HP

Business Process Insight

For the Windows® Operating System

Software Version: 2.20

Reference Guide

Document Release Date: October 2007
Software Release Date: October 2007

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Java™ is a US trademark of Sun Microsystems, Inc.

Microsoft® is a US registered trademark of Microsoft Corporation.

Oracle ® is a registered US trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

This manual's title page contains the following identifying information:

- Software version number, which indicates the software version
- Document release date, which changes each time the document is updated
- Software release date, which indicates the release date of this version of the software

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

http://ovweb.external.hp.com/lpe/doc_serv/

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP Software Support web site at:

www.hp.com/go/hpsoftwaresupport

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

www.managementsoftware.hp.com/passport-registration.html

Contents

1 Introduction 9

2 HPBPI Architecture 11

High-Level Component Architecture 12

Business Process Dashboard 13

The Dashboard and JSPs 15

The Dashboard and Business Process Metrics 15

Email Notifications and the Dashboard 16

Intervention Client 16

HPBPI Database 18

Flow Data 20

Business Process Metric Data 20

Metric Threshold Definitions 21

Event Hospital Data 21

Model Repository Data 21

Email Notification Data 22

HPBPI Server 23

Business Impact Engine 25

Business Process Metric Definer 29

Metric Engine 31

Repository Explorer 33

Notification Server 34

Business Event Handler 37

Modeler 41

Repository Server 42

Administration Consoles and Interfaces 43

HPBPI Security 44

Where to go Next 44

3 Integrating with HPBPI 45
HPBPI Integration Points 46
HPBPI Adapters for HP BTO Software Integration 50
Service Source for Business Availability Center 51
HPBPI SOA Manager Adapter 52
HPBPI and OVIS Probes and Alarms 53
HPBPI Operations Manager Adapter 55
HP Service Desk Adapters 56
HPBPI Accelerator for SAP Applications 58
iWay Integration 59
HP Operations Dashboard 60
Self-Healing Services 62
4 HPBPI and HP Business Availability Center 65
HPBPI and Business Availability Center Integration 66
Business Availability Center and HPBPI Terminology 67
Integration Options 68
Design-Time Integration 71
Run-Time Integration 73
Request for Updates of BPI Steps 74
HPBPI and Business Availability Center Data Sampling 75
Using Business Availability Center as a Source of Operational Service Status Data . . . 76
Configuring Operational Service Sources 76
Configuring a Business Availability Center Data Sample Destination 78
Creating an XML File Source Adapter 80
Using the Metric Definer to Create an Entities File 81
Creating a My Business Availability Center Application for HPBPI 84
Description of Data Samples Sent to Business Availability Center 86
Business Availability Center System Configuration 88
5 HPBPI and SOA Manager 89
SOA Manager and HPBPI Integration 90
Business Events and SOA Manager 93
Configuring Access to the SOA Manager Adapter 94

6 HPBPI and OVIS 97
HPBPI and OVIS Integration 99
HPBPI and OVIS Design-Time Integration 99
HPBPI and OVIS Run-Time Integration 101
Reporting on the Operational Status of Your HPBPI System 104
Reporting SLO and SLA Violations 104
Custom Probes 105
Creating a New Service Group for an HPBPI Service 107
Configuring Alarms and SLOs 115
Defining Objective Information for HPBPI Monitored Services 115
Defining Service Level Agreements for HPBPI Monitored Services 121
Making Sure OVIS Adds Alarm Data to Its Database Tables 122
Defining Probe Locations 122
Configuring Probes for Multiple HPBPI Servers 123
7 HPBPI and HPSD 125
HPBPI and HPSD Service Call and Incident Information 126
HPSD Web API 127
Automatic HPSD Service Mapping 128
Defining HPSD Custom Fields 128
A Database Schemas 133
Building Applications to Use the HPBPI Metrics Data with Microsoft SQL Server . . . 134
Oracle and SQL Server Data Type Definitions 135
Business Metrics Schema 136
Alerts Facts 136
Facts Values 140
Statistics Facts 146
Dimension Tables 151
Business Metric Custom Types 166
Business Metric Failure Messages 167
Metric Views 168
Metric Values 169
Flow Schema 172
Flows 173

Flow Instance 175
Nodes 179
Node Instance 182
Node Instance Started Times 184
Node Instance Completed Times 185
Arcs 186
Services 187
Node2Resources 189
Business Entity Schema 191
Data item names 191
Data types 191
Keys 191
Data Objects 192
Data Definition Instances 193
Business Event Handler Schemas 195
Business Event Handler Event Store 195
Event Hospitals 197
Complete List of HPBPI Database Tables 200

B Expression Grammar in Flow, Data and Filter Definitions 205
Grammar 206
Function Return Values 208
Functions for Dates and Times 208
Functions for Identifying String Values 208
Functions for All Property Types 209
Flow Progression Rules 210
Methods for Progression Rules 212
Case Sensitivity for Expressions 216
Expression Properties 216
Expressions with String Constants 216

C Coercion Rules 217
Assignments 218
Expressions 219

1 Introduction

This guide provides reference information relating to Business Process Insight (HPBPI). This reference information covers HPBPI and the components that HPBPI integrates with as follows:

- Chapter 2, HPBPI Architecture

  This chapter describes the architecture of the HPBPI system and how the HPBPI components relate to each other.

- Chapter 3, Integrating with HPBPI

  This chapter describes the different integration points within your HPBPI system.

- Chapter 4, HPBPI and HP Business Availability Center

  This chapter provides details of how you can integrate HPBPI and Business Availability Center (BAC). It describes how you can export HPBPI Business Flow details and import these details into BAC such that BAC can provide HPBPI with operational service status details.

  You can also configure BAC and HPBPI to enable you to view HPBPI Business Metric and Metric Threshold details through the BAC Dashboard.

- Chapter 5, HPBPI and SOA Manager

  This chapter provides details of how HPBPI integrates with HP SOA Manager. It describes how to import the required SOA Manager business services and link them to your business flows.

  You also configure your HPBPI system to receive SOA business events by configuring an event source for SOA Manager. This is described in the *HP Business Process Insight Training Guide - Business Events*.

- Chapter 6, HPBPI and OVIS

  This chapter provides details of how HPBPI integrates with HP OpenView Internet Services (OVIS), specifically, to configure customized HPBPI probes for OVIS.

  Installing OVIS probes is an optional task, and if you want to use OVIS to report OVIS metric information relating to HPBPI flows, you need to install the probes and configure them as described in Chapter 6, HPBPI and OVIS.

  You can also configure an HPBPI Server to receive service impact information from OVIS. You configure your HPBPI Server to receive these impact reports using the Administration Console as described in the *HPBPI System Administration Guide*.

- Chapter 7, HPBPI and HPSD

  If HP Service Desk (HPSD) is a component in your system, you can configure HPBPI to link to HPSD Incident reports and Service Calls, and display their details through the Business Process Dashboard. This chapter provides details of how HPBPI integrates with HPSD through the HPBPI Dashboard in order that you can achieve this.

- Appendix A, Database Schemas

  This appendix details the database schemas that are defined to generate reports from the HPBPI impact data and for use by the Business Impact Engine, Metric Engine and Business Events Handler.

- Appendix B, Expression Grammar in Flow, Data and Filter Definitions

  This appendix lists the rules for the grammar that can be used for business flow progression rules and expressions, and expressions within business process metric filter definitions.

- Appendix C, Coercion Rules

  This appendix describes the rules for how properties are coerced when evaluating binding, filter and assignment expressions within the HPBPI Modeler.

2 HPBPI Architecture

This chapter provides a high-level description of the architecture of the HPBPI system. The architecture is presented to provide an understanding of the components and concepts that are described in later sections and chapters. You can read the early sections of this chapter to gain an overview of the architecture, and then reread this chapter later, if you want more detail.

This chapter also lists which components can be customized and where to find more information about them.

Specifically, the chapter covers the following topics:

- A high-level representation of the HPBPI component architecture; see Figure 1 on page 12.
- The HPBPI Business Process Dashboard, which is used to display the impact information from the HPBPI system; see section Business Process Dashboard on page 13.
- The HPBPI Intervention Client, which provides you with access to active business flows and flow data; see section Intervention Client on page 16.
- The HPBPI database and how it is used by the HPBPI components; see section HPBPI Database on page 18
- The Metric Definer, which you use to define business process metrics and metric thresholds for your flows; see section Business Process Metric Data on page 20 and section Metric Threshold Definitions on page 21.
- The HPBPI Server components; see section HPBPI Server on page 23.
- The HPBPI Modeler and the data used and stored by the HPBPI Modeler; see section Modeler on page 41.
- The administration GUIs and consoles used within the HPBPI system; see section Administration Consoles and Interfaces on page 43.

High-Level Component Architecture

Figure 1 shows a high-level diagram of the HPBPI architecture.

Figure 1 HPBPI Architecture

Business Process Dashboard

The HPBPI Business Process Dashboard is a Web-based interface for viewing the progress of your business flows. It enables you to monitor business flows and flow instances, including the business events, operational services and business process metrics that you have modeled and deployed using the HPBPI Modeler and the Metric definer.

The HPBPI Dashboard also enables you to associate the service information received from HP Operations Manager (HPOM), OpenView Internet Services (OVIS) and HP SOA Manager with HP Service Desk (HPSD) Service Calls and Incidents; see Chapter 7, HPBPI and HPSD.

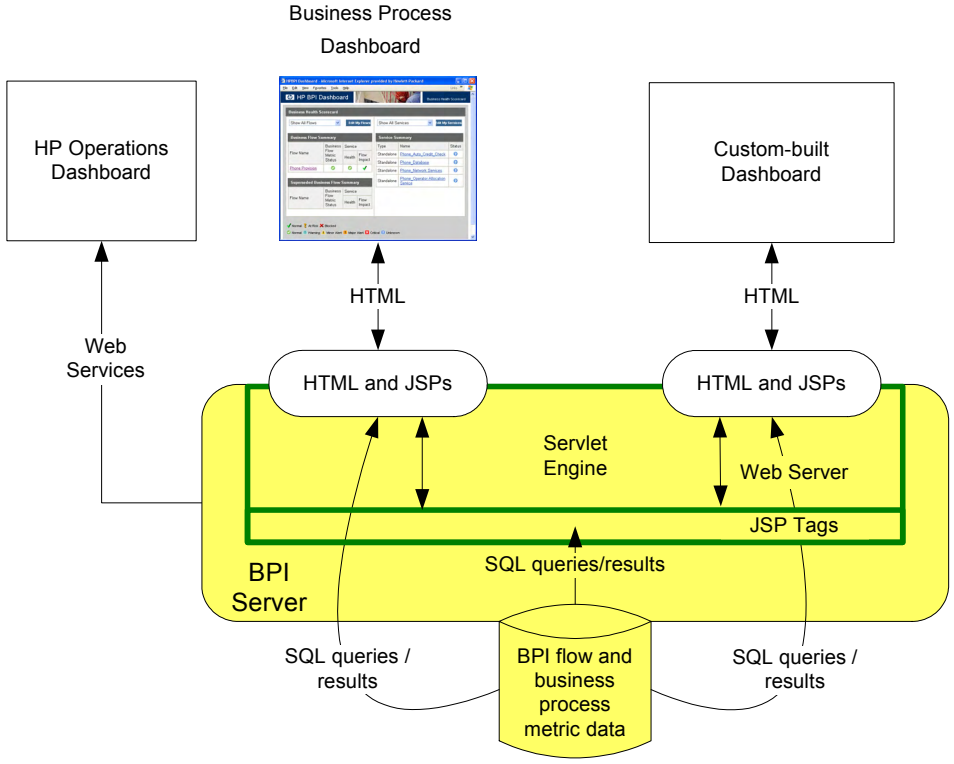
Specifically, the HPBPI Dashboard provides the following through a number of different views and graphical displays:

- An overall business health scorecard, which provides an overall view of the health of business flows that you have deployed.
- Flow instance information related to the flows that you have deployed.
- Business process metric information and threshold alerts related to individual flows and flow instances that are deployed.
- Reports based on the statistical information collected from the business process metrics that you have defined.
- Information relating to the status of the operational Services that your flows are linked to within HPOM, HPSD, SOA Manager and OVIS.
- HP Service Desk Service Calls and Incidents for operational Services, where the information is available.
- Where appropriate, details of the status of your 60-day `Instant On` license. The Dashboard displays the number of days left before the `Instant On` license expires and HPBPI stops running.

If you have installed the HPBPI Business Process Dashboard on a different system to the HPBPI Server, you do not get information about the status of your license.

The HPBPI Dashboard comprises a set of HTML pages, which are created using Java Server Pages (JSP). JSP is a simple way to create dynamic Web pages, which are both platform independent and server independent. A Servlet Engine is used to manage these pages; this Servlet Engine is Tomcat and is installed with the HPBPI Server.

Figure 2 HPBPI Business Process Dashboard



The design of the HPBPI Dashboard means that it is possible to customize it for your specific business flows (see section The Dashboard and JSPs on page 15). This enables you to present data on your business flows within an interface that is tailored to your business area. However, you do not need to make any changes or customizations to the Dashboard as it can be used, without modification, to show information about any flows that you create.

Refer to the *Business Process Insight Integration Training Guide - Customizing the Business Process Dashboard* for information on how to customize the Dashboard and also how to create annotations for your business flows.

The Dashboard and JSPs

The Business Process Dashboard uses a JSP custom tag library to communicate with the HPBPI components. This reduces to a minimum the amount of programming required to make modifications to the Dashboard. Refer to the *Business Process Insight Integration Training Guide - Customizing the Business Process Dashboard* for details of the JSP Tags and how to develop a personalized Dashboard based on these tags.

JSP technology separates content generation from presentation and takes advantage of reusable tags and objects, simplifying the maintenance of your Web applications. Using JSP enables you to access the HPBPI flow data and present it in a form of your choice to your business managers. This allows you to create queries and enhance the flow data with existing business data. This, combined with the JSP custom tag libraries, provides an easy way to modify the example Dashboard or develop a new dashboard to provide the impact information for a specific flow.

The Dashboard and Business Process Metrics

The Business Process Dashboard displays tables, graphs and dials relating to the:

- overall health of the business flows relative to the business process metrics that are deployed within HPBPI.
- business process metrics and metric thresholds that you have configured, including the alert status and individual alerts.
- details of the individual flow instances and metric instances within the HPBPI system.
- historical statistical information relating to the business process metrics.

This business process metric data is displayed automatically through the HPBPI Dashboard as a result of your defining the business process metrics and business thresholds for your flows.

Business process metrics and metric threshold definitions are set up using the Metrics definer, which is described in more detail in the *HP Business Process Insight Integration Training Guide - Defining Business Process Metrics*.

Email Notifications and the Dashboard

In addition to proactively monitoring business flows through the Dashboard, you can configure HPBPI to send email notifications relating to your flows through the Notification Server (see section Notification Server on page 34). These email notifications contain information relating to the business flows that you are monitoring.

The notification emails can also be configured to contain links to your HPBPI Dashboard, where you can then find out more about the status of your IT services and their impact on your business flows.

Intervention Client

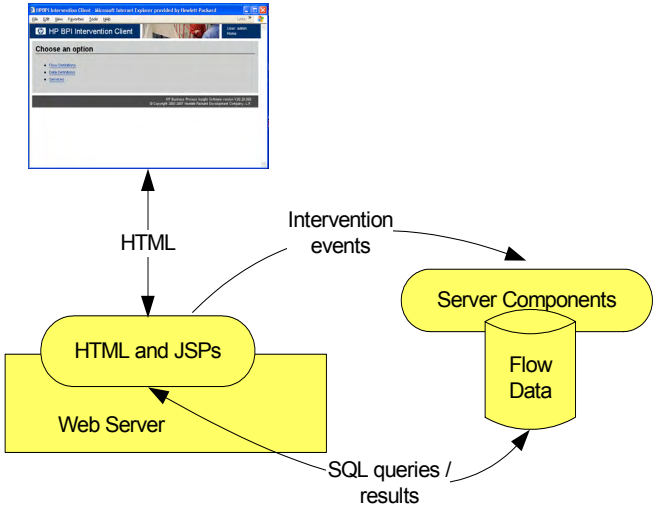
The Intervention Client enables you to access flows that you have deployed in order to modify or delete Flow instances and their associated Data instances. You might need to do this to resolve problems with the flow, or its data, usually after a period of new development or where progression rules are not behaving as expected. For example, you might need to:

- manually progress and delete flow instances.
- update and delete data instances.
- update the status that HPBPI records for an operational service.

  Note that the Intervention Client does not have an effect on the service as it is defined within HPOM or OVIS; it makes changes only to the recorded status within HPBPI.

The Intervention Client is a secure, Web-based, console that uses Web authentication to enable you to make the modifications to your business flows.

Figure 3 Intervention Client



The Intervention Client is similar in architecture to the Business Process Dashboard in that it is a set of JSPs that use SQL to access the HPBPI database; see Figure 3 on page 17. The Intervention Client also uses Tomcat as its Servlet Engine and Web Server.

The Intervention Client is aimed at the person who is managing the HPBPI system and who has authority to make these modifications to the business flows. More details of the Intervention Client and how to use it in your solution are provided in the *HPBPI System Administration Guide*.

HPBPI Database

HPBPI uses a relational database to record the following information:

- Flows and any related data
- Business process metrics
- Metric threshold definitions
- Metric threshold alerts
- Metric statistics
- Event hospital data, including the business event data
- Operational service status data
- Model Repository data
- Email notification data

This information is used by the HPBPI components for monitoring and progressing flow instances, plus reporting flow impact data, threshold violations and business process metrics through the Dashboard. It is also used by the Business Impact Engine to identify the Flow, Data and Service definitions that have been deployed using the HPBPI Modeler.

The database holds all the information that HPBPI needs to operate, with the exception of a number of configuration files, which are located under the HPBPI installation directory.

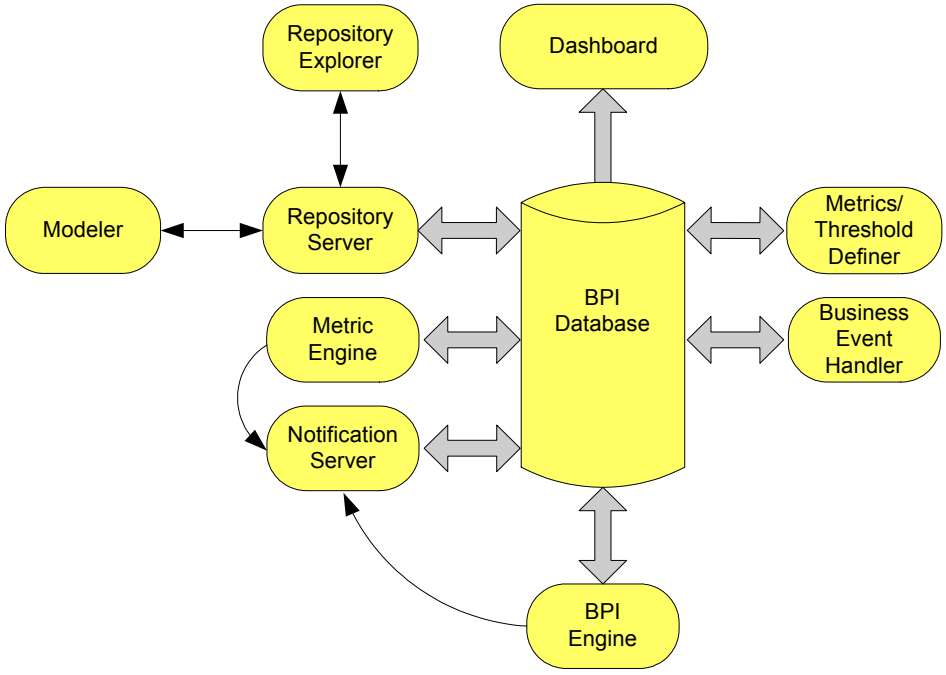
You can access the flow and business process metric data held in the HPBPI database directly, using SQL, and generate your own custom-built reports. You might want to do this in order to integrate these reports into your own reporting applications.

The data schema description that represents HPBPI information in the database is fully described in the *Business Process Insight System Administration Guide*. Be aware that this appendix describes only those tables that are supported for access. There are other tables, which are specifically for use by the HPBPI components and must not be modified, or accessed. These database tables are listed for completeness in the appendix, but are not described. As an example, you might need to know the name of all the tables that are relevant to HPBPI for backup and recovery purposes; however, the tables that are listed and not described must not be modified or used for reporting purposes.

Figure 4 shows a high-level diagram of the relationship between the database and the HPBPI components. The database can be installed and configured on the same system as an HPBPI Server, or on a system that is remote from the HPBPI Server; this is not shown in Figure 4.

Refer to the *Business Process Insight Installation Guide* for details of installing the schema or database files into a local or remote database.

Figure 4 HPBPI Database



The HPBPI components' use of the database tables is described in the following sections.

Flow Data

This is the schema for the Business Impact Engine data. The HPBPI database maintains the state of the Engine database objects (or definitions), for example, the business Data definitions and business Flow definitions, within the HPBPI system. These definitions comprise status information required by the Business Impact Engine for:

- Flows
- Data
- Services

These Flow, Data and Service components, plus the Event information, are all defined through the HPBPI Modeler. The data required to populate the Event definitions is obtained from your business applications through the Business Event Handler.

Defining business flows is described in more detail in the *HP Business Process Insight Integration Training Guide - Modeling Flows*.

Business Process Metric Data

The HPBPI database holds the data collected through the metric tables as a result of the progress of business flows and the status of the flows relative to these defined metrics. Business process metrics are evaluated as part of the flow data as nodes are progressed by the Business Impact Engine. A business process metric database table is populated using database triggers from the Nodes database table. The Metric Engine then processes the data from this metrics table and uses the results to calculate statistics and populate the remaining metrics tables. The results of these calculations are then presented to you through the Business Process Dashboard.

You can also use reporting applications to access the metric and statistical information in the database tables and generate reports from the data that is collected, or you can develop your own custom-built dashboard.

You can read more about the Metric definer and the metric data in the *Business Process Insight Integration Training Guide - Defining Business Process Metrics*. Building a customized Dashboard is described in the *Business Process Insight Integration Guide - Customizing the Business Process Dashboard*.

Metric Threshold Definitions

You can optionally define thresholds for your business process metrics. If you do this you might also want to be notified when these thresholds fall outside acceptable values. HPBPI enables you to create metric threshold definitions for each business process metric so you can then be notified when these metric threshold values are violated.

Metric threshold alerts are shown:

- using the Business Process Dashboard
- as email alerts using your email system
- as HP Operations Manager alerts from other HP BTO software applications

You can read more about metric thresholds and creating them using the Metric definer in the *Business Process Insight Integration Training Guide - Defining Business Metrics*.

Event Hospital Data

Details of the events that are rejected by the Business Impact Engine because it does not recognize them are written to the Event Hospital table in the HPBPI database. From here they can be accessed, modified and resubmitted to the Business Impact Engine at a later date, if required.

Model Repository Data

The design-time Flow modeling information is held in the Model Repository tables in the HPBPI database. This is the information that you enter using the HPBPI Modeler.

Email Notification Data

These are the database tables relating to the information about the Notification Server subscriptions. The information in these tables is defined through the Notification Server Administration Console.

In addition to data relating to subscriptions, these tables contain data for the Notification Server retry mechanism.

HPBPI Server

The HPBPI Server is the core of the HPBPI system; it is where the business and operational events are received, and their impact on the flows evaluated. The HPBPI Server is responsible for:

- Maintaining the Flow, Data, Event and business process metric definitions through the HPBPI database.
- Enabling management for the Model Repository data, including the ability to browse and print the content of the Repository.
- Monitoring business events, through adapters, in order to maintain the flow context for the business.
- Monitoring services and reporting any change that causes an impact on the flows.
- Defining business process metrics and metric thresholds.
- Monitoring metric thresholds and reporting on the threshold violations.
- Validating the status of flows according to progression rules that you have specified.
- Sending email notifications for flow impact alerts that have been subscribed to.
- Sending email notifications for metric threshold alerts that have been subscribed to.

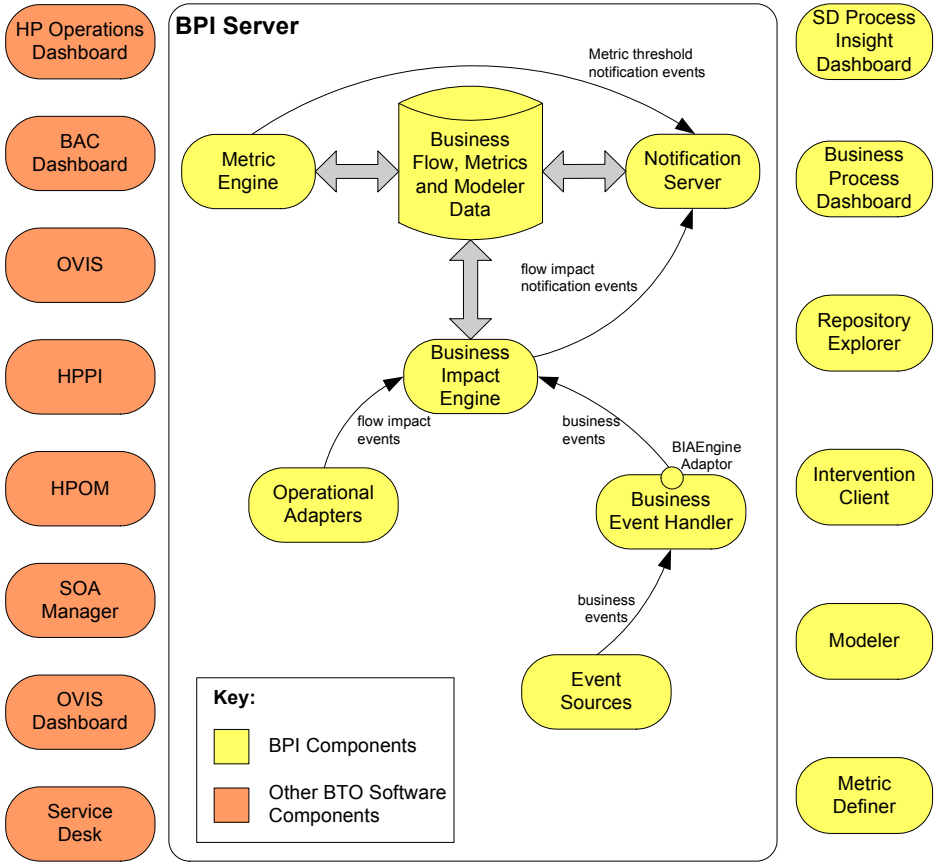
The components of the HPBPI Server that provide these functions are the:

- Business Impact Engine
- Metric Engine
- Business Process Metrics definer
- Repository Explorer
- Notification Server
- Business Event Handler

HPBPI can also accept service impact reports from other HP BTO Software products; see section Chapter 3, Integrating with HPBPI.

The individual HPBPI Server components are shown in the following diagram and described, in more detail, in the following sections.

Figure 5 HPBPI Server Architecture

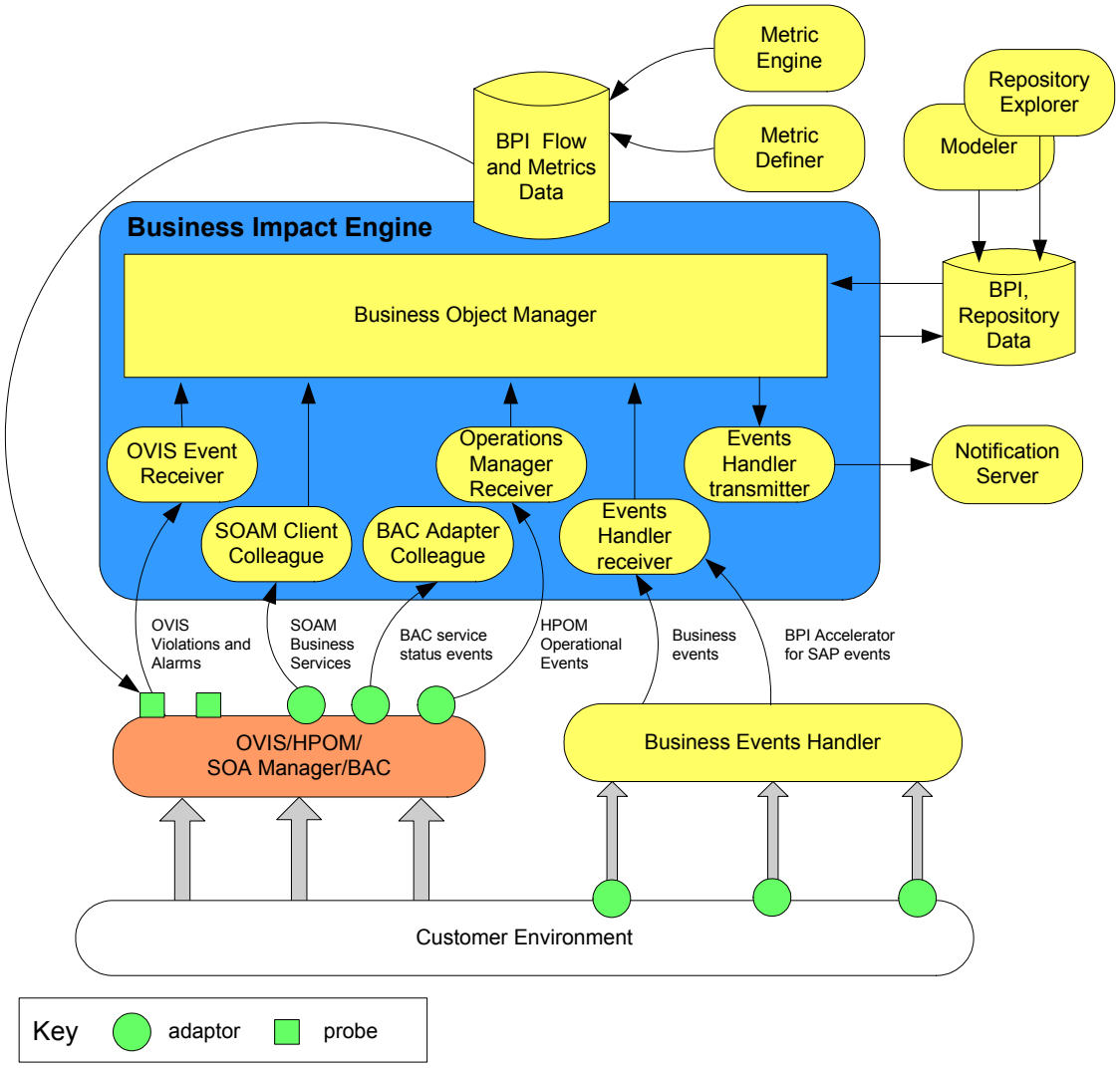


Business Impact Engine

The purpose of the Business Impact Engine is to process operational and business events and derive business impact information from these events. Using the information defined in business flow definitions, the Business Impact Engine raises flow impact alerts when an underlying operational service or business event, which is relied on by the flow, falls outside defined thresholds.

Business Metric Threshold violations are managed by the Metric Engine as described in section Metric Engine on page 31.

Figure 6 Business Impact Engine Architecture



When a flow impact threshold violation is identified, the Business Impact Engine sends an impact alert to the Notification Server. In addition, the Business Process Dashboard checks the database for status changes resulting from these impact alerts to present the alerts to the business manager. This enables you either to be notified, or to proactively monitor the status of your business flows, according to your preferences and requirements.

The Business Impact Engine comprises the following components, which perform functions internal to HPBPI. These components are not generally exposed in any of the interfaces; however, you might see them referenced in log and error messages:

- Business object manager

  This manages the business objects within the Business Impact Engine, for example, the HPBPI business Data definitions and instances and the business Flow definitions and instances. The business object manager also receives events from OpenView Internet Services (OVIS), Operations Manager and the Business Event Handler, plus specific internal events, such as flow progression events. Where appropriate, these events are propagated to the relevant business objects.

  The business object manager is generally a passive component and takes actions only when it receives an event. The exceptions are the `Model Instance Cleaner` and the `Engine Flow and Data Instance Cleaner`, which are active. These instance cleaners are described in the *Business Process Insight Administration Guide*.

- HPBPI Operations Receiver and Adapter

  The HPBPI Operations Receiver accepts operational status events from the Operations Adapter, converts them into an HPBPI event format and then passes them on to the business object manager.

  Operational events are status events, providing information on the status of the underlying applications, and system infrastructure, for example the CRM system, order processing system, routers or firewalls.

  The HPBPI Operations Adapter communicates with HP Operations Manager to obtain the status of HP Operations Manager (HPOM) services. These are the HPOM services that you link into your business flow using the HPBPI Modeler.

- SOA Manager Client Colleague

  The client colleague enables HPBPI to import SOA Manager business service definitions using the SOA Manager adapter. It then propagates the status events from these definitions in order to report on the status of the SOA Manager business services that you have configured.

- BAC Adapter Colleague

  This adapter colleague enables HPBPI to obtain the service status data from BAC using the BAC open API. The service status data is then used to report on the status of BAC services that are linked to node in your HPBPI Business Flows.

- OVIS Event Receiver

  The OVIS Event Receiver polls for OVIS alarms and service level violations. These provide:

  — operational service status information in the form of alarms.

  — SLO and SLA violation information.

  As a result of receiving these alarms and violations, HPBPI can send email notifications to users configured to receive them through the Notification Server; these impact messages can have links to the relevant page in the Dashboard.

  The OVIS Event Receiver converts service impact alarms into HPBPI operational events, which then update the status of the appropriate service definition in the business object manager.

  The OVIS Event Receiver also converts SLA and SLO violations into HPBPI events and sends them directly to the Notification Server.

  OVIS alarms inform you of the status of your Internet and other services. This is in terms of how effectively they are running against the criteria that you configure within OVIS.

