`BASC`).

```
begin-select
  ZN_STATE
  ZN_SKU
  Measures.ZN_ADDTNS
  Measures.ZN_CLINV
  Measures.ZN_COGS
  Measures.ZN_MISC
  Measures.ZN_MRKTNG
  Measures.ZN_OPINV
  Measures.ZN_PROLL
  Measures.ZN_SALES
  From SAP BW
  ORDER BY (ZN_STATE, ZN_STATE .1ZN_STATE) BASC
           (ZN_SKU, ZN_SKU.1ZN_SKU) BASC
end-select
```

Example The following code sorts the selected measure by a specific dimension property (`ZN_STATE, Measures.ZN_ADDTNS`) in ascending order without breaking the hierarchy (`ASC`).

```
begin-select
  ZN_STATE
  Measures.ZN_ADDTNS
  Measures.ZN_CLINV
  Measures.ZN_COGS
  Measures.ZN_MISC
  Measures.ZN_MRKTNG
  Measures.ZN_OPINV
  Measures.ZN_PROLL
  Measures.ZN_SALES
  From SAP BW
  ORDER BY (ZN_STATE, Measures.ZN_ADDTNS) ASC
end-select
```

New Ranking Operators for Filtering

In Hyperion SQR Release 8.3.1, additional syntax for the `FILTER` clause enables you to use ranking functions when filtering information. You can use a ranking function as an operator value in the `Boolean Expression` argument. (See the bold items in the following `Boolean Expression` argument.)

Syntax

`FILTER (DIM, {`*`Boolean Expression`*`} [{`*`Relation`*`} {`*`Boolean Expression`*`}]...)`

Arguments

`DIM`

A string variable, column, or literal that represents a legal dimension.

`Boolean Expression`

The Boolean Expression is of this form:

```
(MEASURE[.VALUE] {Operator} VALUE [SLICED BY DIM[.VALUE]
[DIM[.VALUE]...])
```

or

```
(Dimension[.Dimension Property] {Operator} "VALUE" [SLICED BY DIM[.VALUE]
[DIM[.VALUE]...])
```

Where operator is one of the following symbols:

```
< > <= >= = !=
```

BottomCount
BottomSum
BottomPercent
TopCount
TopSum
TopPercent

`Relation`

The keyword AND or OR.

Hierarchies Under Characteristics

In Hyperion SQR Release 8.3.1, hierarchies are shown under characteristics for an SAP data source. Figure 3 shows the hierarchical structure of the objects.

Figure 3 Hierarchical Structure of Objects for an SAP BW Data Source

Information Providers
-----Cubes
---------**Characteristics**
----------CharacteristicLevels
----------CharacteristicLevelMembers
----------CharacteristicChildMembers
----------CharacteristicMembers
----------CharacteristicChildMembers
----------Property (Optional)
----------PropertyMembers
----------**Hierarchy**
----------HierarchyLevels
----------HierarchyLevelMembers
----------HierarchyChildMembers
----------HierarchyMembers
----------HierarchyChildMembers
----------Key Figures
----------Variable

Figure 4 shows the hierarchies for a single characteristic.

Figure 4 Hierarchies for a Single Characteristic

Hierarchies for a single characteristic

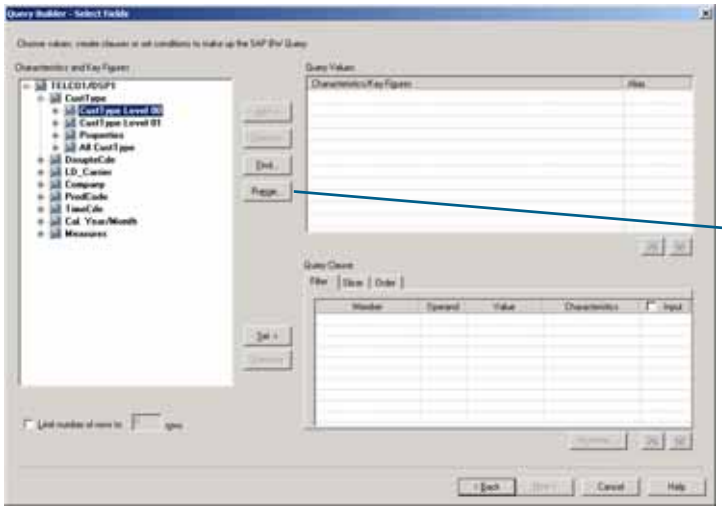
Restricting Data Retrieved

Hyperion SQR Developer Release 8.3 enables you to restrict the data retrieved based on the number of child nodes, level members, hierarchy levels, hierarchy members, or hierarchy child members required for a particular level. This capability provides greater flexibility and enhances performance for levels containing thousands of level members (common for InfoCubes).