- Event Receivers and Transmitters

  The Event Receiver accepts business events, using RMI, as Java objects from the Business Event Handler, and passes them on to the Business Object Manager.

  You can also configure a business event queuing mechanism. In this case, all new business events are placed on a queue by the Business Event Handler. This means that the Business Event Handler does not need to wait for the Business Object Manager to process the business event and can immediately return to monitoring the event source and process the next incoming business event.

If you choose not to use a queue for business events, the Business Event Handler waits until the business event has been processed by the Business Object Manager. This includes waiting for all the necessary information from the event to be committed to the database.

There can be more than one Event Receiver, to improve performance if required. However, if you are not using a queue for business events, there is only one Business Object Manager, in which case there is a limit to the benefit of adding more Event Receivers.

The Event Transmitter accepts alerts from the Business Object Manager, converts them into an RMI message and sends them to the Notification Server.

Business Process Metric Definer

The Metric definer is a Web-based GUI that enables you to define business process metrics and metric threshold definitions for deployed flows.

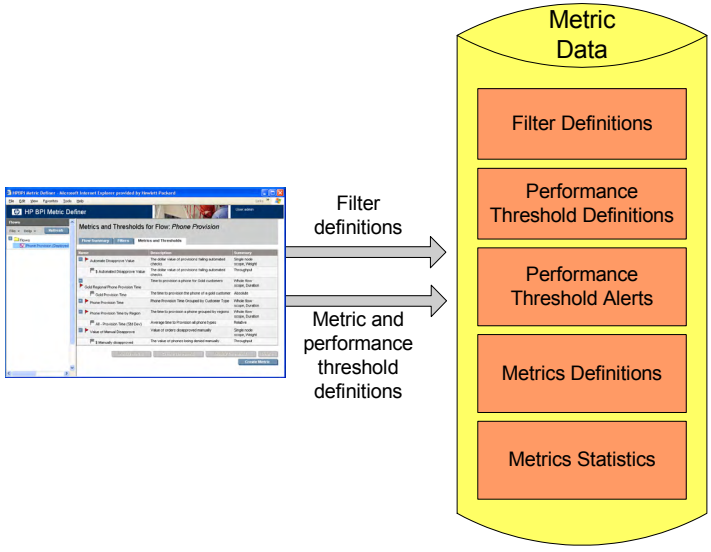
A business process metric is a business measurement that has a specific meaning within your business and can be used to record information about the business flow that you are monitoring. Statistics, such as average and maximum, can be calculated using metric instance data. In addition, metric thresholds can be set on both the individual metric instances and on the statistics generated from the metric instances. When metric thresholds are violated, threshold alerts are generated and can be reported through the Business Process Dashboard. You can also configure the Notification Server to send alert messages when a metric threshold is violated.

Within HPBPI, the process of defining business flows is separated from the process of defining business process metrics for these flows. You use the HPBPI Modeler to define your business flows and you use the Metric definer to define the business process metrics and metric thresholds for these business flows. You define business process metrics for a business flow after the flow has been deployed; this provides more flexibility for defining and modifying business process metrics without the need to redeploy your business flows.

You can also use filters to further target the statistical data that you want to collect. This has the added benefit of reducing the amount of data stored in the database and therefore maintaining the HPBPI system performance.

Figure 7 shows the Business Process Metric definer relative to the Business Metric data stored in the HPBPI database.

Figure 7 HPBPI Business Process Metric Definer



You can define metrics for:

- Single nodes
- Multiple nodes
- The whole flow

These metrics can be duration based or weight based. In addition, you can define a custom metric.

Data are collected based on this configuration, and thresholds can be defined to report on individual flow instances or on statistics for multiple instances, such as averages and standard deviations over a specified time period.

You can also use the Metric Definer to export Flow definitions, including Business Process Metrics, to an XML file, and subsequently import these definitions into BAC as an XML file Adapter.

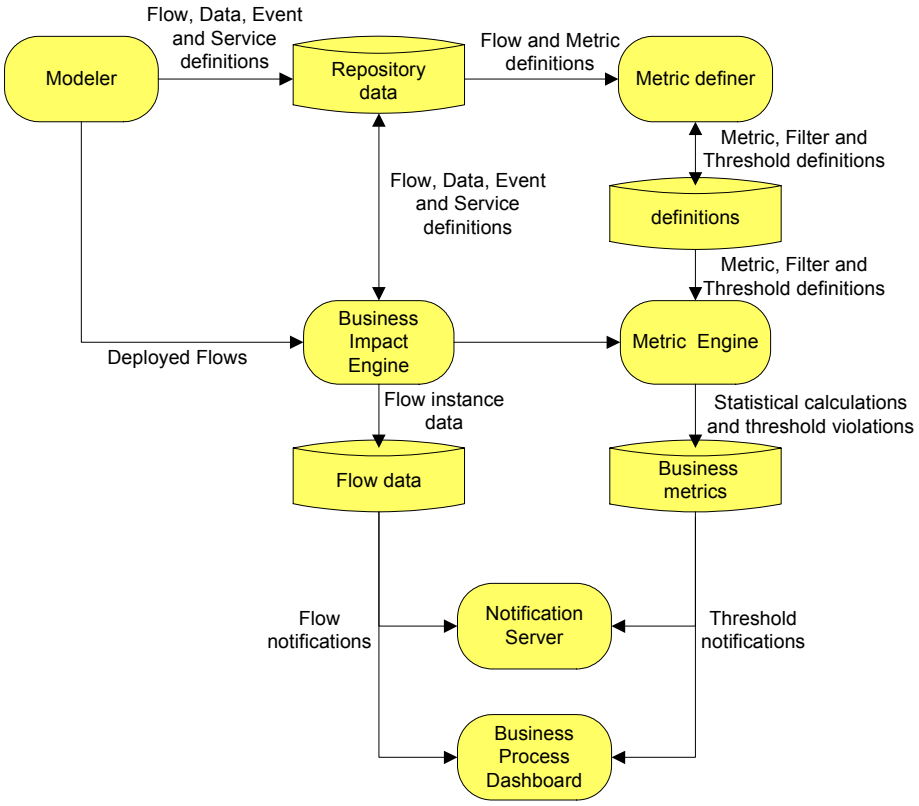
How the Business Process Metrics definer interworks with the Metric Engine is described in section Metric Engine on page 31.

Metric Engine

The Metric Engine is the component of the HPBPI Server that analyzes and provides the statistical results from the business process metric and metric threshold data that you define. The Metric Engine takes data about the individual instances from the Business Impact Engine and combines this data with the data you enter through the Metric definer to collect and report on the required information about the flow.

Figure 8 shows the relationship between the flow impact data and the metric data that you define.

Figure 8 Flow Impact and Metric Data Relationship



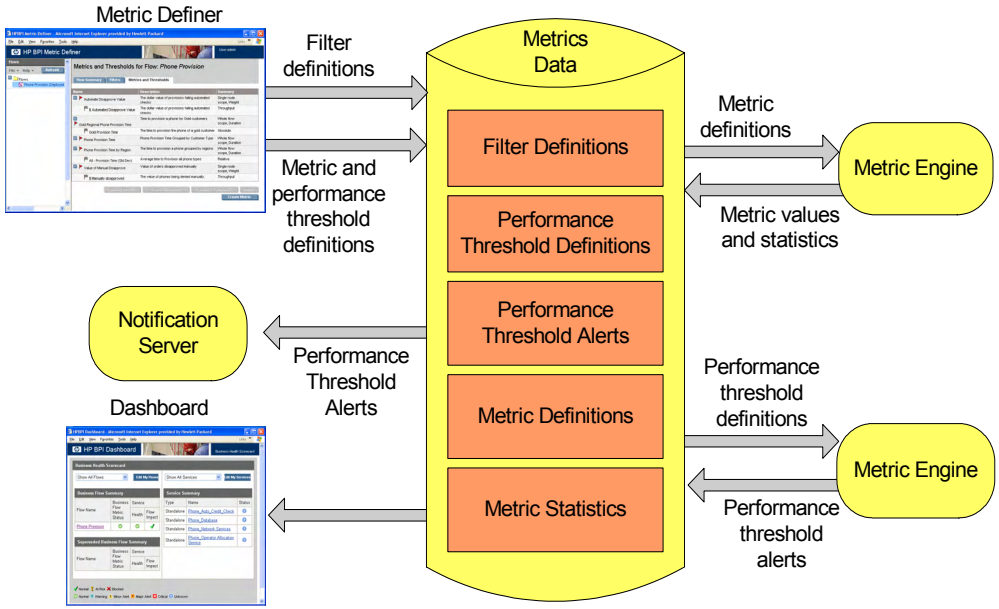
The business process metrics and metric thresholds are configured through the Business Process Metrics definer; see section Business Process Metric Definer on page 29.

The Metric Engine maintains business data for the business flows that are deployed according to the business process metrics that you have defined. It also maintains statistical information about the metric thresholds that are defined.

As an example, if you have defined a business process metric and metric threshold to record the backlog of flow instances at a specific node in the flow, the Metric Engine records and maintains the statistics required to calculate this backlog. It then reports the results through the HPBPI Dashboard.

Figure 9 shows how the metric data is used by the Metric Engine to report statistics and threshold violations to other HPBPI components.

Figure 9 HPBPI Business Process Metric Engine



The statistics that the Metric Engine calculates are stored in Metric database tables and are used by the HPBPI Dashboard and can also be used by other reporting tools if required. The database tables also hold information used to determine if any metric thresholds have been violated. These violations are reported through the HPBPI Dashboard and can also be reported through the Notification Server or other HPOM applications.

Repository Explorer

The Repository Explorer is a Web-based interface that enables you to view, print and manipulate the contents of the Model Repository. The Model Repository is a set of database tables that hold the data for the business processes that you have defined using the HPBPI Modeler.

Note that the data presented by the Repository Explorer might differ from the data that is deployed to the Business Impact Engine. This is because you might have modified the flow within the HPBPI Modeler, but not yet redeployed the flow.

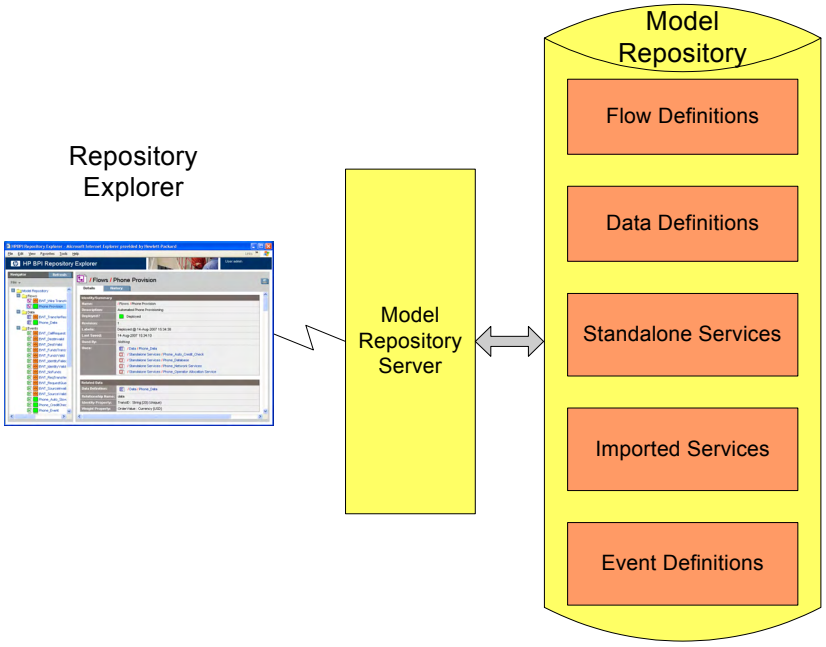
The Repository Explorer also provides access to information about all the superseded versions of the business flow that are currently deployed, enabling you to view and export details of the superseded flows.

Using the Repository Explorer you can:

- View all the details of your currently deployed and superseded flows in a tabular form.
- Print currently deployed and superseded definitions.
- Export the latest versions of your currently deployed definitions.
- Export the superseded versions of definitions.
- Remove (delete) superseded definitions.
- Import previously exported definitions.
- View, definitions that have been deleted using the HPBPI Modeler. These definitions appear in the Recycled folder within the Repository Explorer, from where they can be permanently deleted.

In summary, the Repository Explorer is a tool that you can use to browse and manage the Model Repository data and is accessed through the Repository Server as shown in Figure 10.

Figure 10 Repository Explorer



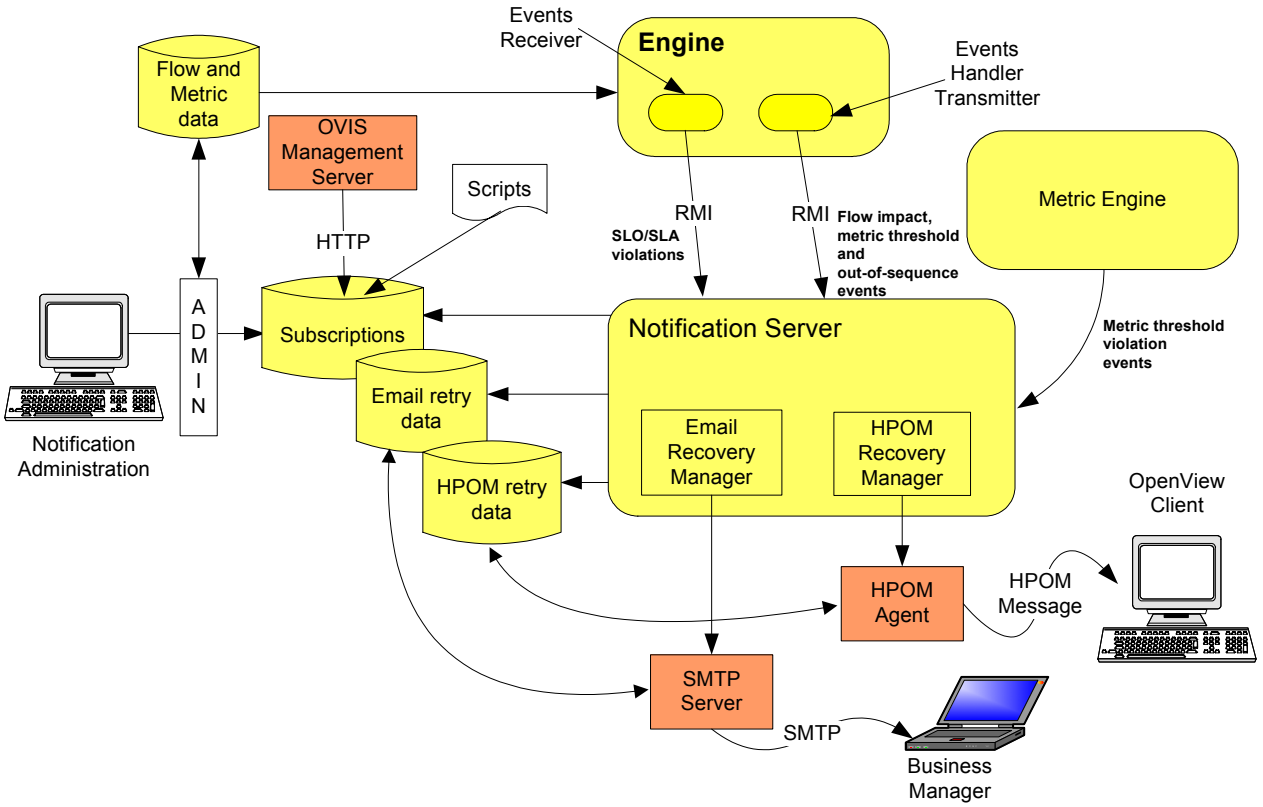
Notification Server

This component of the HPBPI Server is responsible for notifying you of the flow-impact, out-of-sequence and metric-threshold alerts that you have configured through the Notification Server Administration Console. The Notification Server Administration is described in section Administration Consoles and Interfaces on page 43.

Notifications can be sent either through an SMTP server to an email client, or as an HPOM message to an HPOM Server. There is a retry mechanism for both types of notifications if either the SMTP server or HP Operations Manager is not available for any reason. Notifications are queued and retried after a configurable interval.

Figure 11 shows the architecture of the Notification Server and its relationship to other HPBPI components.

Figure 11 Notification Server



Business alerts are created by the Notification Server as a result of receiving:

- flow-impact and out-of-sequence notification events from the business object manager through the Business Event Handler Transmitter.
- SLO and SLA violation notification events directly from the Events Receiver.
- metric threshold violation notification events from the Metric Engine.

The Notification Server sends these events as business alerts to you, based on a set of filters. These are filters that you create and that specify the impact events for which you want notification. You can filter email notifications

based on event name, event type and the name of the business flow. SLAs are filtered according to Customer SLA name, and SLOs are filtered according to Customer, Service Group and SLO name.

In addition to sending alerts when an event is received, the Notification Server can also run any script (`.bat` file) that you have created. Creating scripts is described in the *HPBPI System Administration Guide*.

Email Notification Messages

The email notifications contain impact information about:

- the business flows that you are monitoring.
- out-of-sequence events.

  Out-of-sequence events are those events that you have chosen to monitor using the `Check sequence` option, which is an option on the HPBPI Modeler.
- SLA/SLO violations.
- metric threshold violations.

There is a default template for these email notifications; however, you can change the layout of the email messages that are sent by the Notification Server. You can also configure the templates for the email notifications and add properties from the business alerts into the messages and configure how many email notifications that you receive for a particular metric threshold violation.

The default templates are provided as part of the HPBPI installation and instructions for creating your own templates and configuring metric thresholds are provided in the *HPBPI System Administration Guide*.

The following are the template types that you can use to change the format of your messages, if required:

- Velocity, which is a Java-based template Engine.
- XSLT, which transforms one XML document into another text document.

It does not matter which tool you use, it is entirely dependent on your own preferences.

HP Operations Manager Messages

HP Operations Manager enables you to configure the messages that HP BTO software components can receive, and also enables you to modify the format of the messages. Refer to the HP Operations Manager documentation for details of modifying the format of messages within HP Operations Manager and also filtering messages that HP Operations Manager components can receive.

By sending HPBPI notification alerts to HP Operations Manager, you enable HP BTO software users to receive these alerts in the clients of their choice. For example, HPBPI alerts can be displayed as Incidents on the Service Desk client GUI. Note that this assumes that Service Desk integration is enabled within HP Operations Manager.

Refer to the HP Operations Manager documentation to find out how to configure HP Operation Manager to send the HPOM messages that are sent from HPBPI to an HP Operations Manager management client. For example, the *Service Desk Operations Integration Administrator's Guide* provides more information about configuring HP Service Desk and HP Operations Manager to provide this integration.

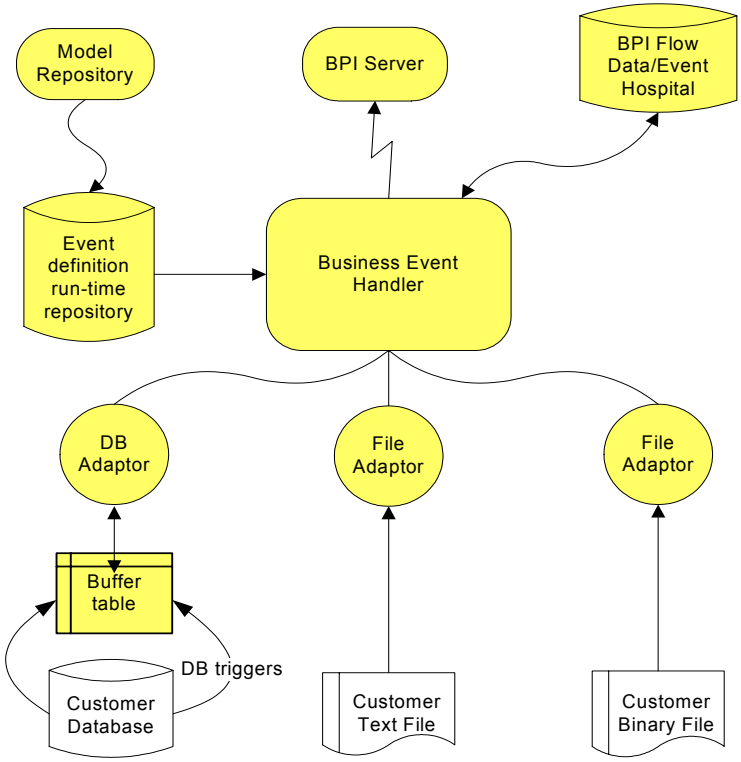
Business Event Handler

The Business Event Handler provides the Business Impact Engine with normalized business events. Normalized events are events that have a standard (known) set of data attributes that enable it to be recognized by the Business Impact Engine. These events indicate the status changes in the underlying business applications.

Business events provide the data required to progress the flow and also assess the business impact of an operational failure; for example, the value of order or details of a new order that has been created.

The Business Event Handler is built using the openadaptor adaptor framework. openadaptor is an open source adaptor integration toolkit, written in Java, and provides a framework for adaptors. It is a message-based integration toolkit, based on the concept of adaptors providing one-way connections between one or more origins and one or more destinations, including publish/subscribe connections. It can be used to connect systems to systems or systems to middleware.

Figure 12 Business Event Handler



In the case of HPBPI, openadaptor is the framework that provides the following HPBPI event adaptors:

- File adaptors

  The flat file adaptor enables you to access information in a binary or ASCII file.

- Database adaptors

  The database adaptor allows you to access information in any JDBC-compliant database. The adaptor uses a polling mechanism to access buffer tables created by database triggers.

- Sockets adaptor

  This is a multi-threaded socket adaptor that by default connects to HPBPI using a specific port.

The Business Event Handler accepts business events received through the adaptors and converts these events into a format that can be accepted by the Business Impact Engine. The Business Event Handler also manages the business events that cannot be immediately delivered to the Business Impact Engine; for example, when the Business Impact Engine has stopped for some reason.

The HPBPI Model Repository holds the design-time Event definitions that you create using the HPBPI Modeler. When these Event definitions are deployed, the Business Events Manager uses the definitions to convert the incoming events into the format required, and to make sure that the correct information is added to the event, so it can be used by the Business Impact Engine.

You can add additional data to events that are received from the individual applications. This allows you to monitor one key application and then assemble the complete set of data required for the event from other applications, rather than monitor all the applications waiting for all the data to become available.

If the Business Impact Engine cannot receive an event from the Business Event Handler for any reason, the Business Event Handler rolls back the event transaction so that it can be retried at a later time. Refer to the *Business Process Insight Integration Training Guide - Business Events* for more details of event adaptors.

In specific cases, the Business Event Handler places a business event in the Event Hospital. It does this if it receives out-of-sequence events or events that contain errors; an event error might be an event that had missing or corrupt data.

Event Propagation

This section gives a brief overview of how events that are received through the Business Event Handler are used to progress Business Flows. The *Business Process Insight Integration Training Guide - Business Events* provides more detail about the Business Event Handler and how to configure it to receive specific business events.

Each time an underlying business system is updated, the Business Event Handler is notified of the change, through a business event, and adds the event details onto the event as defined in the HPBPI Modeler.

The Business Event Handler includes an openadaptor-to-HPBPI adaptor (BIA EngineAdaptor), and all business events for HPBPI are sent to this adaptor. How the adaptor detects the change, depends on how it is configured to communicate with the underlying systems. The adaptor might receive events from a message bus, or it could be notified through a database trigger; see the *Business Process Insight Integration Training Guide - Business Events* for more information about building and configuring the file and database adaptors.

Where the event mapping is successful, the HPBPI event details are updated and the event is then sent to the Business Impact Engine. If there is no HPBPI event defined that matches the data from the event, the event is moved to the Event Hospital by the adaptor. The business event then must be manually discharged from the Event Hospital so it can be resubmitted to the Business Impact Engine. In the case of an out-of-sequence event, the Business Event Handler places the event in the Event Hospital and marks it for automatic discharge. For other cases where the event cannot be delivered to the Business Impact Engine, the Business Event Handler rolls back the event transaction and returns the event to the source adaptor.

When the Business Impact Engine receives a business event, it determines if the event contains details that apply to any of the data instances that currently exist within the Engine. If there is a Data definition subscribed to the Event, and there are associated data instances, the Business Impact Engine updates the data details using the information from the event and the actions specified through the HPBPI Modeler for the appropriate Event and Data definitions. Any changes to the data cause the start and complete conditions for all relevant flows to be re-evaluated, and where appropriate the flows are progressed.

The Business Impact Engine also determines if there is an impact that needs to be reported for the business flows that use this data, for example, those flows with out-of-sequence nodes.

At the same time, the Business Process Dashboard is polling the HPBPI Flow Data and refreshing its display to show the impacts of the changes.

You need to configure adaptors for each data source (application) from which you want to receive business events. These adaptors provides data for the flow, or flows, that you are monitoring. There are different ways that you can create and configure adaptors, which are described in the *Business Process Insight Integration Training Guide - Business Events*. You do not need to understand how to create adaptors at this stage, although you do need to understand them when you come to develop your solution.

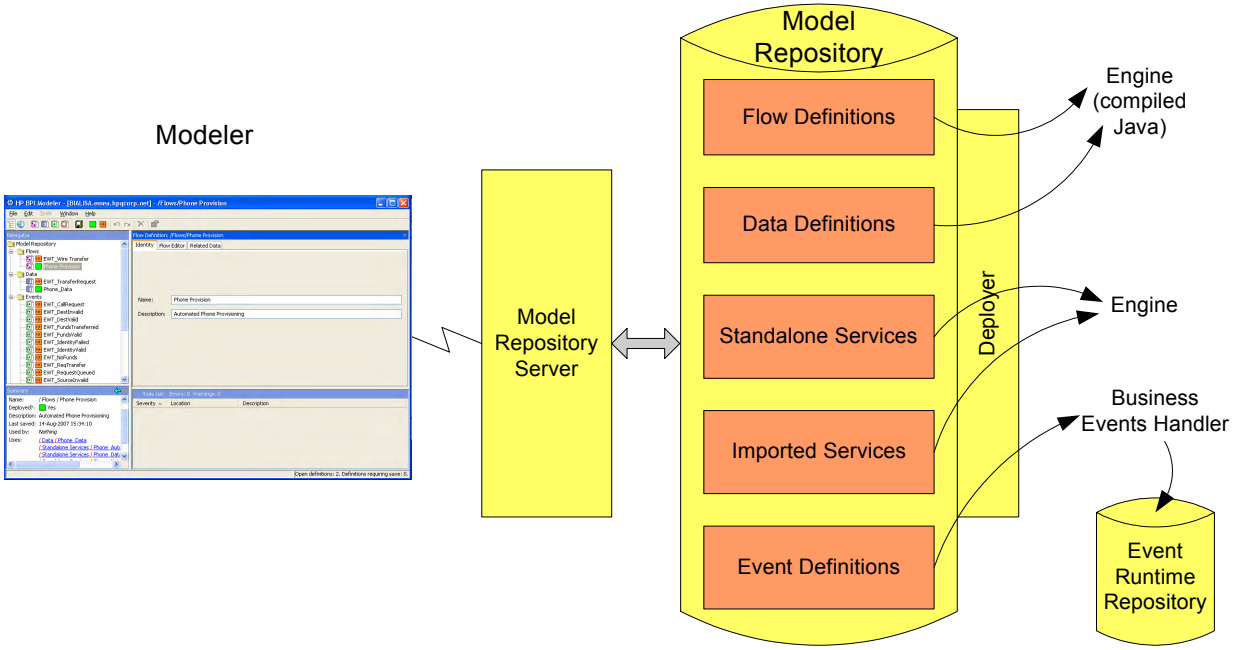
Modeler

The HPBPI Modeler is a client-side HPBPI component that you use to create your business Flow definitions, Data definitions, Service definitions and Event definitions. It is a graphical tool, with a unified interface for all the definition types that you are creating.

You can create and edit business flows quickly and easily using the HPBPI Modeler, and when you are satisfied that the business definitions are complete, you can deploy them to the Business Impact Engine. You can also import Business Process Execution Language (BPEL) into the Modeler and use the BPEL definition as the start point of your Flow definition.

A local repository cache, in the HPBPI Modeler, holds Flow definitions, including Data, Service and Event definitions until they are saved in the Model Repository, which is a set of database tables. The tables are managed by the Repository Server, which is described in section Repository Server on page 42.

Figure 13 HPBPI Modeler



Repository Server

The Model Repository and the Repository Server are the server-based components that manage the data as it is entered through the Modeler. These components store the model definitions in the database and ensure that the details are consistent and deployable. There is a ToDoList associated with each definition, which reports on the consistency of the definition and keeps track of all the outstanding issues and tasks required to ensure that the definition is consistent and complete. Note that a definition cannot be deployed unless it is consistent.

The deployer component of the Repository Server deploys the different definitions to their destinations as follows:

- Flow definitions, including any imported BPEL definitions, are deployed to the Business Impact Engine as compiled Java code.
- Data definitions are deployed to the Business Impact Engine as compiled Java code.
- Event definitions are deployed to the Business Event Handler to be mapped to events from underlying business applications.
- Service definitions are deployed to the Business Impact Engine as compiled Java code.

The Repository Explorer is a Web-based interface that you can use to view and manage the data stored in the Model Repository.

The HPBPI Modeler and its features are described in more detail in the *Business Process Insight Integration Training Guide - Modeling Flows*.

Administration Consoles and Interfaces

The HPBPI components have the following management interfaces that are used for administration and for configuration management:

- HPBPI Administration Console, which is an application that you use to start, stop and configure HPBPI Server Components. Using the HPBPI Administration Console is described in the *HPBPI System Administration Guide*.
- Custom probe dialogs for OVIS. These are Windows interfaces used to configure the HPBPI probes on the OVIS system. Configuring the HPBPI custom probes within OVIS is described in the *HPBPI System Administration Guide*.
- HPBPI Notification Server Administration, which is a Web-based interface (served by Tomcat), to subscribe to the particular events that you want to receive. These events are then sent as email messages, as HP Operations Manager messages, or can invoke scripts. The Notification Server Administration Console and how to use it is described in the *HPBPI System Administration Guide*.
- Intervention Client, which is a Web-based interface that enables you to access, or intervene in, business flow instances. Using the Intervention Client is described in the *HPBPI System Administration Guide*.

Later chapters in this document describe these interfaces, and how to use them, in more detail.

HPBPI Security

You can choose to implement access control for your HPBPI interfaces using HP Select Access or Tomcat (Servlet Engine). When first installed, some of the Web-based interfaces are already configured to use the Tomcat Realm mechanism of authentication; this and how to extend the Tomcat authentication mechanism to all of the HPBPI Web-based interfaces is described in the *Business Process Insight System Administration Guide*.

HP Select Access is part of the HP Identity Management suite of products and enables you to centralize and automate access control for your HPBPI (and other HP software) components. Select Access is based on a policy-based authentication and authorization mechanism for Web-based applications; for example, for the Repository Browser. There is also customized access control provided for the Modeler using Select Access.

For details of how to configure integration with Select Access, refer to the *HP Business Process Insight Administration Guide*.

Where to go Next

You now have an overview of the HPBPI system architecture.

Chapter 3, Integrating with HPBPI describes how HPBPI integrates with other HP and non-HP products and the remainder of this guide provides information that you need to configure these integrations.

Read the chapter appropriate to the task that you are trying to complete.

3 Integrating with HPBPI

This chapter provides an overview of the possible integration points for your HPBPI system.

HPBPI can integrate with an number of HP BTO software and third-party products. Details of how to configure these integrations is provided in other sections of the HPBPI manuals; this chapter provides an overview of the integration points and not details of how to configure the integration.

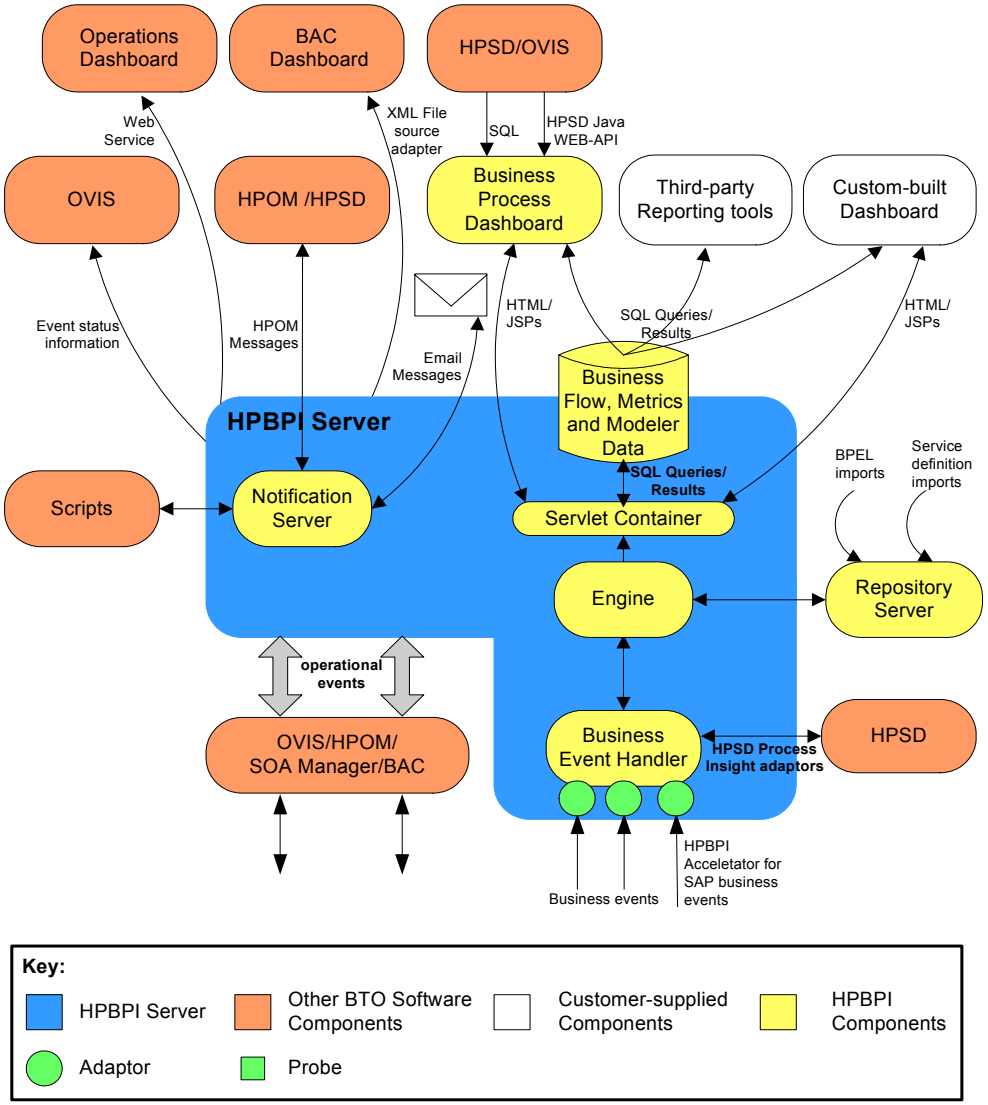
The chapter covers the following topics:

- HPBPI Integration Points on page 46
- HPBPI Adapters for HP BTO Software Integration on page 50
- HPBPI Accelerator for SAP Applications on page 58
- iWay Integration on page 59
- HP Operations Dashboard on page 60
- Self-Healing Services on page 62

HPBPI Integration Points

Figure 14 shows a high-level diagram of how components can integrate with HPBPI; these include both design-time and run-time integrations.

Figure 14 Integrating with HPBPI



There are a number of different ways that you can integrate with HPBPI:

1. Import service definitions from other HP BTO Software components.

   HPBPI has adapters that you can use to integrate with other HP BTO Software products. You can then link to the operational services reported through these HP BTO Software products from your HPBPI business flows and report on the status of these services and how the services are impacting your business flows. Section HPBPI Adapters for HP BTO Software Integration on page 50 describes these adapters and the HP BTO Software products that they enable you to integrate with.

2. Configure data sources through the Business Event Handler and import business events from underlying business applications.

   Using the Business Event Handler, and openadaptor technology, you can import business events and use these events to progress your business flows. You can also use these business events to report the business impact where flows are not progressing as they should be. The Business Event Handler is described in Chapter 2, HPBPI Architecture.

3. Configuring notifications through the Notification Server:

   — You can configure email notifications that can be sent when an alert notification event is received by the Notification Server.

   — You can create scripts that are executed when an alert notification event is received by the Notification Server.

   This enables you to integrate the date from alert notifications that are generated by HPBPI into your own spreadsheet applications. Alternatively, you can create a script that generates an SMS notification message.

   — You can configure HPOM notifications (HPOM messages) that can be sent when an alert notification is received by the Notification Server.

   This enables you to send an HPOM message into HP Operations Manager when a specific event occurs within your business flow. This HPOM message can then be tracked or identified by other HPOM applications that understand the HPOM message format.

   Configuring the Notification Server is described in the *Business Process Insight Administration Guide*.

4. Using reporting tool, such as Crystal Reports and Microsoft Excel, to report on data in the HPBPI Database.

   You can use the data created by HPBPI in the database to report on your business flows and analyze the behavior of your flows as required. The HPBPI database tables are described in Appendix A, Database Schemas. There is also a contributed document on the distribution media that provides an example of using Crystal Reports to analyze the HPBPI data.

5. Customizing the HPBPI Dashboard and create a new Dashboard for your business flows.

   You can make changes to the HPBPI Dashboard to provide a more closely integrated Dashboard solution for your business flows, you might also want to integrate the data collected by HPBPI with existing customer data from another data source. Refer to the *Business Process Insight Integration Training Guide - Customizing the Business Process Dashboard* for more details of how you can make changes to your HPBPI Dashboard.

6. Using the HPBPI Dashboard to provide links through to appropriate HP Service Desk (HPSD) Service Calls and Incident reports for specific flow instances. This is described in detail in Chapter 3, Integrating with HPBPI.

7. Configuring the HPBPI custom probes for OVIS to sample functions of the HPBPI business flows for monitoring purposes.

   HPBPI provides a number of OVIS probes that you can configure to monitor HPBPI business flows, and also report on SLO and SLA violations. This is described in detail in Chapter 6, HPBPI and OVIS.

8. Configure HPBPI and BAC to enable BAC to report on HPBPI Business Flows, Business Process Metrics and Metric Thresholds within the BAC Dashboard and My BAC portlet. This is described in detail in Chapter 4, HPBPI and HP Business Availability Center.

9. Import BPEL definitions into the HPBPI Model Repository using the HPBPI Modeler.

   When you have imported the BPEL definitions, you can use them as the basis, or starting point for your flow definitions. Refer to the *Business Process Insight Integration Training Guide - Importing BPEL*, for more information on the options for importing BPEL definitions.

10. Using any of the predefined flows and adapters, which are specifically designed to be used to monitor HPSD processes for the following HPSD modules:

    — HPSD Helpdesk Manager

    — HPSD Change Manager

    The predefined flows, adapters and an example Dashboard are described in the *Business Process Insight Integration Training Guide - Monitoring Service Desk*.

11. Configure HPBPI to enable the HPBPI Dashboard pages to be navigated by the HP Operations Manager Dashboard; see section HP Operations Dashboard on page 60.

HPBPI Adapters for HP BTO Software Integration

HPBPI provides a number of adapters that provide integration points to other HP BTO Software products. These adapters are part of the HPBPI licensed product.

The main function of these adapters is to enable you to use HPBPI to monitor operational events from these HP BTO Software products. Specifically, these are the events that are exposed through the HPBPI adapter interfaces.

The following are HPBPI adapters that are currently available:

- Service Source adapters, including:
  - Business Availability Center Service Source; see section Service Source for Business Availability Center on page 51.
  - HP Service Oriented Architecture Manager (SOA Manager) adapter; see section HPBPI SOA Manager Adapter on page 52.
- HP OpenView Internet (OVIS) adapter; see section HPBPI and OVIS Probes and Alarms on page 53.
- HP Operations Manager (HPOM) adapter; see section HPBPI Operations Manager Adapter on page 55.
- HP Service Desk adapter; see section HP Service Desk Adapters on page 56

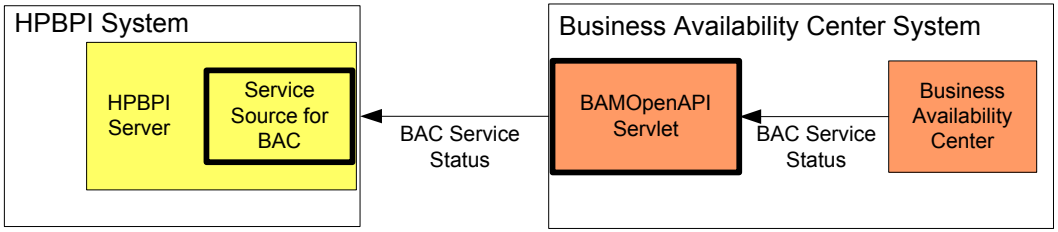
As the model for creating adapters based on Web Services within HPBPI is extensible, the list of adapters using this model will increase in future versions of the product.

Service Source for Business Availability Center

Business Availability Center is one of the HP BTO Software products that you can integrate with. You create an Operational Service Source for each Business Availability Center implementation that you want to integrate with. You can then configure HPBPI to link Business Availability Center Service status events to the business flows that you define.

You use Business Availability Center to model the Service definitions, and the HPBPI Adapter for Business Availability Center (created as a result of defining an Operational Service Source) is responsible for retrieving the status of the Services defined and modeled within Business Availability Center.

Figure 15 HPBPI Service Source for Business Availability Center



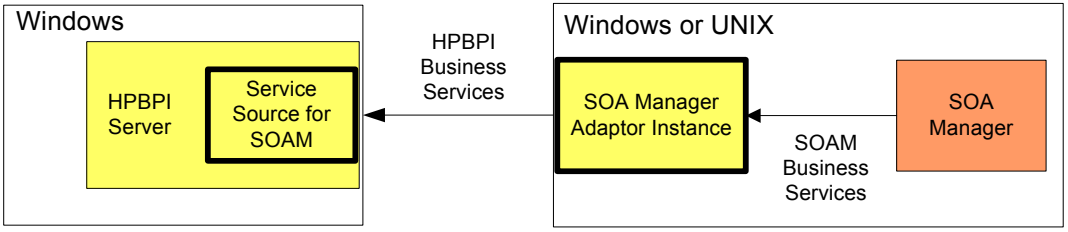
Details of you can integrate HPBPI and Business Availability Center can be found in Chapter 4, HPBPI and HP Business Availability Center.

HPBPI SOA Manager Adapter

SOA Manager is one of the HP BTO Software products that you can integrate with. You can configure instances of the HPBPI SOA Manager adapter to integrate with your SOA Manager implementations. You can then configure HPBPI to link SOA Manager business events to the business flows that you define.

HP SOA Manager enables you to manager your service oriented architecture (SOA) resources to ensure their reliability and optimize their performance. The combination of HP SOA Manager and HPBPI provides you with the ability to monitor the health and performance of services running within a Service Oriented Architecture.

Figure 16 HPBPI SOA Manager Adapter



Details of installing the adapter can be found in the *Business Process Insight Installation Guide* and more details of how the adapter integrates with HPBPI can be found in Chapter 5, HPBPI and SOA Manager.

HPBPI and OVIS Probes and Alarms

OVIS is one of the HP BTO Software products that HPBPI can integrate with in order to provide HPBPI users with information on the Internet-related services that OVIS manages. Specifically, the OVIS integration can be used to predict, isolate, diagnose and troubleshoot problems on the services that HPBPI is configured to monitor.

When you have configured a connection from HPBPI to OVIS, the HPBPI Modeler is able to recognize OVIS Services and import these services into the Model Repository. You can then associate these operational services with nodes in your business flows.

HPBPI integrates with OVIS in the following ways:

1. HPBPI polls OVIS services for alarms and violations.

   OVIS measures the availability, response time, setup time, and throughput of specific Internet and network activity. Using the data it receives, OVIS can generate alarms and make them available to HPBPI, and other HP BTO Software products. These alerts and regular information updates keep you informed as to whether or not your Internet and network services are performing efficiently.

   OVIS also generates SLO and SLA violation events, which can be received by HPBPI and sent to the Notification Server, which in turn sends them on as email alerts to anyone who has subscribed to them.

2. OVIS can be configured to probe HPBPI using HPBPI custom probes, which you can install on the OVIS system.

   The Internet Services Manager component of OVIS is responsible for managing data collected by OVIS probes; these probes provide the data that OVIS uses.

There are three custom probes provided with HPBPI and these probes are used specifically to obtain HPBPI data relating to HPBPI flows and flow instances. The custom probes are installed on the same system as OVIS and are used to calculate:

— the time taken to progress between two defined nodes in the flow, during the configured time period.

— the number of flow instances that are currently active at a specified node, the value of the weight parameter for these flow instances and the throughput rate per hour. As an example, the probe might calculate the number of outstanding claims and the value of these outstanding claims at a particular time.

— the number of flow instances that have been completed throughout the probe period. This provides throughput information for the probe period. The probe also calculates the value of the weight parameter for the flow instances. As an example, the probe might calculate the number of claims that have been completed throughout the probe period and the value of these completed claims. Values returned are presented as an hourly rate.

In addition to the HPBPI custom probes, you can also use OVIS to:

- report on the operational status of your HPBPI system
- define HPBPI-specific SLO and SLAs and report on violations

OVIS probes that report on operational status are created within OVIS as described in the OVIS documentation.

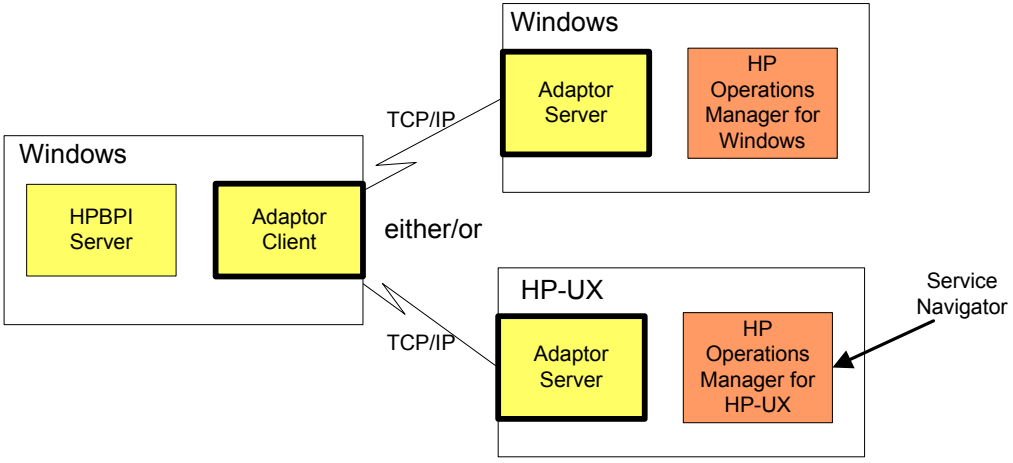
HPBPI Operations Manager Adapter

HP Operations Manager (HPOM) is one of the HP BTO software products that HPBPI can integrate with. HPOM can provide status information to HPBPI about the operational services that HPOM has been configured to monitor.

The HPBPI Operations Manager Adapter is responsible for:

- passing HPOM service definitions from either HPOM for HP-UX or HPOM for Windows to the HPBPI repository. These services can then be made available through the HPBPI Modeler.
- receiving events from either HPOM for HP-UX or HPOM for Windows. These events are related to status changes in the services that they are monitoring. HPOM then passes these events on to the subscribing Flow definitions as HPBPI events.

Figure 17 HPBPI Operations Manager Adapter



An HPBPI Server can be connected to one HPOM Server, which can be HPOM for HP-UX or HPOM for Windows, it cannot be connected more than one HPOM Server at the same time. You configure the adapter connection after you have installed HPBPI using the HPBPI Administration Console.

HPBPI service definitions that you enter using the HPBPI Modeler are linked to service definitions made available from HP Operations Manager, through the HPBPI Operations Manager Adapter.

There are two components that make up the adapter:

- The Adapter Client component (referenced as the Operations Manager Receiver in Figure 17 on page 55), which is installed with the HPBPI Server.
- The Adapter Server component, which is installed as a separate HPBPI component on the system where HP Operations Manager is installed; this is either Windows or HP-UX.

Details of how to install the HP Operations Manager Adapter for HPBPI are provided in the *Business Process Insight Installation Guide*.

HP Service Desk Adapters

HP Service Desk (HPSD) is one of the HP BTO software products that HPBPI can integrate with. HPBPI provides a number of HPSD adapters that you can use to monitor HPSD ITIL processes.

There is one adapter provided for each of the following ITIL processes:

- Service Calls
- Incidents
- Problems
- Changes
- Work Orders

These adapters are part of HPBPI and can be used with an HPBPI license. Alternatively, if your implementation contains only HP Service Desk you can use the HP Service Desk Process Insight license bundle for HPBPI. This license bundle is provided specifically for HPSD integration with HPBPI.

Using the HP Service Desk Process Insight adapters, plus the flow definitions and the customized Dashboard, you can monitor the status of HPSD Change Management and Help Desk activities.

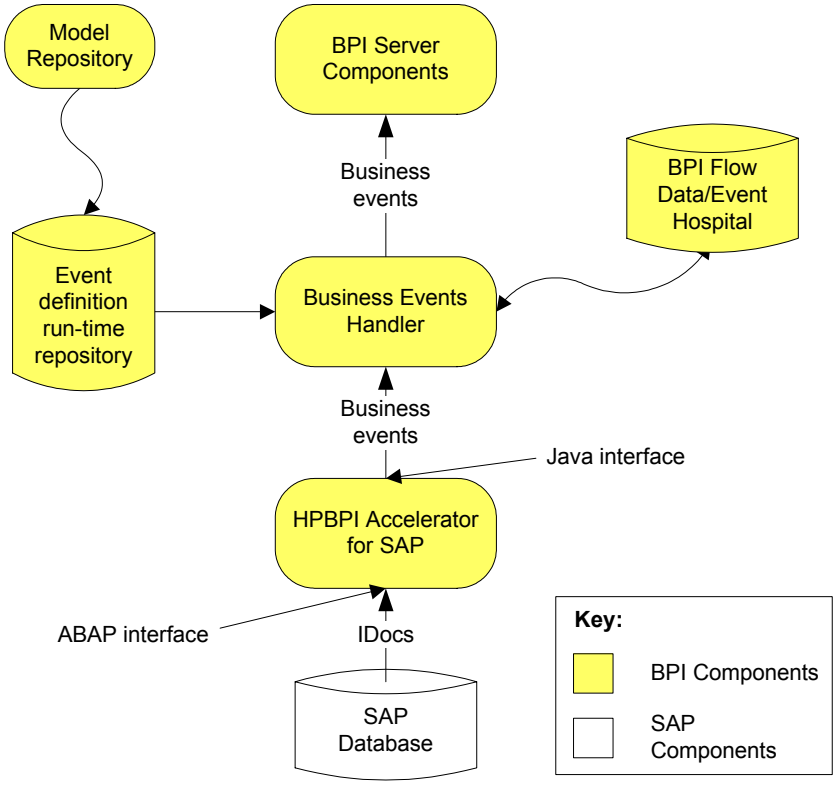
In addition to these adapters, you can configure the HPBPI Dashboard to link through to HPSD Service Calls and Incident reports. These are reports that relate to the impacted instances currently being viewed through the Business Process Dashboard.

HPBPI Accelerator for SAP Applications

In addition to the adaptors for HP BTO Software products, HPBPI provides an Accelerator for SAP applications. The HPBPI Accelerator for SAP applications is specifically designed to integrate with ALE IDocs within SAP; its design is based on the openadaptor framework. The Accelerator uses the Business Event Handler to access the status information within an IDoc header and is a separately licensed product.

Figure 18 shows the integration between the HPBPI Accelerator and SAP.

Figure 18 HPBPI and SAP Integration Using HPBPI Accelerator



The *Business Process Insight Accelerator for SAP Guide* contains details of installing and setting up the Accelerator.

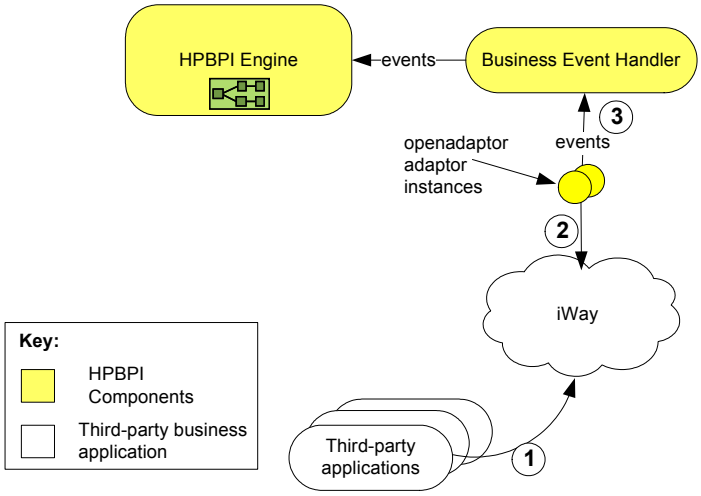
iWay Integration

The iWay Adaptive Framework (iWay) is similar to openadaptor. Both products address the same requirement, which is to enable developers to quickly and easily integrate components to enable data to flow between heterogeneous data sources.

Both iWay and openadaptor provide the ability to read and write to databases and files; however, iWay provide a larger selection of adapters, particularly for integrating well-known business applications, such as SAP, PeopleSoft and JD Edwards.

The HPBPI integration with iWay is achieved through an openadaptor adapter for iWay, as shown in Figure 19.

Figure 19 HPBPI and iWay Integration



From this diagram:

1. Business events are received from third-party applications by iWay.
2. iWay passes the events to openadaptor.
3. openadaptor then passes the events into HPBPI through the Business Event Handler, where they are processed by the Business Impact Engine.

For more information on the iWay integration with HPBPI refer to the *Business Process Insight Integration Training Guide - Business Events*.

HP Operations Dashboard

The HP Operations Dashboard (HPOD) is a BTO-wide Dashboard and should not be confused with the HPBPI Business Process Dashboard, which is for use only with HPBPI.

Using the HP Operations Dashboard you can quickly and easily create management dashboards that address differing requirements; for example, individual dashboards for service managers, business managers, application managers or operations staff. The content and scope of each Dashboard that you create can be tailored for each user or category of user.

As part of these solutions, you can link the HPBPI service impact information into the HP Operations Dashboard; refer to the HP Operations Dashboard documentation for more details of the features and functions that it offers and how to add links to the HPBPI Business Process Dashboard.

HPBPI publishes a Web service on its Web Services Provider port. This Web service provides the HP Operations Dashboard with the ability to import service definitions based on deployed HPBPI flows, metrics and thresholds; this information can then be used for building HP Operations Dashboard configurations.

The Operations Dashboard also uses the status information relating to the HPBPI service definitions and metrics, as events are being processed by HPBPI, and presents these results through the HP Operations Dashboard.

When configuring the HP Operations Dashboard access to the HPBPI Web service and to the HPBPI event status information you need to have the following information available:

1. The fully qualified host name of the machine where the HPBPI Server is installed.
2. The port number for the HPBPI Web Services Provider.
3. The port number for the HPBPI Servlet Engine component.
4. The string `OVBPI`, which is required by HP Operations Dashboard to identify the connection as an HPBPI connection.

You can find the relevant port numbers using the Administration Console on the Port Numbers page. The HP Operations Dashboard uses this information to build a URL to access the HPBPI Web service and the HPBPI Business Process Dashboard pages.

In addition to this information, you need to:

- Make sure that the HPBPI `Web Services Provider` service is running.

  You start this service from the HPBPI Administration Console. When this service is running, HPBPI exposes its data as a Web Service on the port number used by the Web Services Provider service. On a new installation this is port `44014`. You can check the port number used from the `Port Numbers` page on the Administration Console.

  If the `Web Services Provider` service is not started, the HP Operations Dashboard cannot access any of the HPBPI services, metrics or thresholds.

- Make sure the HP Operations Dashboard is configured to link to HPBPI using the data described earlier in this section. Instructions for configuring HP Operations Dashboard are provided in the HP Operations Dashboard documentation.

When the configuration is complete, HP Operations Dashboard can link directly to the HPBPI business impact data; however, if you have configured the HPBPI Business Process Dashboard to be authenticated by either Select Access or Tomcat (Servlet Engine), this authentication is passed to the HP Operations Dashboard. If you do not want to your HP Operations Dashboard users to have to log on to the HPBPI Business Health pages from within the HP Operations Dashboard, you need to disable the security method that you have configured for the HPBPI Business Process Dashboard; you do this using the HPBPI Administration Console.

In addition, if you have configure HPBPI to integrate with HP Service Desk, this information is not available when linking to the HPBPI data from HP Operations Dashboard. However, you can configure HP Operations Dashboard to access this information directly.

Self-Healing Services

HP Self-Healing Services, as used by HPBPI, automates some of the steps involved in collecting information about a problem and then packaging the information to send it to HP. You can use HPBPI's integration with the Self-Healing Services to gather information about a problem and send this information to HP using a secure Web access mechanism.

Self-Healing Services (SHS) is a free service for HP support customers that automates many of the steps involved in troubleshooting, searching for solutions, and submitting support cases for HP BTO Software applications. SHS results in a significant decrease in the amount of time and effort required for you to recognize that a problem occurred, search for documented solutions, submit a support case, collect troubleshooting information, and get a solution.

The type of information that HPBPI collects includes:

- configuration files
- log files
- system environment configuration

The Self-Healing Service is therefore able to reduce the time and effort required to notify HP about a problem with HPBPI.

The *Business Process Insight Installation Guide* explains how to automatically collect information about a problem using the Self-Healing processes.

The following is an overview of how the Self-Healing Services work:

1. When an error occurs on your system, you can start the Self-Healing Services client and run the HPBPI Data Collector. The Data Collector is specific to HPBPI and provides the Self-Healing Services with the data required to troubleshoot your HPBPI system.
2. The data collected can be securely transmitted to HP support systems using HP's Instant Support Enterprise Edition; this tool is already in use at many customer sites. This enables HP support engineers much faster access to the data they need for solving customer problems. The same data is also available for you to use for troubleshooting if required.
3. The HP support systems analyze your information and publish a system-specific incident report. This report contains information such as:
   - links to potential solution documents
   - patch analysis for HP BTO Software applications
   - configuration analysis.

Details of how to set up your system and download the software that you need to run in order to use the HP Self-Healing Services client and ISEE client, can be found at the following URL:

http://support.openview.hp.com/self_healing_downloads.jsp

There is a `download software` option under the `Self-Healing Services Downloads` heading on this Web page.

4 HPBPI and HP Business Availability Center

This chapter describes how you can integrate HPBPI and HP Business Availability Center. Business Availability Center is a solution for real-time performance and availability monitoring for businesses.

The chapter covers the following topics:

- HPBPI and Business Availability Center Integration on page 66
- Design-Time Integration on page 71
- Run-Time Integration on page 73
- Using Business Availability Center as a Source of Operational Service Status Data on page 76
- Configuring a Business Availability Center Data Sample Destination on page 78
- Creating an XML File Source Adapter on page 80
- Creating a My Business Availability Center Application for HPBPI on page 84
- Description of Data Samples Sent to Business Availability Center on page 86
- Business Availability Center System Configuration on page 88

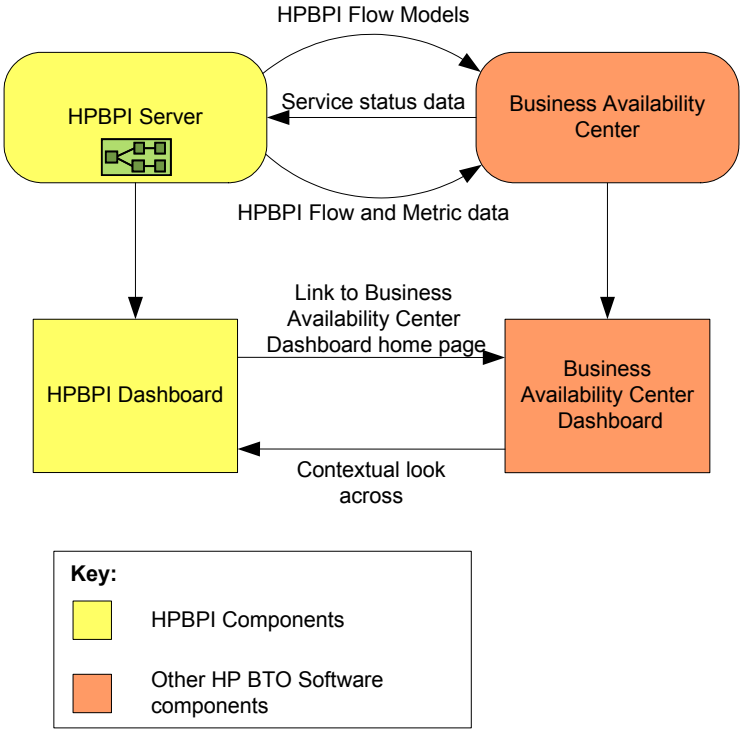
HPBPI and Business Availability Center Integration

HP Business Availability Center enables you to optimize the performance and availability of your business applications and proactively resolve problems when they arise.

Using Business Availability Center, you can model your business's IT processes and your supporting business applications services; you do this by mapping the relationships between your business processes and their supporting infrastructure. In this way, Business Availability Center enables you to optimize your business processes by enabling you to detect problems and proactively monitor the effect that these problems are having on your business. Business Availability Center also provides the tools to help you prioritize problems, based on business impact and service level compliance.

Figure 20 shows the high-level architecture diagram.

Figure 20 HPBPI and Business Availability Center High-Level Architecture



The integration between HPBPI and Business Availability Center enables you to:

- use Business Availability Center to report on the status of operational services and how they impact HPBPI business flows.
- view HPBPI Flow and Metric definitions from within the Business Availability Center Dashboard.
- link from the Business Availability Center Dashboard to the HPBPI Business Process Dashboard to monitor more detail of the business impact for a selected Business flow.
- view HPBPI Flow diagrams and Business Threshold data, in the form of dials and tables, through My BAC.

Business Availability Center and HPBPI Terminology

Within Business Availability Center the HPBPI Flow and Metric definitions are referenced using Business Availability Center terminology as follows:

- A Flow definition is referenced as a Business Process
- A Node (which is a component of a Flow definition) is referenced as a BPI Step
- A Business Process Metric definition is referenced as a BPI Monitor.

  A Monitor is a Business Availability Center data collector that runs profiles and collects performance data, in this case from HPBPI.
- A Metric Threshold is referenced as a key performance indicator (KPI) for a BPI Monitor. A KPI, within Business Availability Center, is a measurement that is calculated for a CI and which can then be compared against defined criteria.
- HPBPI is referred to as an external data source in Business Availability Center terms.

  As an external data source, HPBPI can send data samples to Business Availability Center, where the data samples can be used for Metric Threshold reporting purposes.

Integration Options

HPBPI and Business Availability Center can be integrated in a number of ways and the integration option that you select affects how the HPBPI and Business Availability Center systems can interoperate.

You can:

- Export Flow definitions; see section Exporting HPBPI Business Flow Definitions to Business Availability Center on page 68.
- Export Flow and Metric definitions; see section Exporting HPBPI Business Process Metrics to Business Availability Center on page 69.
- Create a My BAC portlet server; see section Creating a Business Availability Center Portlet Server for HPBPI on page 70.

Exporting HPBPI Business Flow Definitions to Business Availability Center

HPBPI deployed Flow definitions (including Nodes) can be exported to Business Availability Center and represented as CIs within the Business Availability Center Business Processes View.

Superseded Flow definitions are not exported.

Having exported the Flow definitions to Business Availability Center, and configured an Operational Service Source for the HPBPI and Business Availability Center connection, you can map IT operational services to BPI Steps within Business Availability Center.

You can then:

- configure HPBPI to receive status information relating to status of the BPI Steps within Business Availability Center
- monitor the business impact of anomalies in the underlying operational services using the HPBPI Business Process Dashboard.

The HPBPI Business Process Dashboard can also link through to appropriate Business Availability Center service definitions (in the Business Availability Center Dashboard) for impacted instances.

Refer to the following sections for details of how to configure HPBPI to export Business Flow definitions to Business Availability Center:

- Using Business Availability Center as a Source of Operational Service Status Data on page 76
- Creating an XML File Source Adapter on page 80

Exporting HPBPI Business Process Metrics to Business Availability Center

In addition to exporting Business Flows, you can configure HPBPI to send data samples to Business Availability Center. These data samples provide data relating to the Business Process Metrics defined for the Flows.

A data sample, in Business Availability Center terms, is application-specific data sent from an application, in this case, HPBPI. In the case of HPBPI, these data samples can be used by Business Availability Center to show key performance indicators (KPIs), relating to the imported Flow and Metric definitions, on the Business Availability Center Dashboard. The data samples are taken from the values of specific Metric Thresholds in the HPBPI Database. Data samples are sent from HPBPI to all Data Sample Destinations that you configure within the HPBPI Administration Console.

Having exported the Business Metrics to Business Availability Center, you can monitor HPBPI Flows (Business Processes) and Business Metrics (BPI Monitors) from the Business Availability Center Dashboard.

The Business Availability Center Dashboard can also provide contextual links to the details of HPBPI-specific CIs; that is, Flow and Metric definitions, within the HPBPI Business Process Dashboard. Be aware that Metric Threshold values are not shown in this view, but they can be displayed using the My BAC portlets that are specific to HPBPI.

Refer to the following sections for details of how to configure HPBPI to export Business Flow definitions and Business Process Metrics to Business Availability Center:

- Configuring a Business Availability Center Data Sample Destination on page 78

  You must configure the Business Availability Center Data Sample Destination before exporting the Business Metrics. The option to send data samples to Business Availability Center is not enabled until a data sample destination is configured.

- Creating an XML File Source Adapter on page 80

Creating a Business Availability Center Portlet Server for HPBPI

Within Business Availability Center, you can configure a Business Availability Center portlet server (My BAC). My BAC is a portal that you can customize to display specific content that is relevant to you, in this case, HPBPI Business Flow information.

My BAC can provide the following portlet views for HPBPI:

- A display of the Flow diagram for a BPI Business Process that corresponds to the CI selected within Business Availability Center.
- A display of the Metric thresholds for a BPI Business Process that corresponds to the CI selected within Business Availability Center.

You cannot link directly from My BAC to the HPBPI Business Process Dashboard. My BAC calls the Business Process Dashboard as a portlet and because the Business Process Dashboard is not a portlet, it cannot be called directly from My BAC.

The data presented through My BAC is the HPBPI Flow diagram and the Metric Threshold Summary as shown on the summary page for the Business Flow in the Business Process Dashboard.