➤ To restrict the data retrieved:

1. **Log on to an SAP BW data source and select an object on the first page of the Query Builder.**
2. **On the Select Fields Query Builder page, select a level member, hierarchy level, hierarchy member, or hierarchy child member, and click the Range button.**

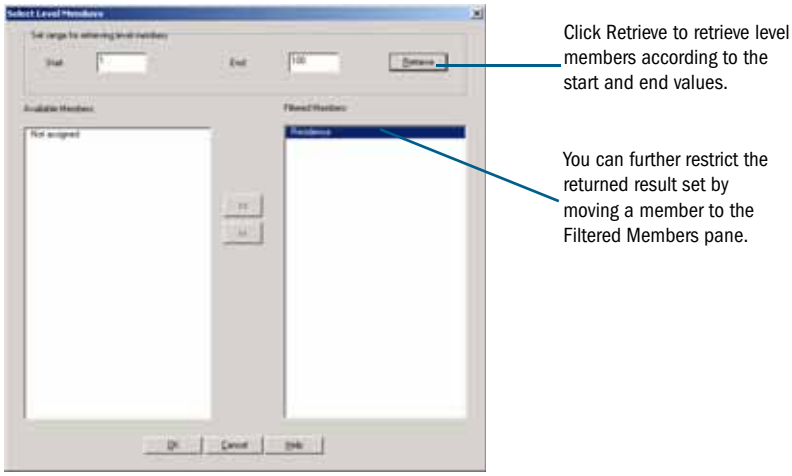
If you select a dimension, hierarchy, property, or property member, the Range button is not selectable.



3 **In the Select Level Members dialog box, enter the start and end values for the range and click Retrieve.**

For example, to restrict the data to the first 100 level members, you would enter 1 for the start value and 100 for the end value. (If you leave the start and end values at their default value of 0, SQR Developer retrieves all the records.)

When you click **Retrieve**, the level members in the defined range are displayed in the Available Members pane.



4 **If you want to restrict further the returned result set for a member in the Available Members box, select the member and click the right arrow to move the member to the Filtered Members box.**

5 **Click OK to return to the Select Fields Query Builder page.**

If you moved a member to the Filtered members box in Step 4, the filtered data is shown in the Filter tab.

Support for BINARY Data

Hyperion SQR Release 8.3.1 enables you to retrieve BINARY column data. Review the following sections for information on these topics:

- "Specifying Whether a Variable or Column Supports BINARY Data" on page 193
- "New Keywords in the SQR.INI File" on page 193
- "Modified SQR Commands" on page 194

Specifying Whether a Variable or Column Supports BINARY Data

To specify that an SQR variable supports BINARY data, use the following syntax in either the `BEGIN-SETUP` section or within a `LOCAL` procedure:

```
DECLARE-VARIABLE
  BINARY $Binary
END-DECLARE
```

To specify that an SQR column supports BINARY data, use the following syntax:

```
BEGIN-SELECT
  column_name $column_name=BINARY
END-SELECT
```

New Keywords in the SQR.INI File

The `SQR.INI` file includes two new keywords in the [Default-Settings] section. Table 4 describes these keywords.

Table 4 New Entries in the [Default-Settings] Section

Entry	Value	Description
TreatBinaryColumnAsText	TRUE | FALSE	Defines whether to treat a BINARY column as a TEXT column. The default is TRUE.
ImageCompression	0 - 9	Defines the compression level when the PRINT-IMAGE command references a BINARY variable. The default is 6.

Modified SQR Commands

The following commands support BINARY data. Any command not in this list produces an error when it references a BINARY variable, column, or literal.