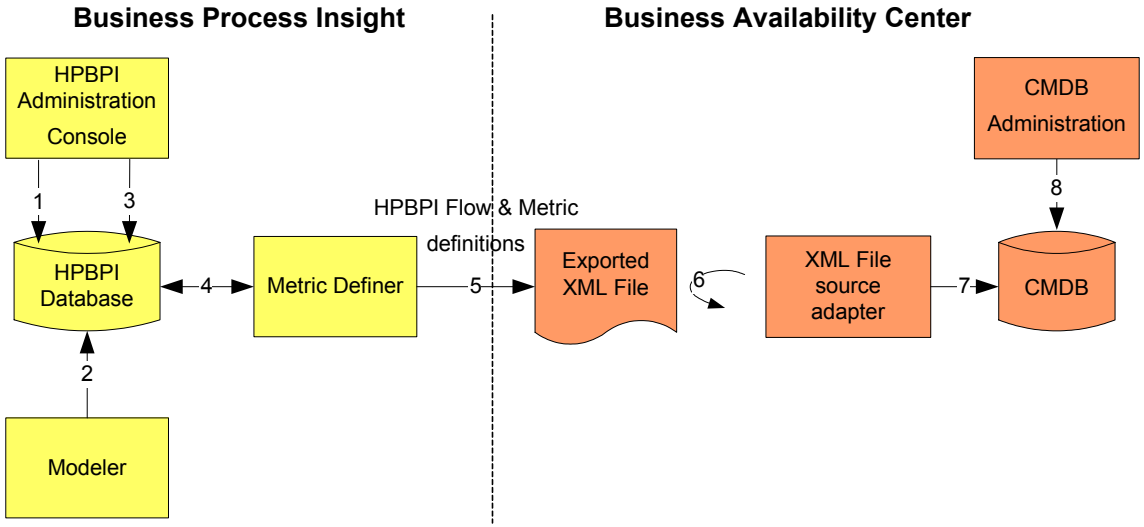
Refer to section Creating a My Business Availability Center Application for HPBPI on page 84 for details of configuring a Business Availability Center portlet server for HPBPI.

Design-Time Integration

This section describes how data is moved between HPBPI and Business Availability Center when you are designing your flows and configuring the HPBPI and Business Availability Center systems.

You do not need to read this section, or the next section, which describes run-time integration, to complete the configuration.

Figure 21 HPBPI and Business Availability Center Design-Time Configuration



The flow of configuration information, as shown in Figure 21, is described in the following steps:

Step 1: Using the HPBPI Administration Console, configure an Operational Service Source connection to the Business Availability Center Server that you intend to integrate with.

Step 2: Create the relevant Business Flow definitions using the HPBPI Modeler and deploy the Flow. The deployed Flow definition is then represented in the HPBPI database tables.

Step 3: Configure a Business Availability Center Data Source Destination using the Administration Console. You need only complete this step if you want HPBPI Flow and Metric data to be viewed from with the Business Availability Center Dashboard or through My BAC.

Step 4: The Metric Definer reads the data from the HPBPI database required to build an XML file. This XML file is subsequently imported into Business Availability Center as an XML File source adapter.

The level of detail included in the export options has an impact on what can be viewed using Business Availability Center in the run-time operation.

If you select to export Business Metric definitions in addition to Flow definitions, the Metric Definer maps Metric definitions to BPI Monitors and maps Business Thresholds to KPIs when creating the XML file.

If you do not select to include Business Metrics, the Metric Definer builds an XML file using only Flow definitions.

Step 5: The XML file generated by the Metric Definer is manually imported into Business Availability Center using Source Manager from within the CMDB Administration. The HPBPI files containing the information required by Business Availability Center are referenced in the XML File Source Adapter.

Step 6: Business Availability Center synchronizes the HPBPI file referenced in Step 5 at intervals that you configure.

Step 7: The XML File Source adapter generates CIs, which are added to the Business Availability Center CMDB.

If you redeploy a Flow definition that has been exported, you must export the Flow definition again and import it into Business Availability Center. You must also include any required Business Metric definitions.

Step 8: You map the IT operation entities within the Business Availability Center environment to the BPI Steps imported from the HPBPI.

If you subsequently add new steps to your HPBPI Business Flow, you need to repeat this step in addition to re-importing the Flow Definition.

More detail relating to these steps and the configuration required is provided in later sections of this chapter.

Run-Time Integration

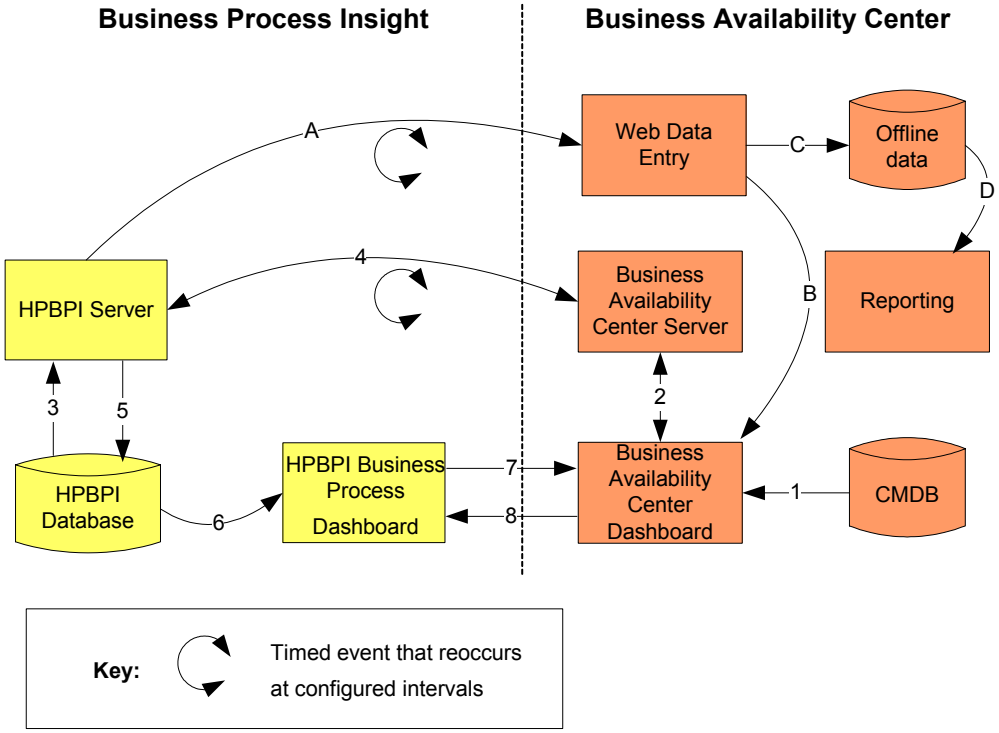
This section describes the behavior of the HPBPI and the Business Availability Center system following the initial design of the systems, as described in Design-Time Integration on page 71.

When you have configured HPBPI and Business Availability Center to operate as you want them to, the behavior of the two systems is divided into two logical areas of operation:

- HPBPI requests ongoing updates of the status of BPI Steps; see section Request for Updates of BPI Steps on page 74.
- HPBPI sends data samples to Business Availability Center; see section HPBPI and Business Availability Center Data Sampling on page 75.

These logical areas are shown in Figure 22.

Figure 22 HPBPI and Business Availability Center Run-Time Behavior



Request for Updates of BPI Steps

The following steps describe the behavior of the HPBPI integration with Business Availability Center when HPBPI requests updates for the status of IT operational resources within Business Availability Center. This behavior occurs only when you configure the Business Availability Center system as an Operation Service Source within the HPBPI Administration Console and export the appropriate Flow definitions.

The steps in the list relate to the numbers in Figure 22:

Step 1: The Business Availability Center Dashboard issues a query to the Business Availability Center CMDB to determine the status of the BPI Step CIs.

Step 2: The results of this query are displayed in the Business Availability Center Dashboard and passed to the Business Availability Center Server.

Step 3: The Operational Service Source configuration for the Business Availability Center connection is read from the HPBPI Server.

Step 4: The configuration for the Business Availability Center connection is used to query the Business Availability Center Server at regular intervals to obtain the updates to the Service status details held for BPI Steps.

Step 5: The Service status details relating to the BPI Steps are written to the HPBPI database tables in order that they can be made available to other HPBPI components.

Step 6: HPBPI Business Process Dashboard takes its data from the HPBPI database and presents the impact information on the Dashboard.

Step 7: The HPBPI Dashboard links to the Business Availability Center system to obtain more details on the status of Services as modeled in the Business Availability Center environment.

Step 8: The Business Availability Center Dashboard links to the HPBPI Business Process Dashboard to obtain more details of the HPBPI-specific CIs that you have exported to the Business Availability Center environment.

More detail relating to these steps and the configuration required is provided in later sections of this chapter.

HPBPI and Business Availability Center Data Sampling

The following list describes the behavior of the HPBPI integration with Business Availability Center when HPBPI sends data samples to Business Availability Center to be mapped onto BPI Monitors and KPIs.

This behavior occurs only when you configure Business Availability Center Data Sample Destinations within the HPBPI Administration Console, and when you export Business Metrics and Metric Thresholds to Business Availability Center.

The steps in the list relate to the alpha characters in Figure 22.

Step A: Using the configuration information for Business Availability Center Data Sample Destinations, the HPBPI Server periodically sends a set of data samples to the Business Availability Center Web Data Entry (WDE) component. The data samples are held on the HPBPI Server for a configurable length of time, until the WDE component in Business Availability Center acknowledges receipt. If a data sample cannot be delivered in the required time, it is deleted. The values of these data samples are KPIs within the Business Availability Center environment.

Step B: The data samples, based on the data exported from HPBPI, are processed by the Business Availability Center system and the status of the related Business Processes, BPI Monitors and BPI Steps are displayed on the Business Availability Center Dashboard.

Step C: The data samples received by Business Availability Center are also written to an offline database where they can be used for reporting purposes.

Step D: The data samples in the offline database are made available for reporting through My BAC. It is within My BAC that the KPIs can be displayed.

Using Business Availability Center as a Source of Operational Service Status Data

In order for HPBPI to receive status information, which relates to operational services that have been configured for BPI Step CIs within Business Availability Center, you need to complete the following within HPBPI and Business Availability Center:

- Configure an Operational Service Source for the Business Availability Center system you are integrating with; see section Configuring Operational Service Sources on page 76.
- Export the required Flow definitions from HPBPI and import them into Business Availability Center; see section Creating an XML File Source Adapter on page 80.

If you do not want to receive operational service status information from Business Availability Center, you can skip this section and continue at the section Configuring a Business Availability Center Data Sample Destination on page 78.

Configuring Operational Service Sources

In order for HPBPI to receive status information relating to the Business Availability Center IT operational resources that are linked to BPI Steps, you need to configure an Operational Service Source for the appropriate Business Availability Center system within HPBPI.

You do this using the HPBPI Administration Console as follows:

1. Start the Administration Console as follows:

   `Start | Programs | HP | HP Business Process Insight | Administration`

2. Click the `Operational Service Sources` option in the left-hand navigation pane.

   The Operational Service Source configuration page is opened in the right-hand pane.

3. Click `Add`, to add a new service source.

   You are presented with the `Add Operational Service Source` dialog.

4. Select `HP Business Availability Center Adapter` to add a Business Availability Center operational service source.

   You are presented with the `Business Availability Center Source Properties` dialog.

5. Enter values for the properties of the Business Availability Service Source. The properties are fully described in the *Business Process Insight Administration Guide*:

   — `Service Source Name`

   — `Description`

   — `BAC Centers Server Hostname`

   — `BAC Centers Server servlet engine port`

   — `BAC Centers Server HTTP port`

   — `BAC View name`

   — `User name`

   — `Password`

   — `Status event poll interval (seconds)`

   — `Connection timeout (seconds)`

   — `Override Service Adapters' log level?`

   — `Log level for this adapter`

6. Click the `OK` button when you modifications are complete.

7. Make sure that the `Enabled` check box is selected in the column next to the new service source entry.

8. Click the `Apply` button to apply your changes

9. Move to the `Component Status` screen and stop and restart all the HPBPI components.

In order to complete the configuration, you need to export the appropriate HPBPI Business Flows and import them into BAC as described in section Creating an XML File Source Adapter on page 80.

Configuring a Business Availability Center Data Sample Destination

You need to configure a Business Availability Center Data Sample Destination in order for HPBPI to send data samples to Business Availability Center. This enables you to view HPBPI Business Process Metrics from within the Business Availability Center Dashboard.

If you are using the Business Availability Center integration solely for a source of IT operational resources, you do not need to configure a data sample destination. In this case, you can skip this section and continue at section Creating an XML File Source Adapter on page 80.

Both the Data Sample Service Provider and the Data Sample Destination are configured through the HPBPI Administration Console as follows:

1. Select `BAC Data Sample Destinations` from the Navigation tree in the Administration Console.

   The right-hand pane shows a list of Data Sample Destinations.

2. From the right-hand pane, select the `Add` button to add a new Destination for the Business Availability Center system to which you want to send data samples.

   The Business Availability Center Data Sample Destination Properties page is opened in the right-hand pane.

3. Enter values for the properties of the `BAC Data Sample Destination`. The properties are fully described in the *Business Process Insight Administration Guide*:

   - `Destination name`
   - `Description` (Optional)
   - `BAC Core Server hostname`
   - `BAC Core Server HTTP port`
   - `Data samples send interval (seconds)`
   - `Connection timeout (seconds)`

- Guaranteed retention period (minutes)
- Override Data Samples Providers' log level?
- Log level for this destination

4. Click the OK button when you modifications are complete.
5. Make sure that the Enabled check box is selected in the column next to the new service source entry.
6. If you have one, you can configure a Web Proxy for your Data Samples Destinations.
7. Click the Apply button to apply your changes.
8. Move to the Component Status screen.
9. Stop and restart all the HPBPI components. In particular, make sure that you start the BAC Data Samples Service Provider.

When you have configured your Data Sample Destination, you can continue and export the appropriate HPBPI Business Flows and import them into Business Availability Center as described in section Creating an XML File Source Adapter on page 80.

Creating an XML File Source Adapter

This section describes how to export the required HPBPI Business Flow data into an XML file, which can then be imported into Business Availability Center as an XML File source adapter.

When you create the XML File source adapter in Business Availability Center, the HPBPI data is available within the Business Availability Center Dashboard to be viewed as Business Processes and BPI Monitors.

The HPBPI data that you choose to export is related to the requirements that you have for making HPBPI data available within Business Availability Center.

- If all you need is to be able to link Business Availability Center IT operational resources (services in HPBPI terminology), you need only export your Flow definitions to Business Availability Center. You also need to configure an Operational Service Source for the Business Availability Center system as described in section Using Business Availability Center as a Source of Operational Service Status Data on page 76.

  You can then use the Business Availability Center tools to map Business Availability Center IT operational resources to the nodes within the HPBPI Business Flow.

  This also enables you to link from the HPBPI Dashboard to the Business Availability Center Dashboard for more detail relating to the Business Availability Center IT operational resources.

- If you want to be able to monitor the status of HPBPI Business Flows, and associated Business Metrics using the Business Availability Center Dashboard, you need to export Flow and Metric definitions. You also need to configure a `BAC Data Sample Destination`, as described in section Configuring a Business Availability Center Data Sample Destination on page 78. The data sampling is used to present HPBPI-specific KPIs within the Business Availability Center Dashboard; these KPIs are described in section HPBPI and Business Availability Center Data Sampling on page 75.

  Exporting Business Flow and associated Business Metrics also enables the Business Availability Center Dashboard to link to the Flow-related information in the HPBPI Dashboard.

- If you want to create a My BAC application within Business Availability Center to provide a real-time view onto the HPBPI Business Process Dashboard Flow summary page, use the Business Availability Center Dashboard interface; refer to section Creating a My Business Availability Center Application for HPBPI on page 84.

When you have imported the required HPBPI data, it can be modeled within the Business Availability Center Dashboard in the same way as any other application that Business Availability Center integrates with.

Using the Metric Definer to Create an Entities File

If you want to have the Business Availability Center Dashboard, or a My BAC page, report on HPBPI Business Processes, you need to import an XML entities file into Business Availability Center with the required XML definitions of the HPBPI entities.

You export the required information from the Metric Definer as a .zip archive. When you have created the .zip archive, you need to copy it to the destination system, unpack the archive and select the files that you want to import into Business Availability Center.

The contents of the .zip archive that is exported from the Metric Definer is as follows:

- a template file, which is imported into Business Availability Center once and is not required again. The content of this file does not change.
- a single entities file (`bpi-entities.xml`), which contains the definitions for all the HPBPI Business Flows and includes Business Metric definitions for each Flow where appropriate.

  Business Metric definitions are included for a Business Flow, only when you have selected them when you export the Flow definition within the Metric Definer.
- individual entity files, which contain the Business Flow, and where appropriate, the Business Metric definitions for the flow. These entity files are named according to the Business Flow that they represent, for example, `1-EWT_Wire Transfer-bpi-entities.xml`.

The reason for having a single entities file and also individual entity files for each Flow definition is for flexibility. Typically, you need only export and import all the Flow definitions from HPBPI to Business Availability Center; however, there are some situations when you might want to import only one

Flow into Business Availability Center. As an example, when you are developing, or testing your integration with Business Availability Center, you might prefer to export and import individual Flow definitions.

To create, or export, your XML entities file from the Metric Definer, complete the following steps:

1. Make sure that the Flow definition that you want to export has been deployed in the HPBPI Modeler. You can not export Flow definitions that have not been deployed.
2. If you want to export Business Metrics, in addition to Business Flows, make sure that you have configured a Data Sample Destination for the Business Availability Center system where you intent to import your Flow definitions. The option within the Metric Definer to export data samples is offered only where there is a Data Sample Destination configured.
3. Open the Business Process Metric Definer as follows:

   `Start | Programs | HP | HP Business Process Insight | Metric Definer`

   If the Metric Definer is already opened, make sure that you use the `Refresh` option to update the Metric Definer display.

   The next step depends on how much HPBPI data you want to export. If you are exporting only Flow definitions in order to obtain IT operational resource information, continue at step 4 to export the file.

   If you want to include Business Metric data in your XML file, complete the following steps for each deployed Flow definition in the Metric Definer:

   a. Select the Flow definition from the left-hand navigation pane.
   b. For each existing Business Metric that you want exported for this Flow definition, select the Business Metric in the right-hand pane and select the `Modify Metric` option.
   c. Select the `BAC Data Samples` checkbox on the Metric definition page.
   d. Select the `Update` or `Replace` button to save your modifications to the Business Metric.

   If you are creating a new Business Metric, select the `BAC Data Samples` checkbox if required when configuring the new Metric definition.

4. Select `File|Export|BAC XML File Adapter files` from the Metric Definer menu in the left-hand navigation pane.

   A `File Download` dialog is displayed asking you if you want to save the file.

5. Click the Save button.

   A `Save As` dialog is displayed where you can specify a location for the .zip archive.

6. Navigate to the location where you want to save the .zip archive file and click `Save`.

7. The .zip archive is created ready for you to unpack and copy to the Business Availability Center system.

   You have now completed the steps to export the .zip archive file containing the HPBPI data for your Business Availability Center system.

You now need to import the exported files into the Business Availability Center system. To do this, you need to create a Business Availability Center XML File source adapter. Instructions for creating XML File source adapters can be found in the Business Availability Center documentation (*Business Availability Center - Source Manager Administration Guide*).

XML File source adapters require two files:

- a template file

  This file contains the information required to map the HPBPI data samples received within Business Availability Center to the correct CI types.

- an XML configuration file

  This is one of the XML files in the .zip archive. It can be the XML file that contains information relating to all the Business Flows, or it can be an XML file that contains information relating to a specific Business Flow. The XML file contains a representation of the HPBPI data, for example, Flow definitions, that can be converted to BPI Business Process CIs in the Business Availability Center CMDB.

When you have imported the XML files, the configuration of your data samples is now complete and the integration between Business Availability Center and HPBPI is also complete.

Creating a My Business Availability Center Application for HPBPI

If you want to have a real-time view of the HPBPI Flow diagram and Metric Threshold dials as shown in the HPBPI Business Process Dashboard you can create a portlet within Business Availability Center.

You can add a Business Process Insight portlet to My BAC to display the Flow diagram. Refer to the Business Availability Center documentation and online help for details of adding portlets to a My BAC page.

Figure 23 shows an example of a flow summary as viewed through a Business Availability Center portlet.

Figure 23 Flow Summary Example

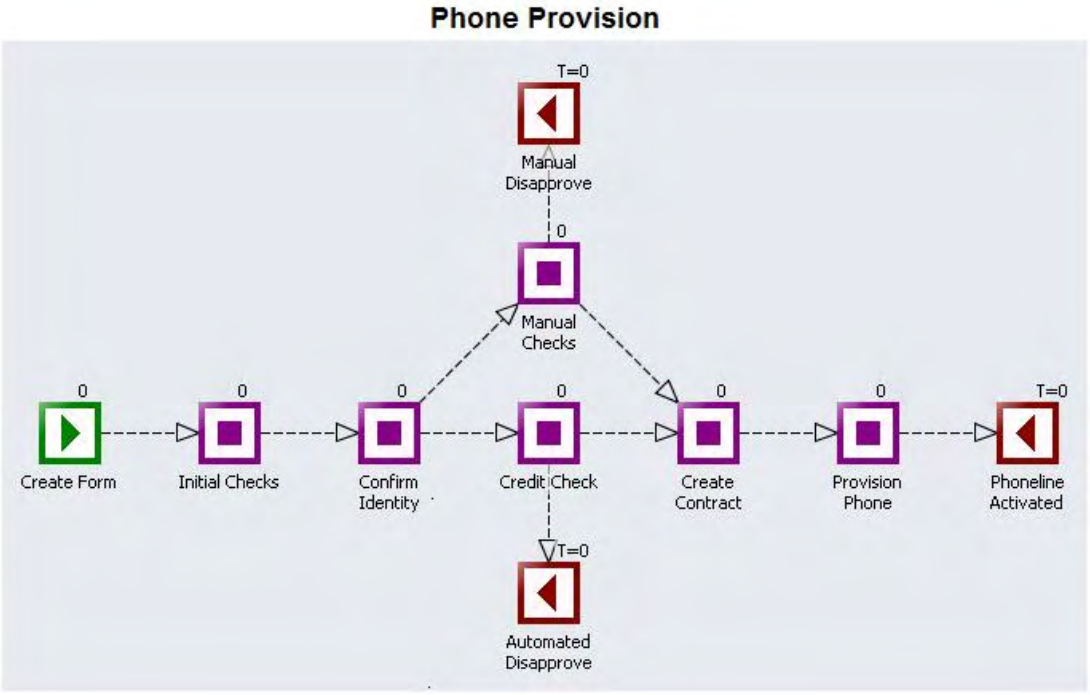


Figure 24 shows an example of threshold dials.

Figure 24 Threshold Dials

In addition to adding the portlet to the My BAC page, you need to create an XML File source adapter. You need to include all the Flow and Business Metric definitions that you want to view in the exported XML file, as described in section Creating an XML File Source Adapter on page 80.

Description of Data Samples Sent to Business Availability Center

HPBPI sends data samples to Business Availability Center and all the HPBPI data samples are mapped to the following KPIs within Business Availability Center:

- backlog

  The data sample for this KPI is taken from the backlog value as calculated as part of the Business Metric statistics (Metric_Fact_Statistics database table).

- throughput

  The data sample for this KPI is taken from the throughput value as calculated as part of the Business Metric statistics (Metric_Fact_Statistics database table).

- duration

  The data sample for this KPI is taken from the duration value as calculated in the Metric_Fact_Values database table.

  This KPI is displayed only if the Business Process includes a Business Metric where the Metric Type is equal to `Duration`.

- business health

  This KPI shows the overall health of the bpi_step in the Business Process within Business Availability Center. It is based on the status of each of the other KPIs. The status of each KPI is compared to the other KPIs and the worst case status is reported as the overall business health for each Node in the Business Process.

These data samples include as much information as is possible to give the KPI context when displayed within the Business Availability Center Dashboard. This includes the text that you add to the Description field when you create the Metric Threshold, which is displayed as the Units Description from within the Business Availability Center Dashboard. The description text displayed depends on status of the backlog, throughput and duration KPIs.

In the case of the business health for a Business Process Metric, the health of the specific Business Metric is reported. When reporting the overall health, any Groups that you have defined in the Metric Definer are ignored, it is the overall metric statistic that is reported

It might seem obvious, but in order to have Business Metric data reported though the Business Availability Center, you must be collecting statistics on your Business Metrics.

If all the backlog, throughput and duration thresholds in the Business Process Metric have a status of Normal, the Business Availability Center Dashboard displays the Business Process Metric Description as the Units Description in all the relevant KPIs.

If any of the backlog, throughput and duration thresholds in the Business Process Metric have a status other than Normal, the Business Availability Center Dashboard displays the Threshold Description for the most severe status as the Units Description in the corresponding Business Availability Center KPI.

The time interval between data samples can be configured using the `Data Sample send interval (seconds)` parameter in the HPBPI Administration Console.

Business Availability Center System Configuration

In addition to the configuration of HPBPI described in this chapter, you also need to configure your Business Availability Center system, and provide details of the HPBPI Server host name and the port number for the servlet engine on the HPBPI Server host.

The servlet engine port number is identified as the `Servlet Engine HTTP` port number within HPBPI Administration Console.

The HPBPI Server host name can be specified as a fully qualified host name, or as an IP address.

Within the Business Availability Center Dashboard, you modify the Infrastructure Settings for Platform Administration. There is an entry listed for the Business Process Insight Integration table where the host name and port number are listed.

5 HPBPI and SOA Manager

This chapter describes how to integrate HPBPI with HP SOA Manager, specifically to enable HPBPI to report on the status of SOA Manager business services.

HPBPI can also integrate with SOA Manager and use SOA Manager as a source of business events. This is described in the *Business Process Insight Integration Training Guide - Business Events*.

The chapter is structured as follows:

- How HPBPI integrates with SOA Manager; see section SOA Manager and HPBPI Integration on page 90.
- Obtaining status information from SOA Manager; see section Business Events and SOA Manager on page 93.
- Configuring the HPBPI SOA Manager Adapter; see section Configuring Access to the SOA Manager Adapter on page 94.

SOA Manager and HPBPI Integration

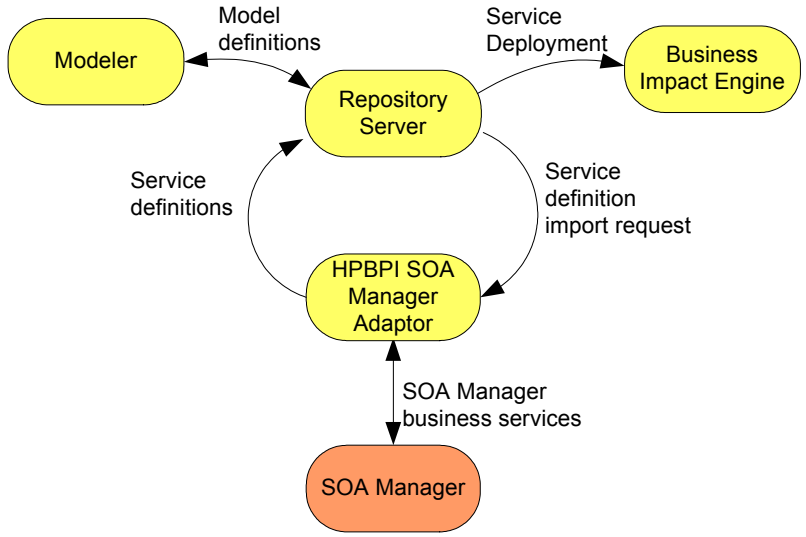
HP SOA Manager enables you to manage your service oriented architecture (SOA) resources to ensure their reliability and optimize their performance. It is a model-based management solution for managing a SOA in the context of understanding the health of the business services that the SOA architecture supports.

There are two integration points between HPBPI and SOA Manager:

1. The first is the ability to use the SOA Manager service model to provide the status of SOA Manager business services to HPBPI. This integration is through the HPBPI SOA Manager Adapter as shown in Figure 25 on page 91. Configuring the SOA Manager Adapter is described in section Configuring Access to the SOA Manager Adapter on page 94.
2. The second is where HPBPI can receive business events from SOA Manager for the HPBPI business flows that it is monitoring. This integration is described in section Business Events and SOA Manager on page 93. The details of how to configure this integration, through the Business Event Handler, is described in the *Business Process Insight Integration Training Guide - Business Events*.

Figure 25 shows how the SOA Manager Adapter enables you to link to SOA Manager business services and make them available to HPBPI business flows through the Repository Server.

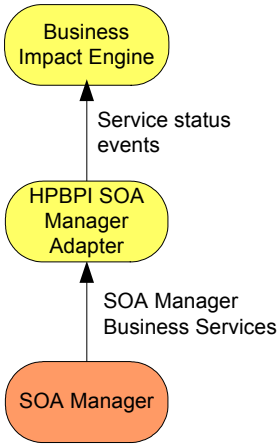
Figure 25 HPBPI and SOA Manager Business Services



When you have configured the SOA Manager Adapter, you can import the required SOA Manager business services and link them to your business flows. If you do not configure the integration, you are unable to use the HPBPI Modeler to link to these business services or synchronize with them.

You then deploy the business flow and the services are deployed to the Business Impact Engine to be monitored; see Figure 26.

Figure 26 HPBPI SOA Manager Adapter - Business Service Deployment



The framework for the SOA Manager Adapter is installed as part of HPBPI. You create an instance of the SOA Manager Adapter for each SOA Manager system that you are monitoring from the HPBPI Administration Console.

HPBPI service definitions that you enter using the HPBPI Modeler are linked to business service definitions made available from SOA Manager, through the HPBPI SOA Manager Adapter.

At run time, status changes for the business services that you have subscribed to in your flows are monitored and managed by the Business Impact Engine.

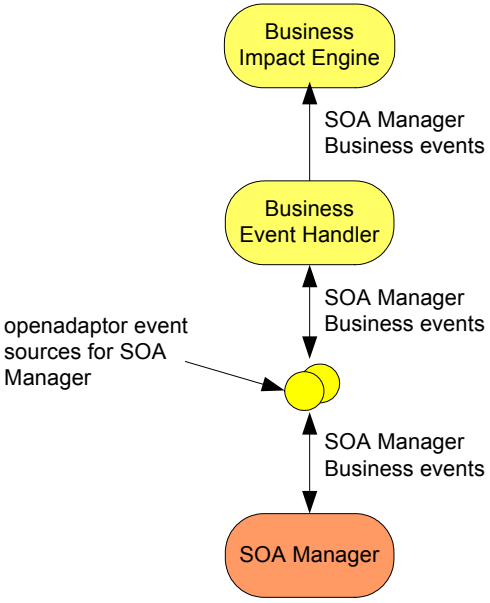
The HPBPI SOA Manager adapter polls SOA Manager for updates to the business services that it is reporting on. These are the services that you have linked to through the HPBPI Modeler.

Business Events and SOA Manager

You can also configure your HPBPI system to receive SOA business events by configuring an event source for SOA Manager using openadaptor; see Figure 27. An event source specifically for SOA Manager is installed with your HPBPI Server.

You need to complete the openadaptor configuration and link it to the HPBPI event sink as described in the *HP Business Process Insight Training Guide - Business Events*.

Figure 27 HPBPI and SOA Manager Business Events



Configuring Access to the SOA Manager Adapter

This section describes how to configure HPBPI to receive business service status information from SOA Manager.

You need to make sure that you have installed the adapter and created an instance of the adapter on the target machine; refer to the *Business Process Insight Installation Guide* for details of installing and starting the adapter.

When you have installed and configured an instance of the adapter, you need to configure the integration (service source) between HPBPI and the SOA Manager adapter using the Administration Console as follows:

1. Select `Operational Service Sources` from the Navigation tree in the Administration Console.

   The right-hand pane shows a list of Operational Service Sources.

2. From the right-hand pane, select the `Add` button to add a new Service Source for the SOA Manager adapter instance that you have created.

   You are presented with the `Add Operational Service Source` dialog.

3. Select the following adapter type:

   `Service Oriented Architecture Manager`

   You are presented with the `Service Oriented Architecture Manager Source Properties` dialog.

4. Enter values for the properties of the SOA Manager Service Source. The properties are fully described in the *Business Process Insight Administration Guide*:

   — `Service Source Name`

   — `Description` (Optional)

   — `Hostname`

   — `Port`

   — `Status event poll interval`

   — `Connection timeout (seconds)`

- Override Service Adapters' log level?
- Log level for this adapter

  In addition, you can configure a Web Proxy for your Web Services connection if you have one.

5. Click the OK button when you modifications are complete.
6. Make sure that the Enabled check box is selected in the column next to the new service source entry.
7. Click the Apply button to apply your changes
8. Move to the Component Status screen and stop and restart all the HPBPI components.

The configuration is now complete and you can access SOA Manager business services from HPBPI.

6 HPBPI and OVIS

This chapter describes how to integrate HPBPI with HP OpenView Internet Services (OVIS). Integration with OVIS includes:

- using OVIS to report on the status of operational services.
- using OVIS to report on SLO and SLA violations.
- configuring the HPBPI custom probes for OVIS. These probes are used to sample functions of the HPBPI business flows for monitoring purposes. You can also set report on SLO and SLA violations resulting from these custom probes.

This chapter describes how to configure the HPBPI custom probes to monitor multiple HPBPI Servers; it does not discuss using OVIS as a source of operational services.

The chapter is structured as follows:

- How HPBPI integrates with OVIS; see section HPBPI and OVIS Integration on page 99.
- Using OVIS to report on the operational status of your HPBPI system; see section Reporting on the Operational Status of Your HPBPI System on page 104.
- Using OVIS to report SLO and SLA violations resulting from changes in the operational status of your HPBPI system; see section Reporting SLO and SLA Violations on page 104. The violations can be reported to the business manager through email using the Notification Server.

  Refer to the *HPBPI System Administration Guide* for details of how to set up the Notification Server to send email notifications concerning these violations.