Note: The CREATE-ARRAY, DECLARE-VARIABLE, and OPEN commands include additional syntax to support BINARY data. (The additional syntax is shown in bold.) The other commands in the list do not require additional syntax to support BINARY data.

- CONCAT
- CREATE-ARRAY

```
CREATE-ARRAY NAME=array_name SIZE=nn
[EXTENT=nn]
{FIELD=name:type[:occurs]
[={init_value_txt_lit|_num_lit|_binary_lit}]}...
```

- DECLARE-VARIABLE

```
DECLARE-VARIABLE
[DEFAULT-NUMERIC={DECIMAL[(prec_lit)]|FLOAT|INTEGER}]
[DECIMAL[(prec_lit)]num_var[(prec_lit)][num_var
[(prec_lit)]]...]
[FLOAT num_var[num_var]...]
[DATE date_var[date_var]...]
[INTEGER num_var[num_var]...]
[TEXT string_var[string_var]...]
[BINARY binary_var[binary_var]...]
END-DECLARE
```

- DO
- EVALUATE
- EXECUTE
- EXTRACT
- GET
- IF
- LET
- LOAD-LOOKUP
- LOOKUP
- MOVE
- OPEN

```
OPEN {filename_lit|_var|_col} AS
{filenum_num_lit|_var|_col}
{FOR-READING|FOR-WRITING|FOR-APPEND}
{RECORD=length_num_lit|_var|_col[:FIXED|:FIXED_NOLF
|:VARY|:BINARY]}]
[STATUS=num_var]]
[ENCODING={_var|_col|ASCII|ANSI|SJIS|JEUC|EBCDIC|
EBCDIK290|EBCDIK1027|UCS-2|UTF-8|others... }]
```

- PRINT CODE-PRINTER
- PRINT-DIRECT
- PRINT-IMAGE

  When a BINARY variable is referenced, the contents are used as the source. When a TEXT variable is referenced, the contents refer to an external file.
- PUT
- READ
- STRING
- WHILE
- WRITE

Specification of a Separator Value for CSV File Generation

By default, comma separated value (CSV) files created by SQR use the comma as a separator or delimiter. In Hyperion SQR Release 8.3, the `SQR.INI` file includes a new entry that enables you to specify an alternate delimiter character. The new entry is shown in the [Default-Settings] section as described in Table 5:

Table 5 Entries in the [Default-Settings] Section

Entry	Value	Description
CSVSeparator	Comma | Semicolon |Space | Tab	Specifies the character used as a delimiter when creating CSV files. **Note**: If the CSVSeparator setting is missing from the SQR.INI file, the default value of Comma is used.

For example, the following setting causes SQR to use the Tab character as the delimiter when creating CSV files:

```
[Default-Settings]
CSVSeparator=Tab
```

Note: The `SQR.INI` setting to specify the CSVSeparator as a semicolon or a space is only supported with SQR Server. Using this setting with SQR iServer is not recommended and may create corrupted BQD files.

Deprecated Thin-Client Analysis Options in SQR Developer

In Hyperion SQR Release 8.3.1, the following thin-client analysis options have been deprecated:

- Pop-up thin client analysis
- Embedded thin-client analysis

The best way to EmPower a chart or a cross-tab is to EmPower it as a Hyperion Intelligence client analysis file. To do this, select Hyperion Intelligence Client Analysis on the Analysis tab in the Chart Properties dialog box.

Note: When you EmPower a chart or cross-tab as a Hyperion Intelligence client analysis file, you create an export data file that is displayed in a BQD format. The downloaded BQD file invokes the Hyperion Intelligence Web Client, if installed.

For additional information on EmPowering charts and cross-tabs, see Chapter 5, "EmPowering Charts and Cross-tabs" in the book titled *Designing Reports with Hyperion SQR Developer*.