- Configuring the HPBPI custom probes for OVIS; see section Custom Probes on page 105.
- Configuring the HPBPI custom probes for multiple HPBPI Servers; see section Configuring Probes for Multiple HPBPI Servers on page 123

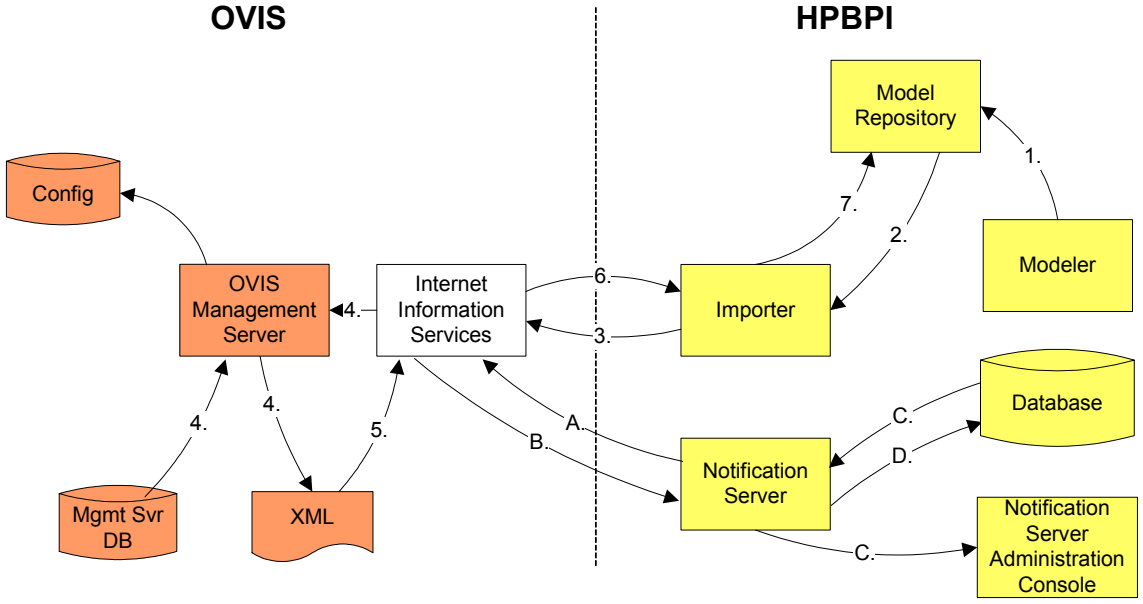
HPBPI and OVIS Integration

There are two aspects to the OVIS integration with HPBPI: design-time and run-time integration. The integrations are shown in section HPBPI and OVIS Design-Time Integration on page 99 and in section HPBPI and OVIS Run-Time Integration on page 101.

HPBPI and OVIS Design-Time Integration

Figure 28 on page 99 shows the HPBPI and OVIS design-time configuration.

Figure 28 HPBPI and OVIS Design-Time Integration



There are two design-time scenarios shown in Figure 28 on page 99:

- Importing the Service Hierarchy
- Making SLO and SLA details available

Importing the Service Hierarchy

The first scenario is where the OVIS service hierarchy is imported into the HPBPI Model Repository so the OVIS services can be referenced within business flows:

Step 1: User selects the `Link to Services` option within the Modeler to import the OVIS service definitions. The import instruction is sent to the Model Repository. (The Model Repository can also import the OVIS service definitions as a background task, according to its configuration.) Refer to the *HPBPI System Administration Guide* for details of configuring the Model Repository parameters.

Step 2: The Model Repository starts the Importer component.

Step 3: The Importer component issues an HTTP request to the OVIS Server (through the OVIS Web Server).

Step 4: Web Server contacts the OVIS Management Server to obtain the requested service hierarchy from the OVIS database and converts it to an XML document.

Step 5: The XML document is sent to the Web Server.

Step 6: The XML document is sent by the Web Server using an HTTP response to the Importer component, which parses the XML.

Step 7: The parsed XML is stored in the Model Repository and made available to the user through the Modeler interface.

SLO and SLA Details Made Available

The second scenario is where the SLO/SLA details in the service hierarchy are provided to the Notification Server Administration Console where you can subscribe to those that you want to monitor for potential violations.

Step A: Using the Notification Server Administration Console, the user selects the option to refresh the OVIS Service Level configuration information. At this point the Notification Server sends an HTTP request to the OVIS Server (through the OVIS Web Server). The next sequence of steps are identical to those above (steps 4 to 6). The OVIS Service Level configuration information is also refreshed when the Notification Server Administration Console is started.

Step B: The XML document containing the service hierarchy is returned to the Notification Server.

Step C: The Notification Server parses the service hierarchy (XML document) and displays the available SLOs and SLAs to the user through the Notification Administration Console. Users can then select to subscribe to the SLO and SLA violations.

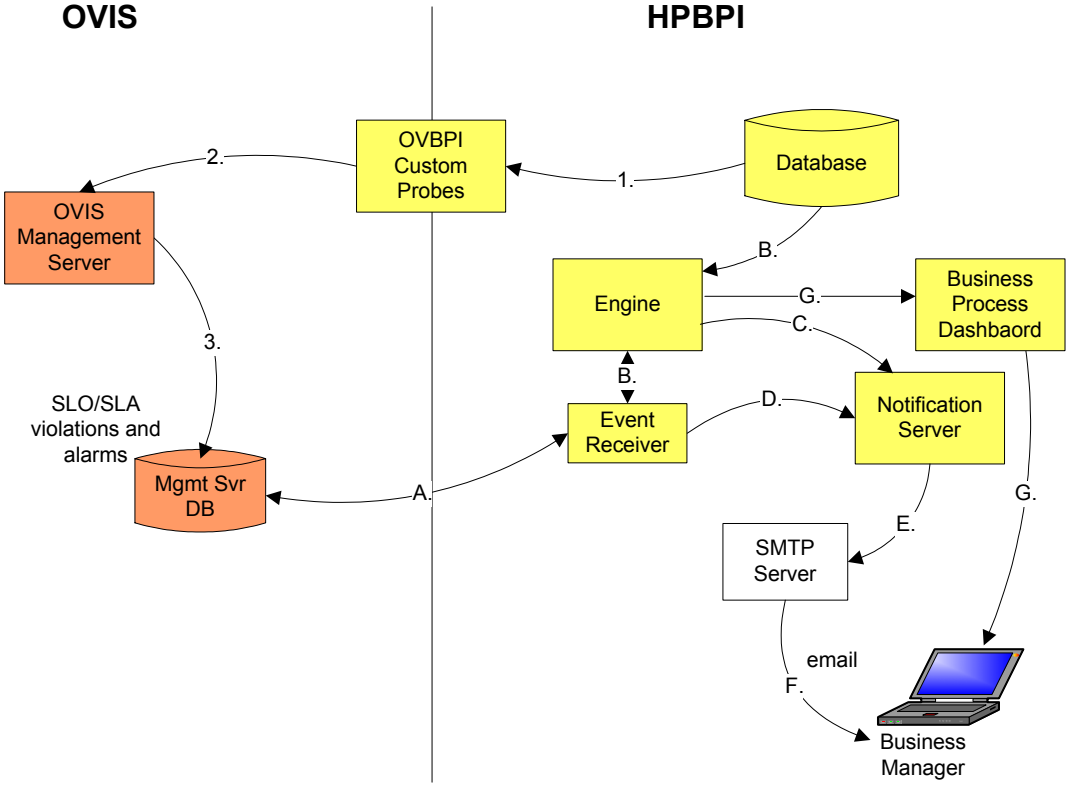
Step D: Details of the user subscriptions are stored in the HPBPI database.

Email notifications are delivered as described in the run-time integration; see section HPBPI and OVIS Run-Time Integration on page 101.

HPBPI and OVIS Run-Time Integration

Figure 29 on page 101 shows the HPBPI and OVIS run-time configuration.

Figure 29 HPBPI and OVIS Run-Time Integration



There are three run-time integration scenarios shown in Figure 29.

- Monitoring HPBPI flows using the HPBPI OVIS custom probes
- Receiving flow impact alarms from OVIS
- Sending emails for OVIS Service Level violations

Monitoring HPBPI Flows Using the OVIS Custom Probes

The following are the steps for monitoring HPBPI, as related to the steps in Figure 29 on page 101:

Step 1: The custom probes take their data from the HPBPI database.

Step 2: An HTTP message containing the probe data is sent to the OVIS Management Server.

Step 3: The probe data is then loaded into the Management Server database in order that it can be measured and used for reporting.

Receiving Flow Impact Alarms from OVIS

The following are the steps for receiving flow impact alarms, as related to the steps in Figure 29 on page 101:

Step A: The HPBPI Event Receiver polls the Management Server database for service impact alarms.

Step B: The data on service impact alarms is sent by the Event Receiver to the Business Impact Engine, where the Engine converts the data into internal business events (service impact events). Alarm events are processed by the Business Object Manager component of the Engine and the Engine then updates the status entry in the HPBPI database.

Step C: `FLOW_IMPACT` events, resulting from any alarms, are sent to the Notification Server.

Step E: The email notifications from the Notification Server are forwarded to your email Server.

Step F: The email server delivers the email notifications containing the alerts for service impacts alarms.

Step G: The Business Process Dashboard is continually polling the database for status changes and, according to how it is configured, displays the current status of your system.

Note that OVIS writes alarms to its database at specified intervals, and HPBPI is configured to read these alarms at specified intervals. It is therefore possible for the Business Process Dashboard to be reporting different results for an OVIS probe, during this period; for example, the Dashboard might report that the overall OVIS Service status is `Healthy`, but when you link to the OVIS Dashboard from the HPBPI Dashboard, the equivalent OVIS Service might be marked as impacted.

Sending Emails for OVIS Service Level Violations

The following are the steps for sending email alerts as a result of SLO/SLA violations, as related to the steps in Figure 29 on page 101:

Step A: The HPBPI Event Receiver polls the Management Server database for SLO/SLA violations.

Step D: The data on any SLO and SLA violations is sent directly to the Notification Server.

Step E: The email notifications from the Notification Server are forwarded to your email Server.

Step F: The email server delivers the email notifications containing the alerts for SLO and SLA violations.

Reporting on the Operational Status of Your HPBPI System

You can use OVIS to report on the operational status of your HPBPI system in the same way as you can use OVIS to report on any other component within your business system.

There are no HPBPI-specific configuration considerations, you use the OVIS Configuration Manager to create and configure probes that simulate the use of the services that you want to monitor within HPBPI. These services might simulate the response from the Business Process Dashboard or the database. You use OVIS to configure probes to report the status of your HPBPI system in the same way as you use it to configure probes for other applications.

For information relating to creating OVIS probes to monitor the IT components on which HPBPI depends, refer to the OpenView Internet Service documentation. This chapter does not explain how to set up standard operational alarms within OVIS.

Reporting SLO and SLA Violations

You can use OVIS to report the HPBPI business flows and their adherence to operational Service Level objectives and conformance to operational Service Level Agreements. Service Level Agreements are created using the OVIS Configuration Manager and can be reported through the OVIS Dashboard.

For information relating to creating SLO and SLA violation alarms using OVIS, refer to the OpenView Internet Service documentation. This chapter does not explain how to set up standard operational alarms within OVIS.

Custom Probes

In addition to creating operational probes and reporting on SLO and SLA violations, you can use OVIS to report on specific details of your HPBPI flows. HPBPI provides three custom probes for this purpose.

You configure OVIS to monitor HPBPI using the OVIS Configuration Manager. Full details of using the OVIS Configuration Manager are provided in the OVIS documentation. This section describes the HPBPI-specific probe configuration that you need to complete within the OVIS Configuration Manger, following the HPBPI Probes installation.

Before starting the configuration, make sure that you have installed the HPBPI custom probes on the system where the OVIS Management Server is running. Instructions for the installation are provided in the *Business Process Insight Installation Guide*.

To configure the HPBPI custom probes within OVIS, you first need to create one or more Service Groups within an existing, or newly defined, customer group (`Customers`). A Service Group is where you define a particular service that you want to monitor within OVIS.

HPBPI provides three probes, which enable you to configure specific Monitored Service types within an OVIS Service Group. Within OVIS these Monitored Service types are listed as:

- `C_OVBPI_FLOW - OVBPI_FLOW Probe`

  This probe monitors the following characteristics of flows:

  — The number of active flow instances at the time when the probe is executed.

  — The sum of the values of the weight property for the active flow instances at the time when the probe is executed.

— Throughput values for completed flow instances. The throughput is calculated based on the number of completed flow instances within the configured probe period. The throughput rate is normalized into an hourly rate.

— Throughput values for the weight values for the completed flow instances. The throughput is calculated based on the weight values for the completed flow instances within the configured probe period. The throughput rate is normalized into an hourly rate.

As an example, the probe might calculate the number of claims that were active when the probe was executed, the value of these active claims and the throughput rate of the claims.

- `C_OVBPI_METRIC - OVBPI_METRIC Probe`

  This probe monitors HPBPI business process metric types that you have defined. The probe returns:

  — The time (in seconds) that the most recent business metric took to complete. This is the most recent business metric completed within the configured probe period.

  — The average time taken for all the business metrics to complete within the probe period.

The result of this OVIS metric is reported as `unavailable` if there is no data available. This is because no flow instances have fulfilled the probe requirement in the time interval configured for the probe.

The result of this scenario is that probe data can be intermittently unavailable.

- `C_OVBPI_NODE - OVBPI_NODE Probe`

  This probe monitors the following characteristics of a node:

  — The number of active flow instances at the node at the time when the probe is executed.

  — The sum of the values of the Weight properties for the active flow instances at the node at the time when the probe is executed.

— Throughput values for completed flow instances at the node. The throughout is calculated based in the number of completed flow instances for the node, within the configured probe period. The throughput rate is normalized to an hourly rate.

— Throughout values for the weight values for the completed flow instances at the node. The throughput is calculated based on the weight values for the completed flow instances for the node, within the configured probe period.

As an example, the probe might calculate the number of insurance claims that have been active at a specified node when the probe is executed, the total value of these active claims and the throughput rate of active claims.

The list of Monitored Services includes many services that you can monitor and configure within OVIS. This section covers only the HPBPI-specific services. Refer to the OVIS documentation and OVIS online-help for details of the Monitored Services that are not specific to HPBPI.

Section Creating a New Service Group for an HPBPI Service on page 107 describes the tasks that you need to complete to create a Service Group to define HPBPI-specific Monitored Services. It also describes the information required to create a Service Target for the Service Group, or Service Groups, that you create.

Creating a New Service Group for an HPBPI Service

To define HPBPI-specific services within OVIS, complete the following steps:

1. Start the OVIS Configuration Manager as follows:

   `Start|Programs|HP OpenView|Internet Services|Configuration Manager`

2. Navigate to the `Customer Name` where you want to add your new service, or create a new `Customer`.

3. Create a new Service Group and select the HPBPI-specific Monitored Service that you want to create. The Monitored Service can be one of:
   - `C_OVBPI_FLOW - OVBPI_FLOW Probe`
   - `C_OVBPI_METRIC - OVBPI_METRIC Probe`
   - `C_OVBPI_NODE - OVBPI_NODE Probe`
4. Provide the information required for the Monitored Service according to the service type that you are configuring as follows:
   - to monitor flow instances across the whole flow, refer to section Flow Instances (C_OVBPI_FLOW) on page 108.
   - to monitor business process metric durations, refer to section Flow Metrics (C_OVBPI_METRIC) on page 110.
   - to monitor flow instances for a specific node, refer to section Node Instances (C_OVBPI_NODE) on page 112.

Flow Instances (C_OVBPI_FLOW)

This probe obtains flow information from the Business Impact Engine database. It uses this information to calculate:

- The number of active flow instances at the time when the probe is executed.
- The sum of the values of the weight property for the active flow instances at the time when the probe is executed.
- Throughput values for completed flow instances.
- Throughput values for the weight values for the completed flow instances.

The values (or metrics) returned by OVIS are compared with the objectives that you set in the `Objective Information` dialog within OVIS. Each Objective setting defines the expected limits for a specific OVIS metric. These limits are applied to every Service Target in the Service Group. The limits are used to determine when a Service Level Objective is violated and when an alarm event should be generated.

This probe type requires the name of the flow for which instances need to be counted.

When creating a Service Target for this HPBPI Flow Metric, you need to enter the information listed in Table 1 for the Service Target named C_OVBPI_FLOW.

Table 1 Flow Instances Probe

Parameter Name	Description
Target Host	The fully qualified host name of the system where the HPBPI database is located. This is the same as the hostname for the system where the HPBPI Server is running. This value must match the hostname in the probes configuration file described in section Configuring Probes for Multiple HPBPI Servers on page 123.
Port	This can be left as the default value presented, as it is not used by HPBPI.
Username	HPBPI does not use the information entered in this field; however, OVIS does require the value to be null, or a value that is valid for the system where you are using the custom probes.
Password	HPBPI does not use the information entered in this field; however, OVIS does require the value to be null, or a value that is valid for the system where you are using the custom probes.
OVBPI_Identifier	A unique identifier that you choose, and that must match the unique identifier in the probes configuration file; see section Configuring Probes for Multiple HPBPI Servers on page 123. The default value for the OVBPI_Identifier is OVBPI.
OVBPI_DB_User_Name[a]	The HPBPI database username that you provided for HPBPI during the HPBPI installation process.

Table 1 Flow Instances Probe

Parameter Name	Description
OVBPI_DB_Password	The password for the HPBPI database user that you provided for HPBPI during the HPBPI installation process.
OVBPI_Flow_Name	The name of the HPBPI flow for which you want to monitor flow instances.

a. This information can be found through the HPBPI Administration Console under the Oracle Server Access or MS SQL Server Access option, according to your configuration.

Flow Metrics (C_OVBPI_METRIC)

This probe obtains HPBPI business process metric information from the Metric Views tables in the database. It uses this information to calculate:

- The time (in seconds) that the most recent business metric took to complete.
- The average time taken for all the business metrics to complete within the probe period.

The values (or metrics) returned by OVIS are compared with the objectives that you set in the Objective Information dialog within OVIS. Each Objective setting defines the expected limits for a specific OVIS metric. These limits are applied to every Service Target in the Service Group. The limits are used to determine when a Service Level Objective is violated and when an alarm event should be generated.

The probe requires details of the name of the flow and the name of the associated business process metric defined in the Metric definer in order to monitor this service.

When creating a Service Target for this HPBPI Flow Metric, you need to enter the information listed in Table 2 for the Service Target named `C_OVBPI_METRIC`.

Table 2 Flow Metric Probe

Parameter Name	Description
`Target Host`	The fully qualified host name of the system where the HPBPI database is located. This is the same as the hostname for the system where the HPBPI Server is running. This value must match the hostname in the probes configuration file described in section Configuring Probes for Multiple HPBPI Servers on page 123.
`Port`	This can be left as the default value presented, as it is not used by HPBPI.
`Username`	HPBPI does not use the information entered in this field; however, OVIS does require the value to be null, or a value that is valid for the system where you are using the custom probes.
`Password`	HPBPI does not use the information entered in this field; however, OVIS does require the value to be null, or a value that is valid for the system where you are using the custom probes.
`OVBPI_Identifier`	A unique identifier that you choose, and that must match the unique identifier in the probes configuration file; see section Configuring Probes for Multiple HPBPI Servers on page 123. The default value for the `OVBPI_Identifier` is `OVBPI`.
`OVBPI_DB_User_Name`[a]	The HPBPI database user name that you specified for HPBPI during the HPBPI installation process.

Table 2 Flow Metric Probe

Parameter Name	Description
OVBPI_DB_Password	The password for the HPBPI database user that you provided for HPBPI during the installation process.
OVBPI_Flow_Name	The name of the flow that the metric that you want the probe to monitor is associated with.
OVBPI_Metric_Name	The name of the business process metric defined for the flow specified in OVBPI_Flow_Name.

a. This information can be found through the HPBPI Administration Console under the Oracle Server Access or MS SQL Server Access option, according to your configuration.

Node Instances (C_OVBPI_NODE)

This probe obtains information from the Business Impact Engine database. It uses this information to calculate:

- The number of active flow instances at the node at the time when the probe is executed.
- The sum of the values of the Weight properties for the active flow instances at the node at the time when the probe is executed.
- Throughput values for flow instances that meet the Complete Conditions for the node.
- Throughout values for the weight values for flow instances that meet the Complete Conditions for the node.

The values (or metrics) returned by OVIS are compared with the objectives that you set in the Objective Information dialog within OVIS. Each Objective setting defines the expected limits for a specific OVIS metric. These limits are applied to every Service Target in the Service Group. The limits are used to determine when a Service Level Objective is violated and when an alarm event should be generated.

This probe type requires the name of the flow and the name of the node in the flow.

When creating a Service Target for this HPBPI Flow Metric, you need to enter the information listed in Table 3 for the Service Target named C_OVBPI_NODE_PROBE.

Table 3 Node Instance Probe

Parameter Name	Description
Target Host	The fully qualified host name of the system where the HPBPI database is located. This is the same as the hostname for the system where the HPBPI Server is running. This value must match the hostname in the probes configuration file described in section Configuring Probes for Multiple HPBPI Servers on page 123.
Port	This can be left as the default value presented, as it is not used by HPBPI.
Username	HPBPI does not use the information entered in this field; however, OVIS does require the value to be null, or a value that is valid for the system where you are using the custom probes.
Password	HPBPI does not use the information entered in this field; however, OVIS does require the value to be null, or a value that is valid for the system where you are using the custom probes.
OVBPI_Identifier	A unique identifier that you choose, and that must match the unique identifier in the probes configuration file; see section Configuring Probes for Multiple HPBPI Servers on page 123. The default value for the OVBPI_Identifier is OVBPI.
OVBPI_DB_User_Name[a]	The HPBPI database username that you provided for HPBPI during the HPBPI installation process.
OVBPI_DB_Password	The password for the HPBPI database user that you provided for HPBPI during the HPBPI installation process.

Table 3 Node Instance Probe

Parameter Name	Description
OVBPI_Flow_Name	The name of the HPBPI flow for which you want to monitor flow instances.
OVBPI_Node_Name	The name of the node within the HPBPI business flow that you are sampling.

a. This information can be found through the HPBPI Administration Console under the Oracle Server Access or MS SQL Server Access option, according to your configuration.

When you have created a Service Group and created and configured a Service Target for one of these Service Groups, you can continue and define alarms, SLO and SLAs for the probes, according to your requirements.

You can configure multiple Service Targets for a particular Service Group; these Service Targets all have the same Monitored Service. You might want to create multiple Service Targets where a Node (or Nodes) in an HPBPI Flow is dependent on the status of a combination of several HPBPI Flows, Nodes or Metrics (according to probe type). In this case, you can create a Service Group that had several Service Targets, each with different probe properties defined.

You also need to create a new probe location for the Service Targets. In the case of the HPBPI probes, the probe location must be Local System.

Configuring Alarms and SLOs

This section describes how to create alarms and SLOs for OVIS Monitored Services. These might be HPBPI-specific Monitored Services or they might be other Monitored Services that you have created in order to monitor the operational services on which an HPBPI Flow relies. This section provides an overview of the tasks that you need to complete where they are specific to HPBPI. The OVIS documentation and OVIS online-help provides detailed information about configuring alarms and SLOs, which you should read in addition to the information supplied in this chapter.

Defining Objective Information for HPBPI Monitored Services

You create Alarms and SLOs for a particular Service Group through the Service Objectives dialog within the OVIS Configuration Manager. Each Service Group that you define has a related Service Target (or Service Targets) and Service Objective. You defined the Service Target in section Creating a New Service Group for an HPBPI Service on page 107.

This section describes the OVIS metrics that are available for the HPBPI custom probes when configuring Alarm and SLO details through the Service Objective dialog.

An OVIS metric is one of a number of values returned by an OVIS probe. As an example, the HPBPI_METRIC_PROBE returns the metrics listed in Table 5 on page 118 each time that the probe is executed.

The HPBPI-specific Service Objective details that you need to supply are specific to each probe type. The required information is described in the following sections:

- C_OVBPI_FLOW_PROBE Objective Information on page 116, for the HPBPI `C_OVBPI_FLOW_PROBE` Monitored Service.
- C_OVBPI_METRIC_PROBE Objective Information on page 118, for the HPBPI `C_OVBPI_METRIC_PROBE` Monitored Service.
- C_OVBPI_NODE_PROBE Objective Information on page 119, for the HPBPI `C_OVBPI_NODE_PROBE` Monitored Service.

C_OVBPI_FLOW_PROBE Objective Information

When configuring one Service Level Objective for this probe, you can use one of the OVIS metrics listed in Table 4 on page 116. These OVIS metrics relate to active flow instances for the flow.

Note that you can define more than one Alarm, Service Objective and associated OVIS metric.

Table 4 Service Objective Metrics for C_OVBPI_FLOW_PROBE

OVIS Metric	Description
AVAILABILTY	Standard OVIS Metric: Set to zero (0) to indicate when no measurement can be retrieved. Set to one (1) to indicate when the service is available.
SETUP_TIME	Standard OVIS Metric: Time taken to establish the connection to the HPBPI database.
RESPONSE_TIME	Standard OVIS Metric: Time taken for HPBPI Monitored Service to respond.
TRANSFER_TPUT	Not used by HPBPI.
ACTIVE_INSTANCES_LOW	HPBPI-specific Metric: The lowest number of active flow instances that are acceptable for the defined flow, before the level is considered to be too low.
ACTIVE_INSTANCES_HIGH	HPBPI-specific Metric: The highest number of active flow instances that are acceptable for the defined flow, before the level is considered to be too high.
ACTIVE_VALUES_LOW	HPBPI-specific Metric: The lowest Weight value for active flow instances that is acceptable for the defined flow, before the level is considered to be too low.

Table 4 Service Objective Metrics for C_OVBPI_FLOW_PROBE

OVIS Metric	Description
ACTIVE_VALUES_HIGH	HPBPI-specific Metric: The highest Weight value for active flow instances that is acceptable for the defined flow, before the level is considered to be too high.
INSTANCE_TPUT_LOW	HPBPI-specific Metric: The lowest number of completed flow instances for the configured probe period (throughput) for the defined flow, before the number is considered to be too low. This value is reported as an hourly rate.
INSTANCE_TPUT_HIGH	HPBPI-specific Metric: The highest number of completed flow instances over the configured probe period (throughput) for the defined flow, before the number is considered to be too high. This value is reported as an hourly rate.
VALUE_TPUT_LOW	HPBPI-specific Metric: The lowest value for the Weight parameter for the completed flow instances over the configured probe period (throughput), before the level is considered to be too low. Shown as an hourly rate.
VALUE_TPUT_HIGH	HPBPI-specific Metric: The highest for the Weight parameter for the completed flow instances over the configured probe period (throughput), before the level is considered to be too high. Shown as an hourly rate.

The remainder of the service level, alarm and objective fields (for example, `Duration`) can be set as they are for all other OVIS Monitored Services according to your business requirements.

C_OVBPI_METRIC_PROBE Objective Information

When configuring an Alarm or Service Level Objective for the probe, you can use one of the OVIS metrics listed in Table 5 on page 118. You can define more than one Alarm, Service Level Objective and associated OVIS metric.

Table 5 Service Objective Metrics for C_OVBPI_METRIC_PROBE

OVIS Metric	Description
AVAILABILTY	Standard OVIS Metric: Zero (0) indicates no measurement can be retrieved, one (1) indicates service is available.
SETUP_TIME	Standard OVIS Metric: Time taken to establish the connection to the HPBPI database.
RESPONSE_TIME	Standard OVIS Metric: Time taken for HPBPI Monitored Service to respond.
TRANSFER_TPUT	Not used by HPBPI.
TBN_DURATION_LOW	HPBPI-specific Metric: The lowest duration (in seconds) that is acceptable for the HPBPI business process metric before it is considered to be too low.
TBN_DURATION_HIGH	HPBPI-specific Metric: The highest duration (in seconds) that is acceptable for the HPBPI business process metric before it is considered to be too high.

The remainder of the service level, alarm and objective fields (for example, `Duration`) can be set as they are for all other OVIS Monitored Services according to your business requirements.

C_OVBPI_NODE_PROBE Objective Information

When configuring one Alarm, or Service Level Objective for an HPBPI business process metric, you can use one of the OVIS business metrics listed in Table 6 on page 119. These OVIS metrics relate to the number of active flow instances at a particular node.

Note that you can define more than one Alarm, Service Objective and associated OVIS metric

Table 6 Service Objective Metrics for C_OVBPI_NODE_PROBE

OVIS Metric	Description
AVAILABILTY	Standard OVIS Metric: Zero (0) indicates not measurement can be retrieved, one (1) indicates service is available.
SETUP_TIME	Standard OVIS Metric: Time taken to establish the connection to the HPBPI database.
RESPONSE_TIME	Standard OVIS Metric: Time taken for HPBPI Monitored Service to respond.
TRANSFER_TPUT	Not used by HPBPI.
ACTIVE_INSTANCES_LOW	HPBPI-specific Metric: The lowest number of active flow instances that are acceptable at the node specified, before the level is considered to be too low.
ACTIVE_INSTANCES_HIGH	HPBPI-specific Metric: The highest number of active flow instances that are acceptable at the node specified, before the level is considered to be too high.
ACTIVE_VALUES_LOW	HPBPI-specific Metric: The lowest Weight value for active flow instances that are acceptable at the node specified, before the level is considered to be too low.

Table 6 Service Objective Metrics for C_OVBPI_NODE_PROBE

OVIS Metric	Description
ACTIVE_VALUES_HIGH	HPBPI-specific Metric: The highest Weight value for active flow instances that are acceptable at the defined node, before the level is considered to be too high.
INSTANCE_TPUT_LOW	HPBPI-specific Metric: The lowest number of completed flow instances at the node specified (over the configured probe period), before the number is considered to be too low. This is a throughput and the parameter is given as an hourly rate.
INSTANCE_TPUT_HIGH	HPBPI-specific Metric: The highest number of completed flow instances at the node specified (over the configured probe period), before the number is considered to be too high. This is a throughput and the parameter is given as an hourly rate.
VALUE_TPUT_LOW	HPBPI-specific Metric: The lowest value for the Weight parameter for completed flow instances at the node specified (over the configured probe period), before the level is considered to be too low. Shown as an hourly rate.
VALUE_TPUT_HIGH	HPBPI-specific Metric: The highest value for the Weight parameter for completed flow instances at the node specified (over the configured probe period), before the level is considered to be too high. Shown as an hourly rate.

The remainder of the service level, alarm and objective fields (for example, `Duration`) can be set as they are for all other OVIS Monitored Services according to your business requirements.

Defining Service Level Agreements for HPBPI Monitored Services

You create an SLA for a particular Service Group through the Service Agreements option for a particular `Customer` within the OVIS Configuration Manager.

A Service Level Agreement can be created based on the results of one or more Service Objectives that are defined for the same Customer.

There are no HPBPI-specific SLA options. You can create SLAs for your HPBPI system using the OVIS options offered through the Service Level Agreements configuration. To access this configuration option, you need to create a new Service Agreement from the OVIS Configuration Manager.

You can also configure users to be alerted to SLA violations using the Notification Server.

Making Sure OVIS Adds Alarm Data to Its Database Tables

Within OVIS there is an option to configure whether or not OVIS enters alarm data to its database, specifically to the database table: `IOPS_ALARM_DATA2`. This option is the `Event DB` or `Database` option on the `Configure Alarm Destinations` dialog. It is the same option as is required to enable NNM integration.

If this option is disabled, no alarm data is written to the table and because HPBPI polls the OVIS database, it cannot therefore report on the OVIS alarms.

You need to make sure that this option is set within the OVIS configuration in order for the integration between HPBPI and OVIS to be successful. Use the OVIS Configuration Manager to access the `Alarm Destinations` option. The option is accessed from the `File|Configure|Alarm Destinations` menu.

Defining Probe Locations

The `Probe Location Info` dialog is the same for all the OVIS probes. You can therefore use the Probe Location Info dialog to configure details of the location of the probes that have been installed for HPBPI.

HPBPI supports local probes only. Therefore, when creating a probe location for an HPBPI probe, you must select `Local System` as the value for the `Probe Location` parameter. The probes are installed on the same system as the OVIS Configuration Manager.

Configuring Probes for Multiple HPBPI Servers

If you have installed more than one HPBPI Server, each using different databases, and you want each HPBPI Server monitored by the same OVIS server, you need to modify the configuration file for the HPBPI custom probes. This file is located at:

OVIS-install-dir\probes\OvbpiProbe.cfg

You can install multiple HPBPI systems (or Servers) within your organization; however, the Servers cannot share business flows, or business flow data in the HPBPI database, they are completely independent implementations.

There is an example format of a section that is used to configure a Microsoft SQL Server database and there is an example format for an Oracle database. You can have as many sections of either database file section type defined in OvbpiProbe.cfg.

The configuration file for the custom probes contains one, or more, of the following sections, where each section starts with a unique identifier:

```
[OVBPI-identifier]
RDBMS_TYPE=SQL Server
ODBC_DRIVER_NAME=SQL Server
OVBPI_DB_SCHEMA_NAME=schema-name
OVBPI_DB_SERVER_HOSTNAME=db-instance-name
```

where:

- *OVBPI-identifier* is a unique identifier that you choose to distinguish this section from other sections in the configuration file.
- SQL Server is the string that you include when you are configuring the custom probes for an HPBPI Server using an RDBMS database type of Microsoft SQL Server database.
- *SQL Server* is the SQL Server ODBC driver name that has been configured for your system. You can find out what it is from:

  Start|Programs|Control Panel|Administrative Tools|Data Sources (ODBC)

- *schema-name* is the name of the database schema that you configured for the HPBPI data in the database specified by *db-instance-name*.
- *db-instance-name* is the name of the database where the OVIS data for HPBPI is located; for example:

  ```
  hostname
  hostname\qualifier
  ```

  The SQL Server database name can be the same as the hostname (including `localhost`) or it can be a combination of hostname and a string that you specify.

```
[OVBPI-identifier]
RDBMS_TYPE=Oracle
ODBC_DRIVER_NAME=Oracle ODBC Driver
ORACLE_NET_SERVICE_NAME=tns-server
```

where:

- *OVBPI-identifier* is a unique identifier that you choose to distinguish this section from other sections in the configuration file.
- `Oracle` is the string that you include when you are configuring the custom probes for an HPBPI Server using an RDBMS database type of Oracle.
- *Oracle ODBC Driver* is the Oracle ODBC driver name that has been configured for your system. You can find out what it is from:

  ```
  Start|Programs|Control Panel|Administrative Tools|Data
  Sources (ODBC)
  ```

- *tns-server* is the TNS name configured in the Oracle file `TNSNAMES.ORA` for the Oracle database that the HPBPI Server is configured for.

The following is an example of the file following an installation of the HPBPI custom probes on the same system as the HPBPI Server, where the HPBPI is configured to use a Microsoft SQL Server database:

```
[OVBPI]
RDBMS_TYPE=SQL Server
ODBC_DRIVER_NAME=SQL Server
OVBPI_DB_SCHEMA_NAME=OvbpiSchema
OVBPI_DB_SERVER_HOSTNAME=host1\sql-svr-2005
```

7 HPBPI and HPSD

HPBPI integrates with HPSD in a number of ways:

- The HPBPI Dashboard can provide links through to appropriate HP Service Desk (HPSD) Service Calls and Incident reports for impacted instances.
- Through a set of predefined flows, adapters and a customized Dashboard, which can be used to monitor HPSD processes; specifically, processes for the following HPSD modules:
  - HPSD Helpdesk Manager
  - HPSD Change Manager
- The HPBPI Notification Server can send HP Operations Manager Messages to an HP Operations Manager system for delivery to HP BTO Software components, including HPSD.

This chapter focuses on the HPSD integration through the Dashboard for Incident reports and Service Calls. For information about the predefined flows and adapters available for monitoring HPSD processes, refer to the *Business Process Insight Integration Training Guide - Monitoring Service Desk*. For information about HPOM notification messages, refer to the *HPBPI System Administration Guide* and the *Business Process Insight Concepts Guide*.

HPBPI and HPSD Service Call and Incident Information

One of the features of HP Service Desk (HPSD) is to provide information about Service Calls and Incidents. A Service Call is a record of a request from a user for support for an IT service. An Incident is an operational event that is not part of the standard operation of the IT system. Both service calls and Incidents relate to operational services that are defined within your HP BTO Software IT implementation. These can be HPOM and OVIS services.

Within HPSD, service items contain information about services that form a particular IT system; these service items can be associated with HP Operations Manager services. Service items can also be associated with Service Level Agreements, which in turn can be monitored using OVIS.

HPBPI integrates with both HPOM and OVIS (at the service group level) for operational service information and, in the case of OVIS, for OVIS metrics definitions.

The information provided through the HPBPI Dashboard's integration with HPSD provides the HPSD context for the services that are being monitored and tracked through the dashboard. For example, if a service is not available for any reason, the HPSD Service Call and Incident information indicates whether the problem is being addressed, who is responsible for resolving the problem, and how many users are affected.

HPBPI integrates with HPSD through the HPBPI Business Process Dashboard. This integration is achieved through the HPSD WEB-API, which is a Java interface used to access the HPSD data. This chapter describes the two main integration options and how you configure both the HPBPI Server and HPSD to achieve the integration that you require.When integrated, the appropriate Service Call and Incident information for a service is shown through the HPBPI Business Process Dashboard.

The following sections describe how you configure HPBPI to integrate with HPSD. The sections assume that you understand how to use HPSD and its management interface.

HPSD Web API

The HPBPI Business Process Dashboard obtains information on Service Calls and Incident reports from HPSD using the HPSD Web API, specifically, using the Java web archive, `web-api.jar`. Both HPBPI and HPSD must use the same version of this file and HPSD changes this `.jar` file with every release, including service pack releases. Therefore, if you are integrating with a version of HPSD that is not the version (including patch version) specified in the *Business Process Insight Installation Guide*, you need to copy the file `web-api.jar` from the HPSD system to the HPBPI system. The file is located in the `api` folder of the appropriate CD-ROM, or disk image, within the Service Desk distribution media.

Before copying this file to the HPBPI system, you need to shut down all the HPBPI Server components, including the Administration Console. This ensures that none of the HPBPI components are using the file before you copy it. When all the components are shut down, locate the `web-api.jar` file on the HPSD system and copy it to the following locations under your HPBPI installation directory:

- *hpbpi-install-dir*`\java`
- *hpbpi-install-dir*`\nonOV\jakarta-tomcat-5.0.19\webapps\ovbpidashboard2-10\WEB-INF\lib`

  The Business Process Dashboard needs to have been started following a new installation for this directory to be populated.
- *hpbpi-install-dir*`\examples\bia\BusinessProcessDashboard\WEB-INF\lib`

Automatic HPSD Service Mapping

HPBPI can integrate with HPSD by attempting to map the HPOM or OVIS services defined within the HPBPI Modeler to the same services defined within HPSD.

For this integration method to be successful:

- HPSD must have imported HPOM services from the same HPOM server as HPBPI (HPOM for Windows or OVSN).
- OVIS service level agreements (SLAs) must have been defined within HPSD and then exported from HPSD into OVIS.
- From the Service Desk Client, use the SLM option to add a new Service and then add the HPOM service name to the `SN Name` field.

If the HPOM or OVIS service is not created in this way, HPBPI is unable to display the HPSD data through the HPBPI Dashboard. If this is the case, you can define a custom field within HPSD in order to integrate HPBPI and HPSD. You also need to define custom fields if you want to report on Standalone services that have been defined within HPBPI. Section Defining HPSD Custom Fields on page 128 describes how to create a custom field for HPBPI, within HPSD, and the data that you need to enter in the custom field.

For details of how to configure HPBPI to interoperate with HPSD automatically, refer to the *HPBPI System Administration Guide*.

Defining HPSD Custom Fields

An alternative method of integrating HPBPI services with HPSD services is to define Custom Fields within HPSD. These Custom Fields can then be used to hold the names of HPBPI services. You need to do this if HPOM and OVIS services have not been created in HPSD as described in section Automatic HPSD Service Mapping on page 128.

You might also want to create custom fields to enable Service Call and Incident information to be reported on Standalone services that are defined within HPBPI.

HPBPI supports a number of HPSD custom fields that you can use to identify HPBPI services; these are HPSD Service fields. The Custom Fields that are available for you to use within HPSD are `Srv.Text1` through to `Srv.Text5`.

In order to use this method of integration, you first need to set up an appropriate custom field within HPSD. You can then select the custom field from the HPBPI Administration Console to complete the integration.

The following steps outline what you need to do within HPSD to configure a custom field for use with HPBPI:

1. Make sure that you are logged into HPSD from an account that has administrator permissions.
2. Start the Administrator Console and select `System...` from the `Tools` menu within HPSD
3. Navigate to the Custom Fields dialog. For example, select:

   `hp OpenView service desk > Data > Custom Fields`

   This is where you can select, rename and activate custom fields within HPSD
4. Open the `Service` option within the `Custom Fields` dialog.

   In order to integrate with HPBPI services, you need to define a custom field of the type `Text 255` for the Service option within HPSD.
5. From the `Service - Custom Fields` dialog, select one of the supported Fields that is currently not activated. This is one of `Srv.Text1` through to `Srv.Text5`.

   If all the fields are already activated, you need to find out if any are now redundant and can therefore be re-used for HPBPI. If none of the supported fields is available, this method of integration cannot be used.
6. If one of the Custom Fields for the Service is available, rename the field to something that is meaningful for your implementation, for example, HPBPI Service. The name needs to be meaningful to the HPSD user who will configure the Service within HPSD as part of the integration with HPBPI.
7. Check the `Activate` check box to enable to enable the custom field.

You have now completed the steps to create a new custom field within HPSD. The next stage is to add the new custom field to the HPSD Service dialog, so it is available to the user to add configuration data.

To add the new custom field to the Service dialog, complete the following steps:

1. Make sure that the Administration Console is active.
2. Navigate to the Forms Designer tool for services, for example:

   `hp OpenView service desk > Presentation > Forms > Service`

   The list of forms that can be modified is listed in the right-hand pane.
3. Open the Service forms designer window.

   The Form Designer dialog showing the current structure of the HPSD Service dialog is displayed.
4. Select the newly created custom field from the Attributes menu and drag it onto the Forms Designer dialog in the position where you want it to appear.
5. Save your changes using the `File|Save` option.

This method of updating the HPSD Service dialog updates all the Service dialogs within HPSD. You can be more selective when you modify forms; refer to the HP Service Desk documentation for full details of customizing HPSD forms.

When you have completed this configuration step, any HPSD service can be linked to an HPBPI Service using the custom field.

The following steps describe how you link an HPBPI Service using this newly created field:

1. From within the HPSD desktop, select the option to create a new service, for example:

   `File|New|Service`
2. From the `Choose Template` dialog, select the template that you want to use to create the service.
3. From the `New Service` dialog, complete the details of the new service that you are creating, and in the new custom field that you have created for the form, you need to enter the Service name for the HPBPI Service that you want HPSD to report on.

   The format of the HPBPI Service name entered within the HPSD must be the Service name as it appears within the HPBPI Modeler, excluding the HPBPI-specific prefix.

The following is an example of a Service Name hierarchy that might be defined within the HPBPI Modeler for an HPOM service:

```
Model Repository
    HPOM Services
      CRM Application
```

In this example, you enter `CRM Application` into the new custom field. You do not include the `HPOM Services` prefix.

The following is an example of a Service Name hierarchy that might be defined for OVIS:

```
Model Repository
    OVIS Services
      Insurance
          CRM Application
```

In this example, you enter `Insurance/CRM Application` into the new custom field. You do not include the `OVIS Services` prefix.

The following is an example of a Service Name hierarchy that might be defined for a Standalone Service:

```
Model Repository
    Standalone Services
          My Service
```

In this example, you enter `My Service` into the new custom field. You do not include the `Standalone Service` prefix.

4. Save the new HPSD service.

   This service is now in a form that can be accessed by HPBPI. When an HPSD service definition is created within HPBPI, and subsequently mapped to an HPSD service as defined in HPSD, the HPBPI Dashboard is automatically notified of the Incident report. As a result, you can link from the impacted Service to the Incident report through the Dashboard.

The final step is to configure HPBPI to integrate with HPSD using the newly created custom field.

When you set up the HPSD integration with HPBPI, you need to create an HPSD user account for HPBPI to use. You are strongly advised to create a user account specifically for HPBPI with the following characteristics:

- the user account should allow for concurrent users
- the user account should have the role `Helpdesk`

You provide details of this account when you set up the `HPSD Interoperability` through the HPBPI Administration Console or through the HPBPI installation.

A Database Schemas

This appendix details the database schemas that are defined to generate reports from the HPBPI impact data and for use by the Business Impact Engine, Metric Engine and Business Events Handler.

The schemas are described so you can use the information generated for your own reporting purposes. Do not modify any of the data in these database tables as HPBPI relies on the data being internally consistent in order to operate. Changing the values in the database tables will impact the behavior and operation of your HPBPI system.

A schema describes the logical structure of the HPBPI database and defines the tables, fields, indexes and views. It also shows the relationship between the fields and the tables.

The Flow schema, Business Event Handler schema and the Business Entity schema are optimized for the application processing. The Metrics schema is optimized for reporting and queries.

You can use the schemas to provide the information that you need to access flow data that you want to incorporate into your reports, or that you want to incorporate into a dashboard that you are developing.

This appendix describes the following schemas:

- Business Metrics schema, which is populated using data generated by the Metric Engine. This is historical data collected as a result of business process metrics that you configure using the Metric definer. Section Business Metrics Schema on page 136 describes this schema.
- Flow schema, which is a static schema created as a result of a flow being defined and subsequently deployed using the HPBPI Modeler; see section Flow Schema on page 172.

- Business Entity Schema, which is directly affected by the users' modeling and reflects the business entities modeled by the user; see section Business Entity Schema on page 191.
- Business Event Handler schema, which describes the Event Hospital and the hospital tables in the database. Out-of-Sequence events and events that contain errors, or that are corrupt, are placed in the Event Hospital. Section Business Event Handler Schemas on page 195 describes this schema.

Building Applications to Use the HPBPI Metrics Data with Microsoft SQL Server

You can use the data in the HPBPI schemas in your own reporting applications, for example, for a custom-built business dashboard.

If you intend to use Microsoft SQL Server and write SQL select statements that access more than one database table, you are advised to set the priority for database access to be low for applications accessing this data using the following SQL statement:

```
set deadlock_priority low;
```

You also need to design your application to retry if it fails to access the data.

This prevents deadlock situations occurring in cases where an application and the Business Impact Engine are accessing database tables at the same time. In this case, the application might initially fail to access the data; however, it will subsequently retry.

By setting the access priority to low for applications, the Business Impact Engine has priority and its performance is not impacted by applications that require only read-access to the data.

Oracle and SQL Server Data Type Definitions

The following are the data type definitions referenced in the following database tables.

Table 7 Oracle and SQL Data Types

Descriptive Type	Oracle Type	SQL Server Type	Description
string	varchar2	nvarchar	Variable length character field.
date[a]	date	datetime	Fixed-length data and time field.
integer	Number (10)	int	Single-length integer.
long	Number (20)	bigint	Double-length integer.
text	clob	text	variable-length single-byte character data.
float	float	float	Real Number.
currency	number	decimal	decimal number with a fixed number of decimal places.
variable-length text	clob	ntext	variable-length text up to 1GB (`ntext`) and 4GB (`clob`).
variable-length binary	blob	image	variable-length binary file up to 2GB (`image`) and 4GB (`blob`).

a. For Microsoft SQL Server, the these times are recorded to the nearest millisecond. In the case of an Oracle Server, these times are recorded to the nearest second. This can lead to a slight loss of precision when some values are displayed; for example, metric values from the Business Metrics Schema.

Business Metrics Schema

This section describes the schema for the business process metrics defined using the Business Process Metrics definer. This is historical data collected as a result of your configuration using the Metrics definer. You can show this data through Business Process Dashboard, or using a reporting application of your choice, for example, Crystal Reports or Microsoft Access.

The Business Metrics schema is based on a dimensional model and the Flow schema is based on an entity-relationship model. This means the Business Metrics schema is designed to maximize the effectiveness of user and application queries. The Flow Schema, which is described in section Flow Schema on page 172, is optimized for application processing and is indexed for this purpose.

Figure 30, Figure 31 and Figure 32 show the dimensional diagrams for the HPBPI business process metrics. In each case, there is a central table, known as the fact table, plus a number of dimension tables. The fact table comprises all the measurements or data that are required for a specific aspect of the metric thresholds. The dimension tables, which are much smaller, comprise descriptive fields. These fields are meaningful when the row headers are presented through user interfaces and reporting applications. The dimension tables are also smaller in order that browse queries on the tables can be completed without delay to the user or application requesting the information.

The metrics data is also used by the Notification Server when generating email alerts relating to threshold violations.

The following sections describe the components that make up the Business Metric schema. The data types listed in the tables are the SQL data types, Table Oracle and SQL Data Types on page 135 describes the SQL Server and Oracle data types mappings and their descriptions.

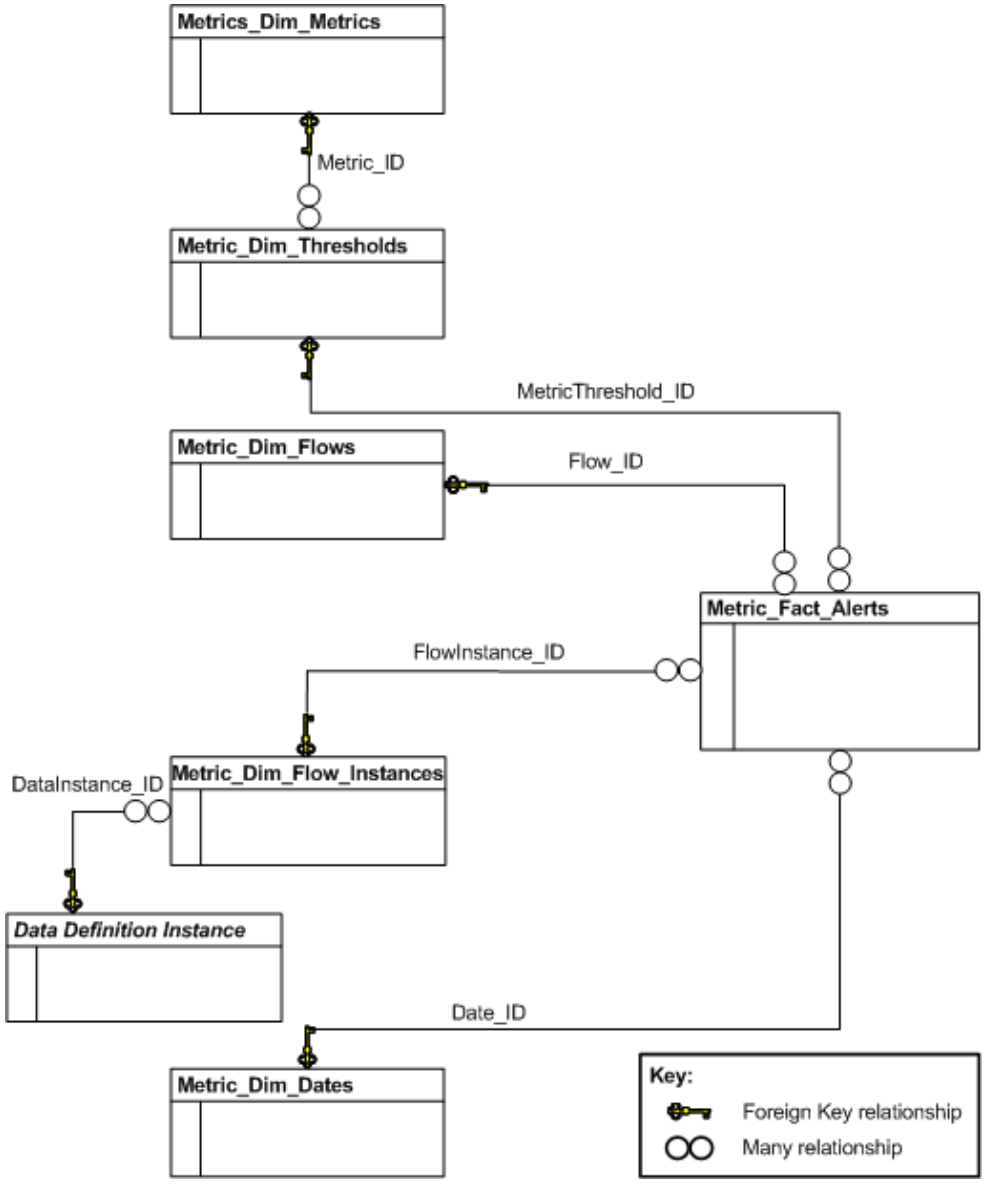
All Data Types of `date` in the Metric schema are specified as local time.

Alerts Facts

This dimension provides information relating to the alerts that are generated when a particular threshold that you have defined is violated. The table describing the facts relating to the alerts, contains information for both instance-level alerts and the statistical alerts.

Figure 30 shows the structure of the Metric_Fact_Alerts dimension model and also the dimension tables that are related to it.

Figure 30 Fact Alerts Dimensional Model



In Figure 30, the Data Definition Instance table cannot be identified until after the Data definition is deployed, when the table is created and named; refer to section Data Definition Instances on page 193 for details of how to identify this table.

Business Process Metric Fact Alerts

The Metric_Fact_Alerts table describes the data that is defined for HPBPI alerts that relate to the threshold violations for all business process metrics. The dimension tables shown in Figure 30 are described in section Dimension Tables on page 151.

The primary key is MetricAlert_ID, which is a system-assigned unique identifier.

Table 8 Metric_Fact_Alerts

Column Name	Data Type	Length	Allow Nulls	Description
MetricAlert_ID	string	36	No	Primary Key[a].
MetricThreshold_ID	string	36	No	Foreign Key[b] to table Metric_Dim_Thresholds.
Flow_ID	string	36	No	Foreign Key[2] to table Metric_Dim_Flows.
Date_ID	string	36	No	Foreign Key[2] to table Metric_Dim_Dates.
FlowInstance_ID	string	36	No	Foreign Key[2] to table Metric_Dim_Flow_Instances.
Time	date		Yes	The time (in date format) that the threshold alert occurred. Specified in local time.
TimeLongMillis	long		Yes	The time (in milliseconds) that the threshold alert occurred. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)

Table 8 Metric_Fact_Alerts

Column Name	Data Type	Length	Allow Nulls	Description
RaisedTime	date		Yes	The time (in date format) that the threshold alert was raised. Specified in local time.
RaisedTimeLong Millis	long		Yes	The time (in milliseconds) that the threshold alert was raised. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.).
AlertStatus	string	12	Yes	The status of the metric, and can be `Critical`, `Major`, `Minor`, `Warning` and `Normal`. These status correspond to the HP Operations Manager severity levels.
AlertLevel	integer		Yes	Numeric equivalent of AlertStatus as follows: • 1 = `Normal` • 2 = `Warning` • 3 = `Minor` • 4 = `Major` • 5 = `Critical`
Value	float		Yes	Metric value or statistic at the time that the alert is generated. The value for this column varies according to the type of threshold. In the case of a Deadline alert, it is the time in seconds since the Deadline was overdue.
IsAcknowledged	integer		Yes	For internal use.

a. This is the column that uniquely identifies rows.

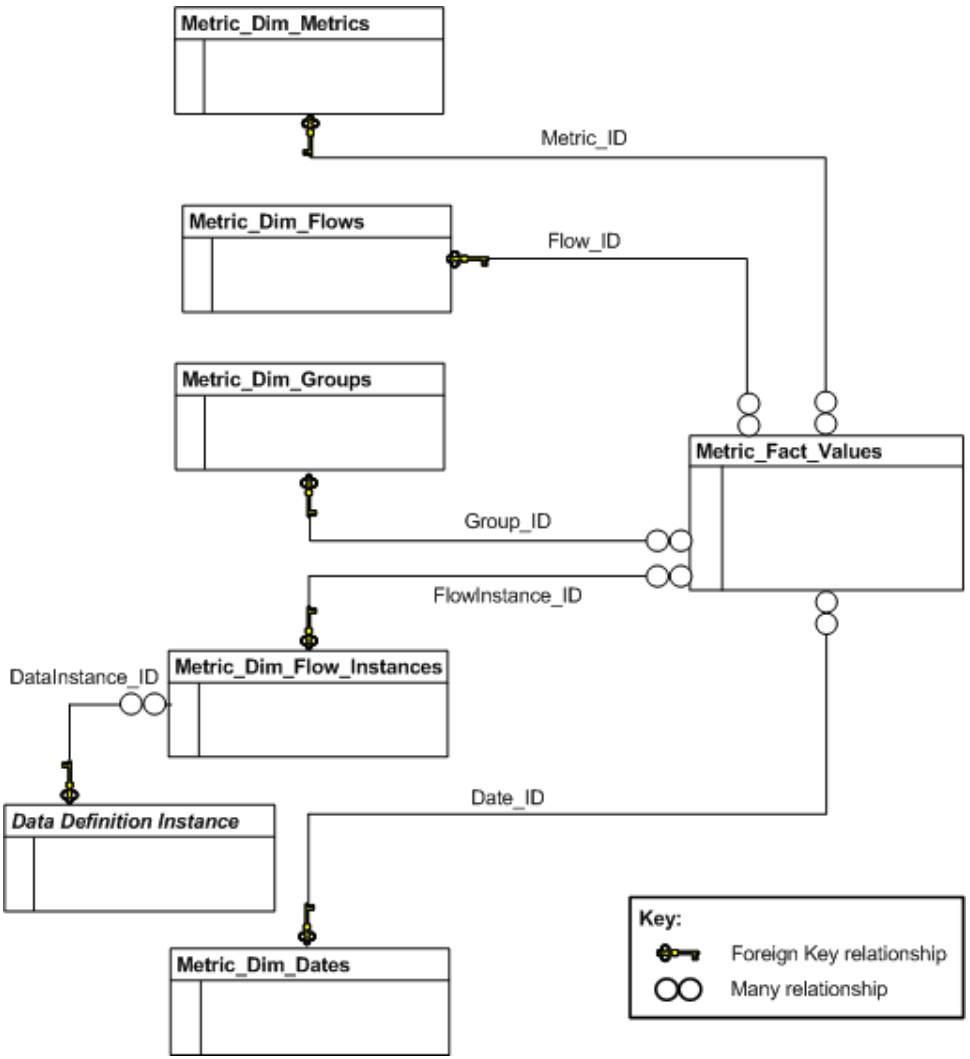
b. This is the column that exists as the primary key in another table.

Facts Values

This dimension provides information relating to the data collected for the thresholds that you define in the Metrics Definer.

Figure 31 shows the structure of the Fact Values dimension model and also the dimension tables that are related to it.

Figure 31 Fact Values Dimensional Model



In Figure 31, the Data Definition Instance table can not be identified until after the Data definition is deployed, when the table is created and named; refer to section Data Definition Instances on page 193 for details of how to identify this table.

Business Metric Fact Value Dimensions

The Metric_Fact_Values table describes the data that is defined for HPBPI threshold values for the business process metrics that you define. The dimension tables shown in Figure 31 are described in section Dimension Tables on page 151.

The fact values table is linked to the metric name and metric type through the Metric_ID column.

The primary key is MetricValue_ID, which is a system-assigned unique identifier.

Table 9 Metric_Fact_Values

Column Name	Data Type	Length	Allow Nulls	Description
MetricValue_ID	string	36	No	Primary Key[a].
Metric_ID	string	36	No	Foreign Key[b] to table Metric_Dim_Metrics.
Flow_ID	string	36	No	Foreign Key[2] to table Metric_Dim_Flows.
Date_ID	string	36	No	Foreign Key[2] to table Metric_Dim_Dates.
FlowInstance_ID	string	36	No	Foreign Key[2] to table Metric_Dim_Flow_Instances.

Table 9 Metric_Fact_Values

Column Name	Data Type	Length	Allow Nulls	Description
Status	string	12	Yes	The status of the metric, which can be one of: • `Active` - the start condition for the metric has been met, but the end condition has not been met. • `Completed` - the metric has completed and has a value for reporting. • `NoStart` - the metric has completed, but does not have a value. • `Aborted` - the flow has completed before the metric completes.
TimeCompleted	date		Yes	The time (in date format) that the metric completes. Specified in local time.
TimeCompleted LongMillis	long		Yes	The time (in milliseconds) that the metric completes. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.).
Weight	float		Yes	This column provides the data that you can sort and use to provide a weighting for how important the metric is. This column contains the value of the `Data Definition` property, nominated to be the Weight property, in the HPBPI Modeler.

Table 9 Metric_Fact_Values

Column Name	Data Type	Length	Allow Nulls	Description
Deadline	date		Yes	Null, unless the metric has a deadline data item defined, in which case, it contains the value of the deadline. Specified in local time.
DeadlineLong Millis	long		Yes	Null, unless the metric has a deadline data item defined, in which case, it contains the value of the deadline in milliseconds. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
Group_ID	string	36	Yes	Foreign Key[2] to table `Metric_Dim_Groups`. If the metric is not part of a group the value is `Null`.
Value	float		Yes	Value of the metric. The units are times for duration-based metrics. In the case of custom metrics, the units are whatever you define; for example, it could be the value of a percentage. In the case of the `Single Node` scope metric, the value is equal to the difference between the end time and the start time. If the metric is not complete, that is, there is a start time, but no end time, the value for the metric is `Null`.

Table 9 Metric_Fact_Values

Column Name	Data Type	Length	Allow Nulls	Description
Idx	integer		Yes	The order of the Metric conditions, where the conditions are met more than once by a flow instance. The index starts at zero (0) and increments one (1) for every node instance that meets its complete conditions.
StartTime	date		Yes	Time (in date format) when the metric met its start condition. Specified in local time.
StartTimeLong Millis	long		Yes	Time (in milliseconds) when the metric met its start condition. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)

Table 9 Metric_Fact_Values

Column Name	Data Type	Length	Allow Nulls	Description
LastUpdateTime	date		Yes	This column is set each time a change is made to this table for this instance of the metric. It is the time (in date format) that a table row was last updated for the metric instance. Specified in local time.
LastUpdateTime LongMillis	long		Yes	This column is set each time a change is made to this table for this instance of the metric. It is the time (in milliseconds) that a table row was last updated for the metric instance. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.).
DeadlineOverdue	float		Yes	Applicable only when a Deadline threshold is set for the metric instance. When the metric is complete, this is the number of seconds outside the deadline recorded for the metric instance. This can be a positive or a negative number; a positive value is over the deadline, a negative value is under the deadline.

a. This is the column that uniquely identifies rows.

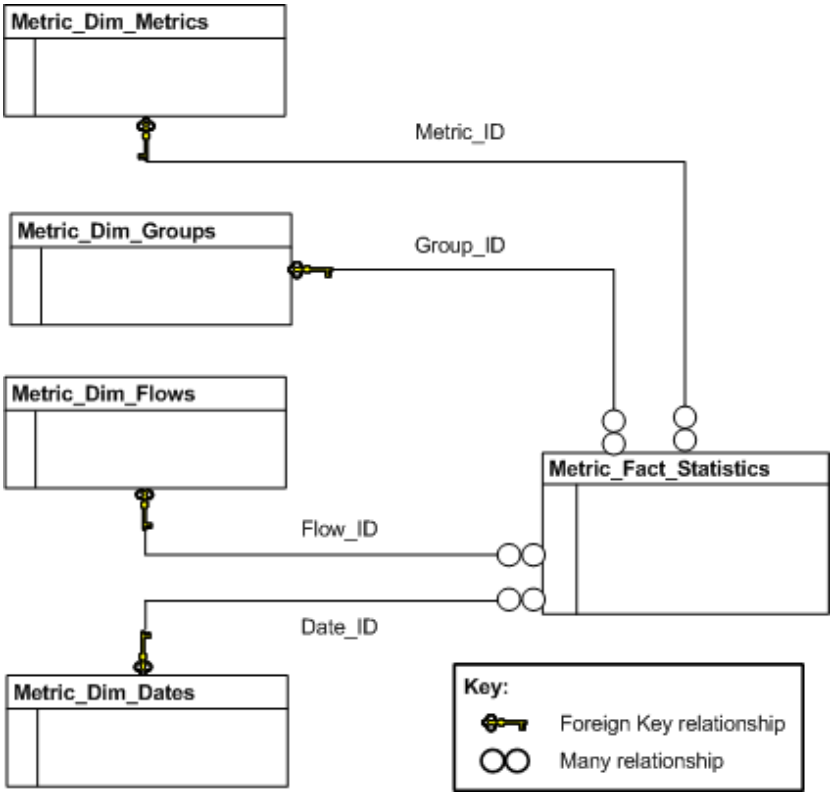
b. This is the column that exists as the primary key in another table.

Statistics Facts

This dimension provides information relating to the historical data that is recorded for the thresholds that you define in the Metrics Definer.

Figure 32 shows the structure of the Fact Statistics dimension model and also the dimension tables that are related to it.

Figure 32 Fact Statistics Dimensional Model



Business Metric Fact Statistics Dimensions

The Metric_Fact_Statistics table describes the historical data that is defined for HPBPI threshold values related to the metrics that you define. The dimension tables shown in Figure 32 are described in section Dimension Tables on page 151.

There can be multiple rows in this table for a given metric in order to collect statistics on active instances and recently completed instances.

For each set of statistics, data is recorded at time periods that are complete multiples of the value of MeasurementPeriod. As an example, if the measurement period is 10 minutes, the statistics are recorded for the hour, and then in periods of 10 minutes (10, 20 30, 40, 50) after the hour.

If the value of MeasurementPeriod is 24 hours, statistics are recorded at 00:00:00 each day (local time). Be aware that when daylight saving occurs, this period can be 23 hours or 25 hours according to the direction of the change.

The primary key is MetricStatistic_ID, which is a system-assigned unique identifier.

Table 10 Metric_Fact_Statistics

Column Name	Data Type	Length	Allow Nulls	Description
MetricStatistic_ID	string	36	No	Primary Key[a].
Metric_ID	string	36	No	Foreign Key[b] to table Metric_Dim_Metrics.
Flow_ID	string	36	No	Foreign Key[2] to table Metric_Dim_Flows.
Date_ID	string	36	Yes	Foreign Key[2] to table Metric_Dim_Dates.
Time	date		Yes	Time (in date format) that the metric was last updated. Specified in local time.

Table 10 Metric_Fact_Statistics

Column Name	Data Type	Length	Allow Nulls	Description
TimeLongMillis	long		Yes	Time (in milliseconds) that the metric was last updated. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.).
IsLatest	integer		Yes	Indicates whether or not this is the latest statistic. A numeric value of 1 indicates this is the latest statistic.
StatisticsType	string	12	Yes	The type of statistic: • `Active` - all currently active metric instances that relate to this statistic. • `Completed` - all currently completed metric instances that relate to this statistic. • `Total` - the total, since the metric was defined, of all the metrics for the metric instances that have completed.
Measurement Period	long		Yes	The period (in seconds), specified in the Metric Definer, for the metric to be measured over. In the case of total metric instances, this is the period (in seconds) from the first set of metric statistics.
Group_ID	string	36	Yes	Foreign Key[2] to table `Metric_Dim_Groups`. If the metric is not part of a group, the value is Null.

Table 10 Metric_Fact_Statistics

Column Name	Data Type	Length	Allow Nulls	Description
CountTotal	long		Yes	Number of metric instances to be included in the calculation for the metric statistics; for example, in the case of flow instance metrics this is the number of flow instances.
Throughput	float		Yes	Applies only to completed metrics. The number of metric instances per hour. This is calculated as follows: `(CountTotal * 36,000)/ MeasurementPeriod`
SumValue	float		Yes	Total for the metric instance values.
SumSquareValue	float		Yes	Total of the square of the metric instance values.
AverageValue	float		Yes	Average for the metric instance values,
StdDevValue	float		Yes	Standard deviation for the metric instance values.
MinimumValue	float		Yes	The minimum of the metric instance values.
MaximumValue	float		Yes	The maximum of the metric instance values.
WeightTotal	float		Yes	Total of the Flow instance Weight attribute for the metric; for example, in the case of an Order Flow instance, this might be the total value of the orders (a historical total).