Index

A

accessibility changes, 170
acquiring jobs, 48
administrator changes for Replace BQY option, 181
Alerting, 46
Apache Tomcat, 156
    default port, 156
    precompile script, 156
applications, Hyperion Web, removing deployed, 167
automatic distributed refresh
    BQY metadata, 41
    global flag, 41
automatic document refresh (ADR), 41

B

background jobs, 47
backward compatibility, 170 to 172
BEA WebLogic 6.x, manually configuring, 157
BEA WebLogic 7.x, manually configuring, 157
bidirectional text, 39
    ligatures, 39
BINARY data
    defining a variable or column to support, 193
    keywords in SQR.INI, 193
    modified SQR commands, 194
BQY documents
    accessibility changes and, 170
    compatibility across versions, 171
    mime types and, 178
    publishing job actions for standalone, 180
    replacing, 181
BQY Job Log, 46
BQY jobs, replacing, 181
briottbl.txt files, 20
Browse module enhancements, 176

C

Cascade Deletion, 46
CellValue_pro, 143
changes, Installation Guide, 7, 155
compatibility, backward, 170 to 172
configuring
    IBM HTTP servers
        local, 160
        remote, 161
    IBM WebSphere 5.0.1, 5.0.2, or 5.1, 157
    Oracle, 162
    SunONE_iPlanet 6.1 on Solaris, 163
    Tomcat port, 156
conflicts, port, 156
considerations, for IBM WebSphere manual configuration, 157
content, storing, between releases, 172
creating, job output as a standalone BQY document, 180
CSV file generation, 195

D

Data Layout
    Chart section, 28
DataDirect Oracle Drive, 36
DataDirect Oracle Driver, 36
default Apache Tomcat port, 156
default installation directory, 12
default installation directory, client tools, 14
default port numbers, 172
default service names, 172
default servlet names, 173
default Web service names, 173
DefaultSortOrderLang (Property), 111
deleting, job outputs for schedules, 178
deploying, Hyperion Web applications on SunONE iPlanet 6.1 (Solaris), 163
deprecated information, thin-client analysis options, 196

different releases, storing content in, 172
directories
default installation, 12, 14
directories, temporary, 168
disableImportSubgroupinfo property, 179
distributed system upgrade, 8
distributing services, tips, 174
document icons, 184
document properties
general, 104
server, 105

E

Embedded Pivot Section Interactivity, 137
end-user changes for Replace BQY option, 180
enhancements
Browse module, 176
Hyperion Foundation, 45, 175
Hyperion Intelligence, 183
Hyperion Performance Suite, 169
Object Model, 185
user interface, 176
ExportToStream (Method), 132
external authentication, nested group support and, 179

F

File Export, 35
File Import, 34
FILTER clause
defining in SQR Developer, 70
Font Configuration, 31
Foreground jobs, 47

G

group memberships, viewing or modifying, 179

H

help files, scripts, 159
Hide output collections until requested preference, 177
HOLD property, 48
HTML file formats, 88
comparison of, 88
Hyperion Foundation, enhancements, 45, 175
Hyperion Intelligence Application Configuration, 29
Hyperion Intelligence enhancements, 183
Hyperion Intelligence iServer, MAGic and, 170
Hyperion Intelligence Web Client versions, 41
Hyperion Performance Suite enhancements, 169
Hyperion Web applications, removing deployed, 167

I

IBM DB2
upgrade requirements, 9
IBM HTTP servers
configuring local, 160
configuring remote, 161
IBM WebSphere 5.0.1, 5.0.2, and 5.1, manually configuring, 157 to 158
icons, launch and document, 184
ImportSQLFile (Method), 136
INSTALL_HOME, defined, 12
installation
available types, 12 to 13
installation directory, default, 12, 14
Installation Guide changes, 7, 155
InstallHistory.xml, 7, 11
Intelligence iServer, 31
Intelligence Service, 48
iplanet61_precompile_solaris.sh sample script, 166

J

JavaScript expressions, 93
job output
creating as standalone BQY document, 180
deleting for schedules, 178
job output listings, hiding, 177
Job Output property, 178
job queuing, 46
scheduled jobs, 47
JOB_LIMIT property, 48
jobs
acquiring, 48
and the Intelligence Service, 48
disabling the job service, 48
limits, 48
jobs, SQR, output formats for, 177
joinoper operator, 42
JSPResources.properties, 22

L

launch icons, 184
legacy chart colors, 102, 104
license keys, location of, 11
locally saved BQYs, 106
Logon page, 176