Table 10 Metric_Fact_Statistics

Column Name	Data Type	Length	Allow Nulls	Description
Weight Throughput	float		Yes	This applies only to Completed metrics. This is number of instance per hour that relate to this metric. The value is calculated as follows: `(WeightTotal * 36,000) / MeasuremenPeriod`
WeightSumValue	float		Yes	Total of metric instance values * Flow instance weight
WeightSum SquareValue	float		Yes	Total of the square of the metric instance values * Flow instance weight
WeightAverage Value	float		Yes	Average of metric instance values weighted by flow instances that have a higher weight than the average.
WeightStdDev Value	float		Yes	Standard deviation for the metric instance values weighted by flow instances that have a higher weight than the average.

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

Dimension Tables

The following tables describe the business dimensions defined for the fact tables described above.

Business Metrics Dimensions

The Metric_Dim_Metrics table is the dimensional table that you can use for creating reports from business process metric information collected by the Business Impact Engine. This is a dimension on the Metric_Fact_Alerts table, Metric_Fact_Statistics table and Metric_Fact_value table.

The primary key is Metric_ID, which is a system-assigned unique identifier.

Flow_ID is a foreign key used to link to the Flow table, which identifies the flow definition for the business process metric.

Table 11 Metric_Dim_Metrics

Column Name	Data Type	Length	Allow Nulls	Description
Metric_ID	string	36	No	Primary Key[a].
MetricName	string	120	No	Name given to a business process metric through the Business Process Metric definer.
Flow_ID	string	36	No	Foreign Key[b] to the Flows table; see Table 20 on page 173.
ValueUnits	string	12	Yes	Specifies the units for the business process metric type; for example, seconds. In the case of a custom metric, you set this to the units for the metric that you are creating; for example, percentage. It is the string to represent the unit in the user interface.

Table 11 Metric_Dim_Metrics

Column Name	Data Type	Length	Allow Nulls	Description
CreatedDate	date		Yes	Time and date when the metric is created. Specified in local time.
Measurement Period	long		Yes	The period, in units of seconds, that the business process metric.
GroupName	string	120	Yes	Name used for grouping statistics.
LastStatistics RecordTime	date		Yes	Time and date for the last statistics report. This is used to work out when to start recording historical information. Specified in local time.
LastStatistics RecordLongMillis	long		Yes	Time and date (in milliseconds) for the last statistics report. This is used to work out when to start recording historical information. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.).
IsDeleted	integer		Yes	Indicates whether or not a metric has been deleted. A setting of one (1) means the metric is deleted.

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

Business Metrics Date Dimensions

The Metric_Dim_Dates table is the dimensional table that you can use for creating reports that contain time dimensions for the Metric_Fact_Alerts table, Metric_Fact_Statistics table and Metric_Fact_value table.

The primary key is Date_ID, which is a system-assigned unique identifier for the date and time.

Table 12 Metric_Dim_Dates

Column Name	Data Type	Length	Allow Nulls	Description
Date_ID	string	36	No	Primary Key[a]
Year	integer		No	The `year` for the time, for example, 2005.
Quarter	integer		No	A number representing a three month period for each quarter of a year as follows: • 1 = January to March • 2 = April to June • 3 = July to September • 4 = October to December
Month	string	30	No	The month for the time, specified as a string, in the configured locale for the database.
Month_Num	integer		No	The `month` for the time, represented as a numeral between 1 and 12.
Day	integer		No	The `day` for the time, represented as a numeral between 1 and 31.
Week	integer		No	The week in the year for the time, represented as a numeral between 1 and 53.
DayOfWeek	string	30	No	The day of the week, specified as a string, in the configured locale for the database.

Table 12 Metric_Dim_Dates

Column Name	Data Type	Length	Allow Nulls	Description
DayOfWeek_Num	integer		No	The day of the week for the time, represented as a numeral between 1 and 7 as follows: • 1 = Sunday • 2 = Monday • 3 = Tuesday • 4 = Wednesday • 5 = Thursday • 6 = Friday • 7 = Saturday
Hour	integer		No	The `hour` for the time, represented as a numeral between 0 and 23.
Minute	integer		No	The `minute` for the time, represented as a numeral between 0 and 59.
Time	date		No	The date and time, rounded down to the nearest minute, in date format as supported by your database. Specified in local time.

a. This is the column that uniquely identifies rows.

Business Metric Flows Dimensions

The Metric_Dim_Flows table is the dimensional table that represents the flows that you want to report thresholds for. This is a dimension on the Metric_Fact_Alerts table, Metric_Fact_Statistics table and Metric_Fact_value table.

The primary key is Flow_ID, which is a system-assigned unique identifier.

Table 13 Metric_Dim_Flows

Column Name	Data Type	Length	Allow Nulls	Description
Flow_ID	string	36	No	Primary Key[a].
FlowName	string	120	No	The name of the business flow as entered in the HPBPI Modeler.
Status	string	12	Yes	The current status of this flow, which can be one of `Active`, `Impeded`, `Blocked` or `Deleted`, where: • `Active` means the flow has been deployed. • `Impeded` means that at least one instance of the flow is in the `Impeded` state; see `Status` in Table 21 on page 175. • `Blocked` means that at least one instance of the flow is in the `Blocked` state; see `Status` in Table 21 on page 175. • `Deleted` means the flow has been undeployed.
DataDefinition_ID	string	36	Yes	Primary entity identifier for the Data definition associated with the Flow.

a. This is the column that uniquely identifies rows.

Business Metrics Flows Instance Dimensions

The Metric_Dim_Flow_Instances table is the dimensional table that represents the flow instances that you want to report thresholds for. This is a dimension on the Metric_Fact_Alerts table and Metric_Fact_value table.

The primary key is FlowInstance_ID, which is a system-assigned identifier, which is unique to the flow instance.

Table 14 Metric_Dim_Flow_Instances

Column Name	Data Type	Length	Allow Nulls	Description
FlowInstance_ID	string	36	No	Primary Key[a].
Flow_ID	string	36	No	Foreign Key[b] to table Flows (Flow_ID) column.
Identifier	string	120	Yes	The identity of the related Data definition for this Flow. When you create a flow using the HPBPI Modeler, you can enter the name of a property of a related Data Definition to be used as an Identity property. This column holds the value of the Identity property for the specific flow instance.
DataDefinition_ID	string	36	Yes	A unique identifier (GUID) for the business object class associated with this metric instance; this column is taken from `Model_ID` in the Data Object table. An example of a `Model_ID` is an Order Definition.
DataInstance_ID	string	36	Yes	A unique identifier (GUID) for the business object instance associated with this metric instance
Weight	float		Yes	The value of the Weight for the flow instance.

Table 14 Metric_Dim_Flow_Instances

Column Name	Data Type	Length	Allow Nulls	Description
StartTime	date		Yes	The time that this flow instance was last started (in date format). Specified in local time.
StartTimeLong Millis	long		Yes	The time (in milliseconds) that this flow instance was last started (in date format). Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.).
EndTime	date		Yes	The time that this flow instance was last completed (in date format). Specified in local time.

Table 14 Metric_Dim_Flow_Instances

Column Name	Data Type	Length	Allow Nulls	Description
EndTimeLong Millis	long		Yes	The time (in milliseconds) that this flow instance was last completed (in date format). Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
Status	string	12	Yes	The current status of the flow instance for this metric, which can be one of `Active` or `Completed`, where: • `Active` means the flow instance has met the start condition for at least one node in the flow. • `Completed` means the flow instance has met the complete conditions for one of the end nodes in the flow. The Metric Engine does not report on the interim status of flow instances (impeded or blocked). For the purpose of the metrics data, all flow instances are active until they have completed.

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

Business Metric Group Dimensions

The Metric_Dim_Groups table is the dimensional table that represents the groups that you might have defined for your metrics in the Metrics Definer. This is a dimension on the Metric_Fact_Statistics table and Metric_Fact_value table.

The primary key is Group_ID, which is a system-assigned unique identifier.

Table 15 Metric_Dim_Groups

Column Name	Data Type	Length	Allow Nulls	Description
Group_ID	string	36	No	Primary Key[a].
GroupName	string	120	No	The name of the group that this metric belongs to; for example, you might have a group called Terminals that represents airport terminals.
GroupValue	string	256	Yes	The values of the groups identified in GroupName; for example, in the case of Terminals, you might have the values 1, 2 and 3, for Terminals 1, 2 and 3.

a. This is the column that uniquely identifies rows.

Business Metric Threshold Dimensions

The Metric_Dim_Thresholds table is the dimensional table that represents the thresholds defined for your metrics. When a threshold for an instance is violated, an alert is generated for each instance violated. When a threshold for a statistic is violated, an alert is generated each time the violation level changes for the statistic.

This is a dimension on the Metric_Fact_Alerts table.

When using this table be aware that the units used to define the alert levels within the Metric Definer (for example, AlertCriticalLevelDefnUnit) are not necessarily the same as the units defined for the thresholds levels (for example, AlertCriticalLevel).

The primary key is MetricThreshold_ID, which is a system-assigned unique identifier.

Table 16 Metric_Dim_Thresholds

Column Name	Data Type	Length	Allow Nulls	Description
MetricThreshold_ ID	string	36	No	Primary Key[a].
Metric_ID	string	36	No	Foreign Key[b] to table Metric_Dim_Metrics.
ThresholdName	string	120	No	Name of the threshold, entered when the Business Process Threshold is defined. It is used within the Metric Definer, the Business Process Dashboard and within email alerts as the display name for the Business Process Threshold.
Threshold Description	string	256	Yes	Description of the threshold as entered in the Metric Definer.
ThresholdMessage	string	256	Yes	A message that can be included in an email notification to provide additional information and context about the alert for the recipient.

Table 16 Metric_Dim_Thresholds

Column Name	Data Type	Length	Allow Nulls	Description
ThresholdType	string	12	No	Indicates whether or not the threshold is set for a particular instance, or for a particular statistics type as follows: • `Instance` - set for a particular instance • `Active` - set for active instances • `Completed` - set for instances that have completed during the metric period • `Total` - set for all instances that have completed since the metric was defined.
ThresholdColumn Name	string	40	Yes	Defines the column that the business metric is set for as follows: • `CountTotal` is the number of instances • `Throughput` is throughput • `AverageValue` is the average • `MinimumValue` is a minimum value • `MaximumValue` is the maximum value • `WeightTotal` is a weight total • `WeightThroughput` is a weight throughput • `WeightAverageValue` is a weight average `ThresholdColumnName` is set to `Null` when the `ThresholdType` is `Instance`.

Table 16 Metric_Dim_Thresholds

Column Name	Data Type	Length	Allow Nulls	Description
ThresholdTest	string	12	No	Defines the type of test applied and the meaning of the alert level thresholds. With the exception of `Deadline`, applies test to compare the most recent threshold data collected to `Value` column of the metric_fact_values table: • `OverValue` - value is equal to or more than value specified in the alert level. • `UnderValue` - value is less than or equal to value specified in the alert level. • `OverUsual` - number of standard deviations that the value is equal to, or above the average for this metric, (applicable to instance, average and weighted average only) • `UnderUsual` - number of standard deviations that the value is equal to, or below the average for this metric, (applicable to instance, average and weighted average only) • `Unusual` - the value of the metric is either above or below the average for this metric (either the `OverUsual` or `UnderUsual` conditions are met). • `Deadline` - the metric has completed within the required time (value in the `Deadline` column in the metric_fact_values table). This is applicable only to instance thresholds.

Table 16 Metric_Dim_Thresholds

Column Name	Data Type	Length	Allow Nulls	Description
AlertCriticalLevel	float		Yes	Critical threshold level, in seconds, or in the case of custom metrics the unit you assigned to `Value` in the Metric_Fact_Values table.
AlertMajorLevel	float		Yes	Major threshold level, in seconds, or in the case of custom metrics the unit you assigned to `Value` in the Metric_Fact_Values table.
AlertMinorLevel	float		Yes	Minor threshold level, in seconds, or in the case of custom metrics the unit you assigned to `Value` in the Metric_Fact_Values table.
AlertWarning Level	float		Yes	Warning threshold level, in seconds, or in the case of custom metrics the unit you assigned to `Value` in the Metric_Fact_Values table.
LastCheckTime	date		Yes	Last time (in date format) that the threshold was checked by the Metric Engine. Specified in local time.
LastCheckTime LongMillis	long		Yes	Last time (in milliseconds) that the threshold was checked by the Metric Engine. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.).

Table 16 Metric_Dim_Thresholds

Column Name	Data Type	Length	Allow Nulls	Description
CurrentAlert Status	string	12	Yes	The highest level recorded for the alert status of a metric instance within the current Collection interval, and can be `Critical`, `Major`, `Minor`, `Warning` and `Normal`. These status correspond to the HP Operations Manager severity levels.
CurrentAlertLevel	integer		Yes	Numeric equivalent of CurrentAlertStatus as follows: • 1 = `Normal` • 2 = `Warning` • 3 = `Minor` • 4 = `Major` • 5 = `Critical`
CurrentAlert FromTime	date		Yes	The start date and time (in date format) of the start of the current Collection interval for the CurrentAlertLevel and CurrentAlertStatus values. If the Collection interval is zero (not set), then this is the data and time of the start of the current threshold polling interval. Specified in local time.
CurrentAlertFrom TimeLongMillis	long		Yes	Same as `CurrentAlertFromTime` in milliseconds rather than date format. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)

Table 16 Metric_Dim_Thresholds

Column Name	Data Type	Length	Allow Nulls	Description
StagingAlertLevel	integer		Yes	The highest level recorded for the alert status of a metric instance since the end of the current Collection interval. Used internally as an ongoing record of the metric instance alert status in readiness for the next Collection interval.
AlertCriticalLevel DefnUnit	string	12	Yes	The unit used to define the Critical Alert value for the Metric Threshold within the Metric Definer, and as defined in the Metric Definer. These are the display units used for the user interface.
AlertMajorLevel DefnUnit	string	12	Yes	The unit used to define the Major Alert value for the Metric Threshold within the Metric Definer, and as defined in the Metric Definer. These are the display units used for the user interface.
AlertMinorLevel DefnUnit	string	12	Yes	The unit used to define the Minor Alert value for the Metric Threshold within the Metric Definer, and as defined in the Metric Definer. These are the display units used for the user interface.

Table 16 Metric_Dim_Thresholds

Column Name	Data Type	Length	Allow Nulls	Description
AlertWarning LevelDefnUnit	string	12	Yes	The unit used to define the Warning Alert value for the Metric Threshold within the Metric Definer, and as defined in the Metric Definer. These are the display units used for the user interface.
CreatedDate	date		Yes	Time and date when the threshold is created. Specified in local time.

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

Be aware that the `ThresholdTest` value `Deadline` has the following behavior:

- In the case of metrics for Node instances, an alert is generated if the node is not completed for cases where the node is not started.
- The alert levels are the number of seconds before the deadline occurs, for example, it is possible to have a warning alert (`AlertWarningLevel`) at four hours before the deadline and a critical alert (`AlertCriticalLevel`) when the deadline is reached.

Business Metric Custom Types

The Metrics Definer enables you to define your own custom metrics.

The Metric_CustomTypes table is the dimensional table that represents the custom metrics that you can define and are described in Table 17 on page 167. The Metrics Definer uses this table when presenting the custom metrics that you can select.

The primary key is CustomMetricName, which is the unique name that is given to the custom metric when you define it.

Table 17 Metric_CustomTypes

Column Name	Data Type	Length	Allow Nulls	Description
CustomMetric Name	string	120	No	Primary Key[a]. This is the name that appears in the Metric definer as the `Metric value type` identifier.
CustomMetric Description	string	256	Yes	Description for the custom metric.
CustomSPName	string	120	No	Name of the stored procedure associated with this custom metric.
ValueUnits	string	12	Yes	Specifies the units for the custom-metric type. This string can be shown through your presentation interfaces (for example a reporting tool that you use).

a. This is the column that uniquely identifies rows.

Defining custom metrics is described in more detail in the *HP Business Process Insight Integration Training Guide - Modeling Flows*.

Business Metric Failure Messages

All SQL errors generated by the custom metrics that you define are propagated to the Business Impact Engine, which logs an errors directly into its log file.

Always refer to the Business Impact Engine log file when you want to identify errors with the custom metrics that you define.

Metric Views

There are two additional database tables that are defined for compatibility with the metrics tables provided in previous versions of HPBPI. These are:

- Metrics view (Table 18 on page 168)
- Metrics_Value view (Table 19 on page 169)

Table 18 on page 168 is a view onto the Metrics_Dim_Metrics table described in section Business Metrics Dimensions on page 151.

Table 18 Metrics

Column Name	Data Type	Length	Allow Nulls	Description
MetricName	string	40	No	Primary Key[a] and the name given to the metric through the HPBPI Modeler.
MetricType	string	6	Yes	The type of metric, for example: • NET (Single Node scope) • TBN (Multiple Node scope) • FET (Whole Flow scope) • CUSTOM (custom-written metric)
Flow_ID	string	36	No	Primary Key[1] and Foreign Key[b] to the Flows table; Table 21 on page 175.
StartNode	string	36	Yes	In the case of NET and TBN, this is a unique identifier of the node where the metric data collection is to be started. In the case of FET, `StartNode` is null. In the case of CUSTOM, `StartNode` is either a unique identifier, or null, depending on whether it is for one or more nodes within the flow (unique identifier), or a whole flow (null).

Table 18 Metrics

Column Name	Data Type	Length	Allow Nulls	Description
StartCondition	integer		Yes	For internal use.
EndNode	string	36	Yes	In the case of NET and TBN, this is a unique identifier of the node where the metric data collection is to be stopped. In the case of FET, `StartNode` is null. In the case of CUSTOM, `EndNode` is either a unique identifier, or null, depending on whether it is for one or more nodes within the flow (unique identifier), or a whole flow (null).
EndCondition	integer		Yes	For internal use.

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

Metric Values

The Metric_Values view holds the details of the metrics data for flow instances. There is an entry for each flow instance for each metric in the flow.

Table 19 Metric_Values

Column Name	Data Type	Length	Allow Nulls	Description
Name	string	40	No	Metric name.
Type	string	6	Yes	The type of metric, for example: • NET (Single Node scope) • TBN (Multiple Node scope) • FET (Whole Flow scope) • CUSTOM (custom-written metric)

Table 19 Metric_Values

Column Name	Data Type	Length	Allow Nulls	Description
Flow_Instance	string	36	Yes	Foreign Key[a] to Flow_Instance table; see Table 21 on page 175.
StartTime	date		Yes	Time when flow instance met the Start condition for `StartNode` (in date format); see Table 18 on page 168 for a description of `StartNode`. Specified in local time.
StartTimeLong Millis	long		Yes	Time (in milliseconds) when flow instance met the Start condition for `StartNode`; see Table 18 on page 168. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
EndTime	date		Yes	Time when flow instance met the complete condition for `EndNode` (in date format); see Table 18 on page 168. Specified in local time.
EndTimeLong Millis	long		Yes	Time (in milliseconds) when flow instance met the complete condition for `EndNode`; see Table 18 on page 168. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
Value	float		Yes	For internal use.

Table 19 Metric_Values

Column Name	Data Type	Length	Allow Nulls	Description
Time	date		Yes	The time that the metric was updated. Specified in local time.
Idx	integer		Yes	Index or counter for an instance where the metric conditions are met multiple times by one flow instance, for example, when the instance is in a loop condition within the flow.

a. This is the column that exists as the primary key in another table

Flow Schema

This section describes the schema for the business flows defined using the HPBPI Modeler; this data is used by the Business Impact Engine to process the impact data collected from the flows that you define.

If you want to report measurements for your flows, use the business process metrics data held in the metrics schema as described in section Business Metrics Schema on page 136. These business process metric tables hold all the measurement data that you have requested is to be recorded and are designed to be effective for queries and searches. The Flow schema is an entity-relationship schema and is designed for transaction processing. The Flow tables hold all the data collected for the Flows and Flow instances that you have defined and deployed using the HPBPI Modeler. Use this information to show details of the Flow through the Dashboard.

If you have configured an Oracle Server to store your Flow Schema data, the length of some of the fields in the Flow Schema tables is three-times the length shown. The fields affected are user defined labels, such as names. Fields that are internally generated, such as identifiers are not impacted. This allows for the fact that UTF8 can be used as the database character set, in which case, up to three bytes might be required to store each character.

As an example, the length of the `FlowName` column is 120 rather than 40.

As the flow data stored in the database represents a dynamic data model, the schema is designed with performance as the design goal, rather than reporting techniques. To improve performance, database indexes, which improve the rate of access to the database tables, are defined for both Oracle and SQL Server.

The database objects for the Flow schema are shown as an entity-relationship diagram; see Figure Flow Schema Entity-Relationship Diagram on page 190.

The following sections describe the components that make up the flow schema.

The data types listed in the tables are the SQL data type. Table Oracle and SQL Data Types on page 135 describes the SQL Server and Oracle data types mappings and their descriptions.

Flows

The Flows table is the primary table as it details the identities of the Flow definitions. As is shown in Flow Schema Entity-Relationship Diagram on page 190, The Flow table is related to the Nodes, Arcs and Flow_Instances tables. The primary key for Flows is the Flow_ID, a unique identifier derived from the Flow definition's Java object model.

Table 20 Flows

Column Name	Data Type	Length	Allow Nulls	Description
Flow_ID	string	36	No	Primary Key[a].
FlowName	string	120	No	The name of the business flow as entered in the HPBPI Modeler.
FlowDescription	string	256	Yes	The description of the business flow as entered in the HPBPI Modeler.
AvrgTime	float		Yes	The average time taken to complete all flow instances.
SampleCount	integer		Yes	Internal value used for calculating the average time,
ActiveFlows	integer		Yes	The number of flow instances currently being monitored.
TotalFlows	integer		Yes	The total number of flow instances, including the flow instances that have completed.

Table 20 Flows

Column Name	Data Type	Length	Allow Nulls	Description
Status	string	12	Yes	The current status of this flow, which can be one of `Active, Impeded`, `Blocked` or `Deleted`, where: • `Active` means the flow has been deployed. • `Impeded` means that at least one instance of the flow is in the `Impeded` state; see `Status` in Table 21 on page 175. • `Blocked` means that at least one instance of the flow is in the `Blocked` state; see `Status` in Table 21 on page 175. • `Deleted` means the flow has been undeployed.
RepositoryId	string	36	Yes	A unique identifier that can be used to associate different versions of the same object; different versions of a Flow definition or Data definition have different Flow_IDs, but they will have the same RepositoryId.
Subclass	string	256	Yes	For internal use.
Primary_entity	string	36	Yes	A unique identifier (GUID) for the business object class associated with this Flow definition; this column is taken from `Model_ID` in the Data Object table. An example of a `Model_ID` is an Order Definition. Identifies the Data definition for the Flow.

Table 20 Flows

Column Name	Data Type	Length	Allow Nulls	Description
Repository Revision	integer		Yes	The revision number identifying the the version of the Flow that is held in the Repository table.
FolderPath	string	604	Yes	For internal use.

a. This is the column that uniquely identifies rows.

Flow Instance

The Flow_Instance table describes each instance of a specific flow. The primary key is FlowInstance_ID, which is a system-assigned unique identifier.

Primary_entity and Primary_entity_inst are two foreign keys that can be used to link the Flow schema to the Business Entity schema.

Table 21 Flow_Instance

Column Name	Data Type	Length	Allow Nulls	Description
FlowInstance_ID	string	36	No	Primary Key[a].
Flow_ID	string	36	No	Foreign Key[b] to table Flows (`Flow_ID`) column.
Identifier	string	120	Yes	The identity of the related Data definition for this Flow. When you create a flow using the HPBPI Modeler, you can enter the name of a property of a related Data Definition to be used as an Identity property. This column holds the value of the Identity property for the specific flow instance.

Table 21 Flow_Instance

Column Name	Data Type	Length	Allow Nulls	Description
Primary_entity	string	36	Yes	A unique identifier (GUID) for the business object class associated with this flow instance; this column is taken from `Model_ID` in the Data Object table. An example of a `Model_ID` is an Order Definition. Identifies the Data definition for the Flow.
Primary_entity_inst	string	36	Yes	A unique identifier (GUID) for the business object instance associated with this flow instance; this column is taken from `InstanceTable` in the Data Object table. Identifies the instance of the Data Definition for the Flow instance.
Weight	float		Yes	This column provides the data that you can sort and use to provide a weighting for how important the instance is. This column contains the value of the Data Definition property, nominated to be the Weight property, in the HPBPI Modeler. You can use this column to use to sort flow instances into a required order, for example, it might be the value of an order expressed as a classification, such as Gold/Silver/Bronze. This information is mapped from the instance of the Data definition and provides a quick view of the data object from within the flow instance.

Table 21 Flow_Instance

Column Name	Data Type	Length	Allow Nulls	Description
Weight_type	string	120	Yes	This column contains the name of the Data definition property nominated to be the `Weight` property for this flow in the HPBPI Modeler. The `Weight_type` column provides the context for the values in the Weight column. As an example, the Business Process Dashboard might have an option to view Orders sorted by Order Value, where the actual value of the order ($25) is the Weight and Order Value is the Weight type.
StartTime	date		Yes	The time that this flow instance was last started (in date format). Specified in local time.
StartTimeLong Millis	long		Yes	The time (in milliseconds) that the flow instance was last started. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
EndTime	date		Yes	The time that this flow instance was last completed (in date format). Specified in local time.
EndTimeLong Millis	long		Yes	The time (in milliseconds) that the flow instance was last completed. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
EndTimeRecorded	date		Yes	The date and time that EndTime was written to the database table. Specified in local time.

Table 21 Flow_Instance

Column Name	Data Type	Length	Allow Nulls	Description
EndTimeRecorded LongMillis	long		Yes	The date and time (in milliseconds) that EndTime was written to the database table. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
Status	string	12	Yes	The current status of this flow instance, which can be one of `Active`, `Impeded`, `Blocked` or `Completed`, where: • `Active` means the flow instance has met the start condition for at least one node in the flow. • `Impeded` means that the flow instance cannot progress to completion without encountering a blocked node. • `Blocked` means that the flow instance is active at a node where the node's services have failed. • `Completed` means the flow instance has met the complete conditions for one of the end nodes in the flow.
Subclass	string	256	Yes	For internal use

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

Nodes

The Nodes table details individual nodes defined for a flow. The nodes are not defined as reusable entities as this property is only useful during the modeling stage. Representing the node as a unique entity also makes is easier to create queries and views. The primary key for the Nodes table is Node_ID and the foreign key is Flow_ID, which relates the nodes to their parent flow.

Table 22 Nodes

Column Name	Data Type	Length	Allow Nulls	Description
Node_ID	string	36	No	Primary Key[a].
Flow_ID	string	36	No	Foreign Key[b] to the table Flows (`Flow_ID`).
NodeName	string	120	No	The name of the node, taken from the HPBPI Modeler.
NodeDescription	string	256	Yes	The description of the node, taken from the HPBPI Modeler.
NodeType	string	12	Yes	The node type, which can be one of `START`, `WORK`, `END`.
X_Pos	integer		Yes	The X coordinate of the node icon position on the flow graph for display purposes.
Y_Pos	integer		Yes	The Y coordinate of the node icon position on the flow graph for display purposes.
ActiveCount	integer		Yes	The numbers of current, or active, flow instances at this node.
TotalCount	integer		Yes	The number of times the node has been started.

Table 22 Nodes

Column Name	Data Type	Length	Allow Nulls	Description
TotalTime	float		Yes	The accumulated time that all the node instances for this node have been active (or the accumulated time for each progression of the node).
AvrgTime	float		Yes	The average time taken to complete this node instance.
SampleCount	integer		Yes	Internal value used for calulating the average time,
ActiveWeight	float		Yes	Total value of weight parameter for flow instances that are currently active at this node.
TotalWeight	float		Yes	Total value of weight parameter for all flow instances that have been processed at this node.
InstanceRate	float		Yes	The measure of the number of instances per hour that are currently being processed at this node.
WeightRate	float		Yes	The measure of the weight per hour for the flow instances that are currently being processed at this node, for example, the value of customer orders per hour currently being processed at this node.

Table 22 Nodes

Column Name	Data Type	Length	Allow Nulls	Description
LastRateUpdate	date		Yes	The time that the weight and instance rates were last updated (in date format). Specified in local time.
LastRateUpdate LongMillis	long		Yes	The time (in milliseconds) that the weight and instance rates were last updated. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
ResourceStatus	string	12	Yes	The status of the service, and can be `Critical`, `Major`, `Minor`, `Warning` and `Normal`. These status correspond to the HP Operations Manager severity levels. If the node uses more than one service, this field is set to the most serious status for the services used.

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

Node Instance

The Node_Instance table details the instances of nodes associated with each of the flow instances. The primary key for this table is NodeInstance_ID.

Table 23 Node_Instance

Column Name	Data Type	Length	Allow Nulls	Description
NodeInstance_ID	string	36	No	Primary Key[a].
FlowInstance_ID	string	36	No	Foreign Key[b] to table Flow_Instance (`FlowInstance_ID`).
Node_ID	string	36	No	Foreign Key[2] to table Nodes (`Node_ID`).
StartTime	date		Yes	The time the node was last started (in date format). Specified in local time.
StartTimeLong Millis	long		Yes	The time (in milliseconds) that the node was last started. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
EndTime	date		Yes	The time the node last completed (in date format). Specified in local time.
EndTimeLong Millis	long		Yes	The time (in milliseconds) that the node was last completed. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
EndTimeRecorded	date		Yes	The date and time that `EndTime` was written to the database table. Specified in local time.

Table 23 Node_Instance

Column Name	Data Type	Length	Allow Nulls	Description
EndTimeRecorded LongMillis	long		Yes	The date and time (in milliseconds) that `EndTime` was written to the database table. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
Status[c]	string	16	No	The status of the node instance as represented by the most recent event and can be one of `Initial`, `Started`, `Started Again` and `Completed`.
ResourceStatus	string	12	Yes	The status of the service, and can be `Critical`, `Major`, `Minor`, `Warning` and `Normal`. These status correspond to the HP Operations Manager severity levels. If the node uses more than one service, this field is set to the most serious status for the services used.

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table.

c. `Initial` indicates that neither the start condition nor the complete condition for the node have been met. `Started` means the start condition for the node have been met. `Started Again` means the start condition for a node have been met more than once, but does not indicate that any complete conditions have been met. `Completed` means that the complete conditions for the node have been met, but does not indicate that any start conditions have been met. The Business Impact Engine evaluates the status of a node instance based on the order in which the events were submitted, not by the order in which they were received.

Node Instance Started Times

The Node_Instance_StartedTimes table lists the times that a node instance is started, as a node instance can be started multiple times. This table is linked to the Node_Instance table though the NodeInstance_ID column. As it is possible to progress through a node more than once, a single node can have multiple database entries. The Idx column distinguishes each of these entries in the database.

Table 24 Node_Instance_StartedTimes

Column Name	Data Type	Length	Allow Nulls	Description
NodeInstance_ID	string	36	No	This column, plus Idx make up the Primary Key.
Idx	integer		Yes	The order of the start times.The index starts at zero (0) and increments one (1) for every time a node instance that meets it start condition.
StartedTime	date		Yes	The time that the start condition for this node instance were met (in date format). Specified in local time.
StartedTimeLong Millis	long		Yes	The time (in milliseconds) that the start conditions for this node instance were met. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)

Node Instance Completed Times

The Node_Instance_CompletedTimes table lists the times that the node instance meets its complete conditions, as a node instance can be completed multiple times. It is linked to the Node_Instance table through the NodeInstance_ID column.As it is possible to progress through a node more than once, a single node can have multiple database entries. The Idx column distinguishes each of these entries in the database.

Table 25 Node_Instance_CompletedTimes

Column Name	Data Type	Length	Allow Nulls	Description
NodeInstance_ID	string	36	No	This column, plus `Idx` make up the Primary Key.
Idx	integer		Yes	The order of the completed times.The index starts at zero (0) and increments one (1) for every time a node instance that meets it complete condition.
CompletedTime	date		Yes	The time that the complete conditions for this node instance is met In date format). Specified in local time.
CompletedTime LongMillis	long		Yes	The time (in milliseconds) that the complete conditions for this node instance were met. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)

Arcs

Arcs are a component of the Flow definition and describe the links between the nodes that create a particular flow.

Arcs have a many-to-one relationship with the Flows table.

Table 26 Arcs

Column Name	Data Type	Length	Allow Nulls	Description
Flow_ID	string	36	No	Foreign Key[a] to table Flows (`Flow_ID`).
Source	string	36	No	GUID of source node Node_ID.
Destination	string	36	No	GUID of destination node Node_ID.
Type	string	12	Yes	Type can be `NORMAL` or `SEQUENCE`. `SEQUENCE` indicates a Check Sequence arc is defined.

a. This is the column that exists as the primary key in another table.

Services

The Resources table represents the services, or the IT applications and utilities that a node is linked to and is required to monitor, for example, the Take Order node might rely on a Web site and e-commerce application. These are the services that have been defined in OVIS and HP Operations Manager and that have been linked into a specific business flow.

Table 27 Resources

Column Name	Data Type	Length	Allow Nulls	Description
Resource_ID	string	296	No	Primary Key[a].
ResourceName	string	256	No	The name of the service.
ResourceType	string	40	No	Identifies the type of service and can be one of `OVIS`, `OVSN`, `SALS` or `SOAM` `OVSN` is used to indicate either an OpenView Service Navigator or an Operations Manager for Windows Service. `SALS` is used to indicate a Standalone Service. `OVIS` and `SOAM` are used to represent OpenView Internet Services and SOA Manager Services respectively.
Resource Description	string	256	Yes	A description of the service as entered in the HPBPI Modeler.
Status	string	12	Yes	The status of the service, and can be `Critical`, `Major`, `Minor`, `Warning` and `Normal`. These status correspond to the HP Operations Manager severity levels.
RootCause	text		Yes	The current root cause obtained from OVIS or HP Operations Manager.