M

MAGic screen magnification software, 170
memberships, group, viewing or modifying, 179
Microsoft Office Excel, 90
mime types, for BQY documents, 178
modifying group memberships, 179

N

national strings, 39
nested group support, external authentication and, 179
null values, in Pivot sections, 184

O

Object Model enhancements, 185
Object Model Exclusion List, 148
OCE File Encoding, 29
OnCellDblClick_meth, 138
open catalog extensions, 180
Open Client Connectivity, 37
Oracle 9.2.x port conflicts, 156
Oracle SQL* Net Connection, 36
Oracle SQL*Net Connection, 36
Oracle, configuring, 162
ORDER BY clause
    defining in SQR Developer, 73
output formats, for SQR jobs, 177
output, job, deleted for schedules, 178

P

port conflicts, 156
port numbers, default, 172
pre_install_db2_upgrade.cmd file, 9
precompiling, SunONE iPlanet 6.1, 166
preferences, for hiding job output listings, 177
Printout (Method), 148
properties
    disableImportSubgroup, 179
    Job Output, 178
Publish Document from the Action option, 180
publishing, standalone BQY document job actions, 180

Q

Query Builder
    limiting number of rows, 67

R

releases
    compatibility between, 170
    storing content between different, 172
removing, deployed Hyperion Web applications, 167
replacing, BQY documents or jobs, 181
replicating services, tips, 174
repository content, storing in different releases, 172
requirements, upgrade, 9
ResetDefaultSortOrderLang (Method), 128
Row Level Security, 42
row level security
    joinoper operator, 42

S

samples
    Apache Tomcat precompile.sh, 156
    iplanet61_precompile script, 166
schedules, deleted job output for, 178
scripts
    Apache Tomcat precompile.sh, 156
    help files and, 159
    iplanet61_precompile_solaris.sh sample, 166
    startCommonServices, 179
separator value, in CSV file generation, 195
servers, Web, supported, 174
service names, default, 172
services
    default settings, 172
    stopping before updgrading, 162
servlets
    default settings, 173
    testing, 162
Session (Object), 186
setting
    general chart properties, 104
settings
    services, default, 172
    servlets, default, 173

setupAixPlatform.bin, 10
setupHpBrioPlatform.bin, 10
setupLinuxBrioPlatform.bin, 10
setupSolarisPlatform.bin, 10
setupWinPlatform.exe, 10
SideLabelValues (Array), 139
slicers, defining in SQR Developer, 72
Sort Order Language, 28
SortOrderLang (Property), 109
specifying, job output folders, 178
speed menu frames, Hyperion Intelligence iServer, focusing on, 170
SQR, 27
SQR jobs, output formats, 177
SQR.INI, new keywords in Default-Settings, 193, 195
startCommonServices script, 179
stopping services before upgrading, 162
storing content, between releases, 172
SunONE iPlanet 6.1
    configuring, 163
    precompiling, 166
supported Web servers, 174
Sybase, 37

T

temporary directories, 168
testing, servlets, 162
Tomcat, port conflicts, 156
TopLabelValues (Array), 140

U

UILanguage (Property), 117
Unicode, 27
    BI Application Configuration Files, 29
    BQY Sort Order Language, 28
    DB Connectivity, 36
    File Export, 35
    File Import, 34
    file path handling on Unix, 31
    Font Configuration, 31
    OCE File Encoding, 29
Unicode Compliant Font Preferences, 31
    Dashboard Preferences, 32
    General Preferences, 32
Unicode Compliant Fonts, 31
Unicode Font Configuration, 31
Unicode User Defined Languages, 18
UnicodeEnabled (Property), 108
Unix, 31
UNIX temporary directories, 168
unzipWebComponentHelp.bat, 159
unzipWebComponentHelp.sh, 159
Upgrade Prerequisites, 9
upgrade requirements, 9
upgrading, 10
upgrading services, stopping before, 162
upgrading, distributed system, 8
user interface enhancements, 176

V

VBA functions, 99
viewing group memberships, 179
vpd.properties, 7

W

Web applications, removing deployed Hyperion, 167
Web servers, supported, 174
Width (Property), 185