Table 27 Resources

Column Name	Data Type	Length	Allow Nulls	Description
LastChange	date		Yes	The time that the service status was last modified (in date format). Specified in local time.
LastChangeLong Millis	long		Yes	The time (in milliseconds) that the service status was last modified. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
Deployment Status	string	12	Yes	The current deployment status of this service, which can be one of `Active` or `Deleted`, where: • `Active` means the service is deployed. • `Deleted` means the service is undeployed, but is still used by a flow.

a. This is the column that uniquely identifies rows.

Node2Resources

The Node2Resources table maps nodes to services.

As services are re-usable, it is possible for nodes and services to have a many-to-many relationship, that is, many services can be used by many nodes and more than one node can use more than one service. As an example, the Get Order and Update Order nodes might both link to the Web site and e-commerce services.

Table 28 Node2Resources

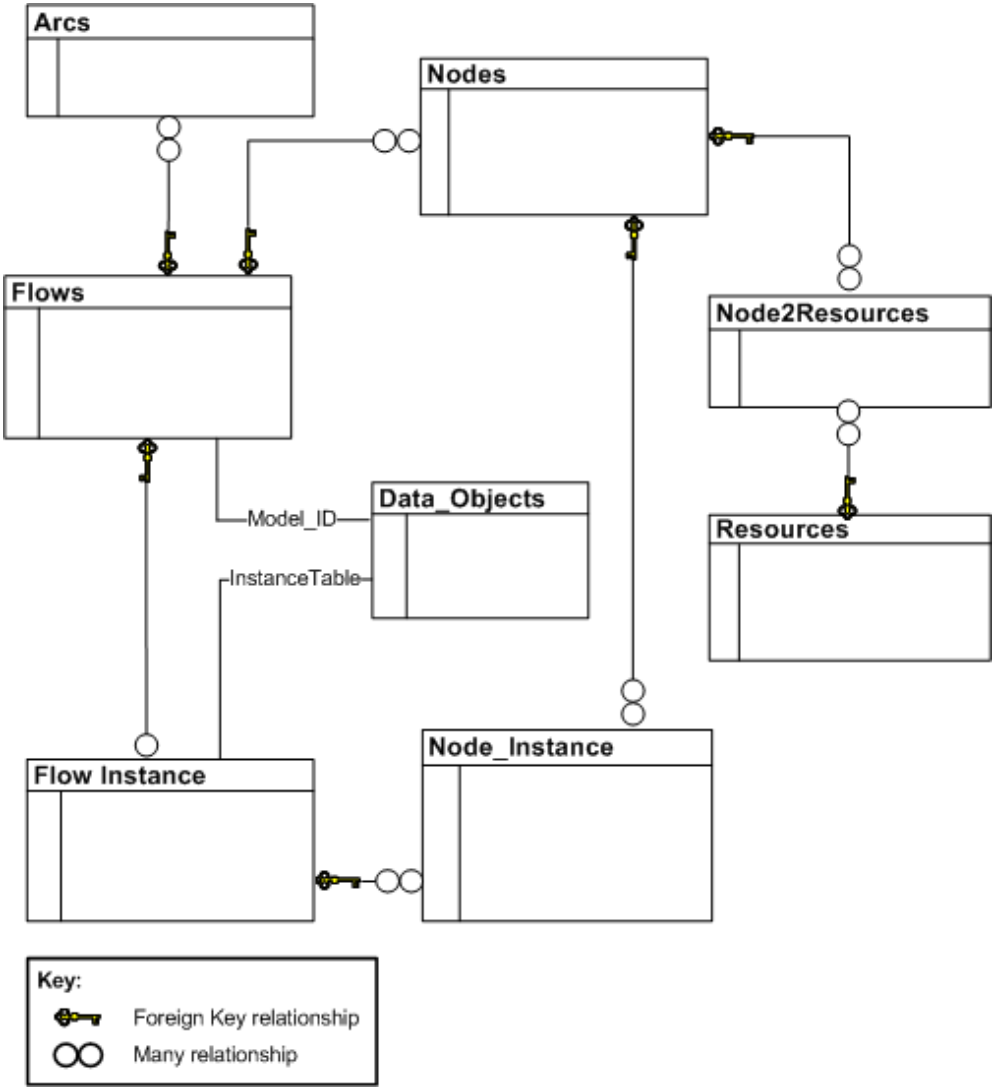
Column Name	Data Type	Length	Allow Nulls	Description
Node_id	string	36	No	Foreign Key[a] to table Nodes (`Node_ID`).
Resource_ID	string	296	No	Foreign Key[1] table Services (`Resource_ID`).

a. This is the column that exists as the primary key in another table.

Entity Relationships for Flow Schema

The following diagram shows the relationship between the database tables for the Flow schema.

Figure 33 Flow Schema Entity-Relationship Diagram



Business Entity Schema

This section describes how data entered in the HPBPI Modeler is defined in the database. For each business object there is a corresponding table in the dynamic schema. This table is created according to the guidelines described in the following sections.

Data item names

The names for columns in the schema are based on the values you enter as the data items names in the Modeler. When added to the database, any illegal characters are replaced using a unicode representation, for example, `Orderu20Amount`, is a representation of Order Amount, where the `u20` is the hexadecimal for space.

If you enter u*nn* (where *nn* is a number) as part of a data item, be aware that this could cause data clashes with another data item that includes the same string as unicode character replacement.

Data types

The dynamic schema uses a subset of data types used in the Flows schema; see Table Oracle and SQL Data Types on page 135 for the complete list.

Keys

The tables that are generated have internally-generated Primary keys defined to assist in referential integrity management and for performance reasons. Referential integrity is a feature that prevents users or applications from entering inconsistent data.

Data Objects

The Data_Objects table lists the types of Business Data definition that have been defined using the Modeler, plus statistics and statuses, which are calculated for these objects. The Primary Key is the Model_ID. Each Data Definition has a corresponding Instance table, which lists the instances of that object.

Table 29 Data_Objects

Column Name	Data Type	Length	Allow Nulls	Description
Model_ID	string	36	No	Primary Key.
Name	string	120	No	The Data definition name.
Description	string	256	Yes	The description of the Data definition.
AvrgTime	float		Yes	Average lifetime of instances of this data type in milliseconds.
SampleCount	integer		Yes	Internal value used for calulating the average time,
ActiveInstances	integer		Yes	Current number of instances.
TotalInstances	integer		Yes	Total number of instances.
Status	string	12	Yes	The current status of the data object, which can be `Active` or `Deleted`.
RepositoryId	string	36	Yes	A unique identifier that can be used to associate different versions of the same definition; different versions of a Flow definition or Data definition have different Flow_IDs or Model_IDs but they have the same RepositoryId.
InstanceTable	string	30	Yes	The name of the database table used to store instances of this Data definition.

Table 29 Data_Objects

Column Name	Data Type	Length	Allow Nulls	Description
Subclass	string	256	Yes	For internal use. This column contains the name of the Data definition instance table.
Repository Revision	integer		Yes	Revision of Data definition in the Model Repository. This revision information is displayed using the Repository Explorer.
FolderPath	string	604	No	For internal use.

Data Definition Instances

The Data definition instance table cannot be named or fully specified as it depends on the data entered using the Modeler; however, some columns appear in every Data definition instance and these are listed in the following table.

The name of this table is referenced in the `InstanceTable` column of the `Data_Objects` table when the Data definition is deployed within the Modeler.

Table 30 Data Definition Instance

Column Name	Data Type	Length	Allow Nulls	Description
StartTime	date		Yes	The time that this instance was started. Specified in local time.
StartTimeLong Millis	long		Yes	The time (in milliseconds) that this instance was started. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
EndTime	date		Yes	The time (in milliseconds) that this instance completed. Specified in local time.
EndTimeLong Millis	long		Yes	The time (in milliseconds) that this instance completed. Specified as the number of milliseconds since epoch (00:00:00 01/01/1970 UTC.)
Status	string	12	Yes	The current status of the instance, which can be one of `Active` or `Completed`.
Id	string	36	No	Primary Key[a].
Model_ID	string	36	Yes	Foreign Key[b]to Data_Objects table (Model_ID).

a. This is the column that uniquely identifies rows.

b. This is the column that exists as the primary key in another table

Business Event Handler Schemas

The following schemas are defined for the Business Event Handler. These schemas are used for the Event Store and for the openadaptor patients that are stored in the event hospital; the Event Hospital.

The tables structures are defined by openadaptor and in this context, a patient is an HPBPI event, which is in an openadaptor format as it is passed through openadaptor.

Business Event Handler Event Store

The Event Store is an openadaptor Sink component that persists messages to a database repository. The messages are stored in the table in a string or equivalent format using Data Object XML format.

The Event Store is a persistent store that can be used to either:

- Take a copy of every event for archiving or auditing purposes
- Provide persistence in the Event Layer

Note that the Event Store can be use in combination with Hospitals; see the *Business Process Insight Integration Training Guide - Business Events* for details of the Event Store and how it can be used in conjunction with the Event Hospital.

Table 31 EVENT_STORE

Column Name	Data Type	Length	Allow Nulls	Description
GUID	string	256	No	A unique identifier.
EVENT_GROUP	string	256	Yes	The event group that this event belongs to. Event grouping is a way of logically organizing events. An event group might be a path containing slashes (/) for further subgrouping.
EVENT_NAME	string	256	Yes	The event name.

Table 31 EVENT_STORE

Column Name	Data Type	Length	Allow Nulls	Description
DOXML	Variable length text	up to 2GB	No	The message.
STATUS	string	100	Yes	The status of the event, which can be one of `Unsent` or `Sent`.
TIME_RX	date		No	Time when the event details are written to the Event Hospital. Specified in local time.
TIME_TX	date		Yes	Time when the event details are retrieved from the Event Hospital.

Event Hospitals

Event hospitals, can also be used to provide persistence. Event Hospitals are used to store events when an event can not be delivered to HPBPI for some reason. This might be because the event is invalid, or because the Business Impact Engine is offline.

Table 32 DBusMH_Patient

Column Name	Data Type	Length	Allow Nulls	Description
PatientId	integer		No	A unique identifier for each patient added to the event hospital. The identifier starts at 1 for the first event.
PatientStatus	string	20	No	The status of this patient, which can be one of: • `New`, a newly added patient • `Treated`, this patient message has been updated • `Examined`, this patient message has been examined • `Discharge_Wait`, this patient is waiting to be discharged • `Dishcharge`, this patient has been discharged • `Re_News`, this patient has been re-added to the hospital • `Dead`, this patient has an unrecoverable error
DestAppName	string	60	No	The destination of this patient message; within HPBPI, this value is set to `BIA_app`.
Subject	string	255	Yes	The subject of this patient message; within HPBPI, this value is set to `BIA_subject`.

Table 32 DBusMH_Patient

Column Name	Data Type	Length	Allow Nulls	Description
RejectReason	string	255	Yes	The PipelineException message content, that is, the result of PipelineException.getMessage() from Java.
SourceAppName	string	60	Yes	The publisher of this patient message.
PubUniqueId	string	255	Yes	This is reserved for the publisher's unique message identifier.
IndexParam1	string	60	Yes	Contains attributes of this patient data object and is used for performance efficiencies when unpacking the patient data object.
IndexParam2	string	60	Yes	Contains attributes of this patient data object and is used for performance efficiencies when unpacking the patient data object.
IndexParam3	string	60	Yes	Contains attributes of this patient data object and is used for performance efficiencies when unpacking the patient data object.
IndexParam4	string	60	Yes	Contains attributes of this patient data object and is used for performance efficiencies when unpacking the patient data object.
Admitted	date		No	The date that this patient was given the status `New`.
Examined	date		Yes	The date that this patient was given the status `Examined`.
Treated	date		Yes	The date that this patient was given the status `Treated`.

Table 32 DBusMH_Patient

Column Name	Data Type	Length	Allow Nulls	Description
Discharged	date	8	Yes	The date that this patient was given the status `Discharged`.
MarkedDead	date	8	Yes	The date that this patient was given a status of `Dead`.
Retried	integer		Yes	The number of times the hospital has tried to send a message that has been marked for `DISCHARGE` back into the HPBPI system.
AdmittedUser	integer		No	The database user that admitted this patient.
ExaminedUser	integer		Yes	The database user that marked this patient as `Examined`.
TreatedUser	integer		Yes	The database user that marked this patient as `Treated`.
DischargedUser	integer		Yes	The database user that marked this patient as `Discharge_Wait` or `Discharge`.
MarkedDeadUser	integer		Yes	The database user that marked this patient as `Dead`.
Patient	Variable length text	up to 2GB	Yes	The openadaptor data object that has been added to the hospital in XML format.

Complete List of HPBPI Database Tables

The following tables list the all the database tables and views that are created for HPBPI. Many of these tables are for internal use only and should not be modified.

This is list is provided in order that you can make sure that all the tables are included in any backup and recovery procedures and for you to check for any potential database table name clashes.

The capitalization used in the table is for ease of reference only. How the table names are represented in your database is database specific; for example, for an Oracle Server all table and view names are upper case.

Table 33 on page 200 lists all the HPBPI database tables and Table 34 on page 203 lists all the views.

Table 33 Complete List of HPBPI Database Tables

Table Name	Internal Use Only (yes/no)
AdaptorConfig	yes
Arcs	no
BACDataSampleDestinations	yes
BCE_Metric_Id_Gen	yes
DataInstIdToBeDeleted	yes
Data_Objects	no
DBusmh_Attribute	yes
DBusmh_EditableAttribute	yes
DBusmh_Patient	no
DBusmh_Role	yes
DBusmh_User	yes
DBusmh_UserRole	yes
EventQueue	yes

Table 33 Complete List of HPBPI Database Tables

Table Name	Internal Use Only (yes/no)
Event_Store	no
FailedBACSamples	yes
FlowDataFilters	yes
FlowIdsToBeUpdated	yes
FlowInstIdToBeDeleted	yes
FlowNodeZones	yes
Flows	no
Flow_Instance	no
ImpactThreshold	yes
MetricToBACDestination	yes
Metric_CustomTypes	no
Metric_Definitions	yes
Metric_Dim_Dates	no
Metric_Dim_Flows	no
Metric_Dim_Flow_Instances	no
Metric_Dim_Groups	no
Metric_Dim_Metrics	no
Metric_Dim_Thresholds	no
Metric_Events	yes
Metric_Events_Engine_Status	yes
Metric_Events_Seqs_To_Remark	
Metric_Fact_Alerts	no
Metric_Fact_Statistics	no

Table 33 Complete List of HPBPI Database Tables

Table Name	Internal Use Only (yes/no)
Metric_Fact_Values	no
Metric_Generate_Errors	yes
Metric_Notification_Check_Time	yes
Metric_Staging_Statistics	yes
Metric_Staging_Stats_Modified	yes
Metric_Stored_Procedures[a]	yes
NodeIdAndResourceStatus	yes
NodeInstToBeDeleted	yes
Nodes	no
Nodes2Resources	no
Node_Instance	no
Node_Instance_CompletedTimes	no
Node_Instance_StartedTimes	no
NS_Digesterstore	yes
NS_DigesterSubscriptions	yes
NS_EmailRetry	yes
NS_EmailSubscriptions	yes
NS_EmailUsers	yes
NS_OVORetry	yes
NS_OVOSubscriptions	yes
NS_ScriptRetry	yes
NS_ScriptSubscriptions	yes
NS_SLAEmailSubscriptions	yes

Table 33 Complete List of HPBPI Database Tables

Table Name	Internal Use Only (yes/no)
NS_SLOEmailSubscriptions	yes
OVIS_AlarmStatus	yes
OVISTimeStamps	yes
Repos_defns	yes
Repos_Folders	yes
Repos_Labels	yes
Repos_Labels2Defns	yes
Resources	no
RMIEventRetry	yes
SOAPEventRetry	yes
Version	yes

a. this table is available only with Microsoft SQL Server.

The following table lists all the HPBPI database views.

Table 34 Complete List of HPBPI Database Views

View Name	Internal Use Only (yes/no)
Dbusmh_View	yes
Metrics	no
Metric_Values	no
Repos_Definitions	yes

The following table lists the HPBPI database indexes.

Table 35 Complete List of HPBPI Database Indexes

Index Name	Internal Use Only (yes/no)
Flow_Instance_Idx_*nnn*	yes
Idx*n*	yes
Metric_Dim_Dates_Idx_*nnn*	yes
Metric_Dim_Groups_Idx_*nnn*	yes
Metric_Events_Idx_*nnn*	yes
Metric_Fact_Groups_Idx_*nnn*	yes
Metric_Fact_Alerts_Idx_*nnn*	yes
Metric_Fact_Statistics_Idx_*nnn*	yes
Metric_Fact_Values_Idx_*nnn*	yes
Metric_Staging_Statistics_Idx_*nnn*	yes
Node_Instance_Comple_Idx_*nnn*	yes
Node_Instance_Idx_*nnn*	yes
Node_Instance_Starte_Idx_*nnn*	yes
Sys_*string*	yes

There are also stored procedures, database triggers and other database scripts defined by HPBPI and potentially defined by you. These also need to be taken into account for backup purposes or if you need to move the HPBPI data.

It is strongly recommended that you create an Oracle User or Microsoft SQL Server Database specifically for HPBPI as it makes the task of identifying this HPBPI data much easier.

B Expression Grammar in Flow, Data and Filter Definitions

This appendix lists the rules for the grammar that can be used for business flow progression rules and expressions, and expressions within business process metric filter definitions.

The progression rules and expressions are within the HPBPI Modeler for the start and complete conditions for nodes, and for filter expressions for Data definitions when subscribing to events.

Business process metric filter expressions are used within the Metric definer to optionally specify the conditions for collecting statistical data for the business process metrics.

The grammar used for expressions is similar to Java expressions. The following sections describe the grammar and its construct. Further examples of using this grammar are provided in the *Business Process Insight Integration Training Guide - Modeling Flows*.

Grammar

This following is an informal description of the grammar, it shows how expressions can be constructed. Note that this construct is subject to specific limitations according to how it is being used. Examples of the use of the grammar are provided in the following sections.

```
expression =>
 property-value = expression
|expression + expression
|expression - expression
|expression * expression
|expression / expression
|expression % expression
|- expression
|expression && expression
|expression || expression
|! expression
|expression == expression
|expression > expression
|expression >= expression
|expression < expression
|expression <= expression
|expression != expression
|(expression)
|expression ? expression : expression
| value

value =>
       constant_value
      |property_value
      |function_return_value
```

See section Function Return Values on page 208 for possible values for `function_return_value`.

```
constant_value =>
       string_contant
      |integer_constant
      |real_constant
      |true
      |false
      |null

property_value =>
       root_object.relationship_name.property_name
```

where:

- `root_object` is `this` or `event`.

  If no `root_object` is specified, the expression assumes `this`.

- `relationship_name` can be included zero or more times, depending on how the property relates to the root object.

Possible `property_value` expressions are:

- `this.property`
- `this.data.property`
- `event.property`
- `property`

The following are examples that you might use in your expressions. Further examples are given in the *Business Process Insight Integration Training Guide - Modeling Flows*:

- `this.OrderNumber`
- `this.Customer.Customer_ID`

Function Return Values

The following sections describe the data and string values that can be returned from the `value` expression.

These functions can be used in filter expressions, binding expressions, progression rules and assignment actions.

Functions for Dates and Times

The following functions convert expressions into milliseconds:

- `hours (`*`number`*`)`
- `minutes (`*`number`*`)`
- `seconds (`*`number`*`)`
- `days (`*`number`*`)`

These functions convert *`number`* to the equivalent number of milliseconds (which are the Business Impact Engine time units).

Functions for Identifying String Values

The following functions take a string parameter and return a boolean result.

- `string_property.contains(`*`value`*`)`

  Returns `true` if *`value`* is found in the property value .

- `string_property.starts(`*`value`*`)`

  Returns `true` if the property value starts with *`value`*.

- `string_property.ends(`*`value`*`)`

  Returns `true` if the property value ends with *`value`*.

Functions for All Property Types

The following function takes one or more parameters and returns a boolean result:

`property.in(value,value,...)`

This expression returns `True` if the value of `property` is one of the listed set of values (`value, value,...`), and `False` if it is not.

Flow Progression Rules

The grammar for flow progression rules is slightly different from the simple expression grammar, as it is used to describe deltas or changes to Data definition properties.

There are four styles of progression rule provided in HPBPI as follows:

- Complete on first assignment
- Complete on transition
- Start and complete on transitions
- Advanced conditions

The Advanced conditions style of progression rule requires you to enter the methods for the progression rules directly as described in section Methods for Progression Rules on page 212.

For the remaining progression styles, you do not need to enter the methods, you just select the style of the progression rule that you want for the node. Ultimately, each style uses a subset, or selection of the methods described in section Methods for Progression Rules on page 212. You can see these methods if you define a progression rule using one of the styles and then change the style to Advanced conditions to view the methods created for the style.

More detailed information about using these progression rules in your flows is provided in the *Business Process Insight Integration Training Guide - Modeling Flows*.

The methods used for the progression rule styles are listed in Table 36 on page 210.

Table 36 Methods Used for Progression Rule Styles

Style	Methods Used	Comments
Complete on first assignment	`before()==null`	This style of progression rule uses the method listed to determine when the property value changes from Null to any other value.

Table 36 Methods Used for Progression Rule Styles

Style	Methods Used	Comments
Complete on transition	Complete Condition: `before().in()` and `after().in()`	This style of progression rule uses the methods listed to determine when the node complete condition is met. It does this by testing the original and modified value of a property against one or more values in the rule. Use of the `in()` method enables you to enter a selection of values that can be tested. If you do not enter a value, any value for the property can satisfy the condition.
Start and complete on transitions	Start Condition: `before().in()` and `after().in()` Complete Condition: `before().in()` and `after().in()`	This style of progression rule uses the methods listed to determine when the node start and node complete conditions are met. It does this by testing the original and modified value of a property against one or more values in the rule. Use of the `in()` method enables you to enter a selection of values that can be tested. If you do not enter a value, any value for the property can satisfy the condition.

Methods for Progression Rules

In order to express progression rules using the `Advanced Conditions` option in the HPBPI Modeler, there are methods for Data definitions and their properties. A method represents the value relating to the data object or property at a point in time.

The methods are described in the following table and apply only to a property in a flow progression rules.

Table 37 Methods for Progression Rules

Method	Description
`this.data.property.changed()`	This function is used to test when the property value changes. When the property value changes it returns a result of `True`.
`this.data.property.before()== "prop-value"`	This evaluates to the value of the property before a change and is executed only when the property values changes. It is implicit in this expression that the property value was as stated by *`prop-value`* and is now no longer that value. The type returned is the same as the type of the property.
`this.data.property.after()== "prop-value"`	This evaluates to the value of the property after a change and is executed only when the property value changes. It is implicit in this expression that the property value was a value other than *`prop-value`* and is now *`prop-value`*. The type returned is the same as the type of the property.

Table 37 Methods for Progression Rules

Method	Description
`this.data.created()`	This evaluates to true when the data definition is created. This method is executed only when a new Data definition is created. The method returns a Boolean result.
`this.data.property.in( "prop-value1", "prop-value2", ...)`	This evaluates to true when the value of the propery is set to any of the listed values. The method returns a Boolean result.
`this.data.terminated()`	This evaluates to true when the data definition is terminated, specifically when the Data definition received an event that is flagged as terminating it. This is specified in the Event Subscription dialog for the Data Definition; `Terminate this instance of the Data Definition after handling the event.` The method returns a Boolean result.

Example of Complete Condition for Start Node When Data Definition Created

The following shows an example of the expression that you use when you want a flow instance to be started when a Data definition for the flow is created. In this case, a new flow instance is started only when the start and complete conditions for the start node in the flow are met, for example:

```
this.order.created()
```

where:

- `this` is the reference to the flow, in this case the `Order` flow.
- `order` is the Data definition.
- `created()` is a method that evaluates to true when the `order` Data definition is created.

Example Start Condition for Start Node When Data Definition Modified

The following shows an example of the expression that you use when you want a flow instance to be started when the Data definition for the flow is modified. In this case a flow instance is started not when the Data definition is created but when there is a change to the Data definition property.

```
this.order.priority.before() == null &&
this.order.priority.after() <= 3
```

where:

- `this` is the reference to the flow, in this case the `Order` flow.
- `order` is the primary Data definition.
- `priority` is an integer property.
- `before()` is a method, which returns the value of the property before the change.
- `null` indicates that the property was not initialized before the change.
- `after()` is a method, which returns the value of the property after the change.

Example Start Condition using a Method on a Method

The following example is similar to the example above; however, shows an example of the expression that you use when you want a flow instance to be started when the Data definition for the flow is modified and the Data definitions contains a specific string.

```
this.order.customer_id.after().contains("X")
```

where:

- `this` is the reference to the flow, in this case the `Order` flow.
- `order` is the primary Data definition.
- `customer_id` is a string property that is a unique identifier for the customer.
- `after()` is a method, which returns the value of the property after the change.
- `contains()` is a string function, which returns a boolean result depending on whether the string `"X"` is present in the `customer_id`.

Case Sensitivity for Expressions

This section describes case sensitivity for expression properties and for expressions containing string constants.

Expression Properties

Properties and methods in expressions are case sensitive, which means the following expression are not equivalent within your HPBPI system:

- `this.order.priority.before() == null && this.order.priority.after() <= 3`
- `this.Order.Priority.before() == null && this.Order.Priority.after() <= 3`

Expressions with String Constants

Expressions containing string constants are case sensitive, which means the following expressions are not equivalent within your HPBPI system:

- `this.order.status.after() == "a"`
- `this.order.status.after() == "A"`

C Coercion Rules

This appendix describes the rules for how properties are coerced when evaluating binding, filter and assignment expressions. This is an enforced evaluation by the HPBPI Modeler, based on the rules described in this Appendix.

Valid data types for use within HPBPI are:

- String
- Numeric, which can be one of:
  - Integer
  - Long
  - Double
  - Currency
- Boolean
- Date

Assignments

The following table shows how assignments are coerced:

Table 38 Assignment Coercion

Data Type...	Assigned from...
`String`	Any type
`Numeric`	Any other numeric value (with possible truncation if assignment is to an `integer`), or from `Strings` provided that the string is numeric. If the assignment does not produce a valid number, you receive an error similar to the following in the Business Impact Engine log file: `Cannot assign value` *`prop-name`* `to property` *`prop-name`* `as it is not a compatible type.`
`Boolean`	Other Boolean values, or from `Strings` provided that the string is either “true” or “false”. If the assignment does not produce a valid boolean expression, you receive an error similar to the following in the Business Impact Engine log file: `Cannot assign value` *`prop-name`* `to property` *`prop-name`* `as it is not a compatible type.`
`Date`	Other `Dates` or a `Numeric` value (treated as number of milliseconds since 1970).

Expressions

The following table shows how values within expressions are coerced.

Table 39

Operator	Behavior
Arithmetic: • + • – • * • / • %	Supported between Numeric properties or values that can be coerced to numeric using the assignment rules described in Table 38.
Comparison: • < • <= • = • != • > • >=	The comparison operators can be used to provide a: • numeric comparison between Numeric properties or values that can be coerced to numeric using the assignment rules described in Table 38. • date comparison between Date properties. • alphabetic comparison (locale-specific) between String properties or values that can be coerced to Strings using the using the assignment rules described in Table 38 (including Boolean). The result of a comparison is always a boolean value.
String tests: • starts() • ends() • contains() • in()	Performs a test on a String property. Note that these tests cannot be applied to non-string properties.

Table 39

Operator	Behavior
Logical: • `!` (Not) • `&&` (And) • `||` `(Or)`	Supported between Boolean properties or values that can be coerced to boolean using the assignment rules described in Table 38.
Conditional value `if, then, else:` • `?:`	First operand must be Boolean or be coerced to boolean, other operands can be any type; however they must be the same each other.

Using NULL Within an Expression

The HPBPI Modeler allows only the `equals` and `not-equals` tests with the `null` constant. However, there might be expressions using other operators involving values that are `null` when these expressions are executed, for example how is the following expression evaluated when *index* is `null`?

```
index + 1
```

The following table shows the rules that apply when comparing data items within an expression where one data item contains a `null` value.

Table 40

Operator	Comparison
• == • !=	Returns TRUE if the other operand is (or is not) `null`.
All other comparison operators	Always return `FALSE`. For example the following expressions both return `FALSE` if `index` is `null`: • `index > 10` • `index <= 10`
Arithmetic operators	Arithmetic operators for expressions containing a null value always return null as the result, for example: `index + 1` returns `null` if `index` is equal to `null`

Note that the `null` constant is case sensitive and when used in Java expressions, should always be lower case.

Index

A

ACTIVE_INSTANCES_HIGH, 116, 119

ACTIVE_INSTANCES_LOW, 116, 119

ACTIVE_VALUES_HIGH, 117, 120

ACTIVE_VALUES_LOW, 116, 119

ActiveCount
  Nodes table, 179

ActiveFlows
  Flows table, 173

ActiveInstances
  Data_Objects table, 192

ActiveWeight
  Nodes table, 180

Adapter
  Business Event Handler, 38
  for HPBPI Server, 27
  for iWay integration, 59

Administration Console
  architecture, 43
  HPBPI Notification Server, 43
  HPBPI Server Administration Console, 43

Admitted
  DBusMH_Patient table, 198

AdmittedUser
  DBusMH_Patient table, 199

Advanced conditions
  flow progression rules, 210
  grammar rules, 210

Alarms and SLOs
  configuring, 115

AlertCriticalLevel
  Metric_Dim_Thresholds table, 163

AlertCriticalLevelDefnUnit
  Metric_Dim_Thresholds table, 165

Alert facts dimension, 136

AlertLevel
  Metric_Fact_Alerts table, 139

AlertMajorLevel
  Metric_Dim_Thresholds table, 163

AlertMajorLevelDefnUnit
  Metric_Dim_Thresholds table, 165

AlertMinorLevel
  Metric_Dim_Thresholds table, 163

AlertMinorLevelDefnUnit
  Metric_Dim_Thresholds table, 165

AlertStatus
  Metric_Fact_Alerts table, 139

AlertWarningLevel
  Metric_Dim_Thresholds table, 163

AlertWarningLevelDefnUnit
  Metric_Dim_Thresholds table, 166

Architecture
Administration Console, 43
and HPBPI database, 18
Business Event Handler, 37
Business Impact Engine, 25
Business Process Dashboard, 13
diagram for HPBPI, 12, 46
for Business Availability Center, 66
HPBPI, 11, 45
HPBPI Operations Manager Adapter, 55
HPBPI Probes and alarms, 53
HPBPI Server, 23
Intervention Client, 16
Metric definer, 29
Metric Engine, 31
Notification Server, 34
Repository Explorer, 33
Security, 44
Self-Healing Services, 62
Arcs table, 186
Destination, 186
Flow_ID, 186
Source, 186
Type, 186
Arithmetic operator
coercion, 219
Assignment
coercion and data types, 218
expressions and coercion rules, 217
AVAILABILITY, 116, 118, 119
AverageValue
Metric_Fact_Statistics table, 149
AvrgTime
Data_Objects table, 192
Flows table, 173
Nodes table, 180

B

backlog
Data Sample for Business Availability Center, 86
BIAEngineAdaptor
Engine Adapter, 39
Binding expressions
coercion rules, 217
Boolean
data type coercion rule, 218
BPEL
importing into Model Repository, 41
BPI
Administration Console, 43
and HPSD integration, 125
and HPSD service mapping, 128
and iWay, 59
and SAP, 58
flow model deployer, 42
Modeler, 41
Model Repository, 41
propagating events, 39
BPI. See also HPBPI
Building applications
using HPBPI data, 134

Business Availability Center
adapter colleague for HPBPI Server, 28
and BPI data sampling, 75
and MY BAC, 67
architecture diagram, 66
configuring Operation Service Source, 76
creating a File Source adapter, 80
creating a portlet Server for HPBPI, 70
data sample destination, 78
design-time integration, 71
exporting HPBPI Flow definitions, 68
exporting metric definitions, 69
integrating with HPBPI, 65
integration options, 68
modeling IT processes, 66
monitoring business impact data from HPBPI, 67
monitoring by HPBPI, 51
operational service status, 67
run-time integration, 73
terminology, 67
using as a source of operational service data, 76
Viewing flows and metrics, 67
Business Entity schema, 134, 191
Data_Object-Instance table, 193
Data_Objects table, 192
data item names, 191
Business Event Handler
architecture, 37
event adapter, 38
HPBPI Server component, 23
openadaptor, 37
receiving events, 39
schema, 134, 195
business health
Data Sample for Business Availability Center, 86
Business Impact Engine
architecture, 25
BIAEngineAdaptor, 39
Business Metrics
architecture diagram for Business Process Metric definer, 30
schema, 133, 136
Business object manager
HPBPI Server, 27
Business Process Dashboard
and business process metrics, 15
architecture, 13
JSPs, 15
Business process metrics
Business Process Metric definer an HPBPI Server component, 29
data in the HPBPI database, 20

C

C_OVBPI_FLOW_PROBE
ACTIVE_INSTANCES_HIGH, 116
ACTIVE_INSTANCES_LOW, 116
ACTIVE_VALUES_HIGH, 117
ACTIVE_VALUES_LOW, 116
AVAILABILITY, 116
INSTANCE_TPUT_HIGH, 117
INSTANCE_TPUT_LOW, 117
objective information, 116
RESPONSE_TIME, 116
SETUP_TIME, 116
TRANSFER_TPUT, 116
VALUE_TPUT_HIGH, 117
VALUE_TPUT_LOW, 117
C_OVBPI_FLOW OVBPI_DB_Password
OVBPI_DB_Password, 110
C_OVBPI_FLOW - OVBPI_FLOW probe, 108

C_OVBPI_FLOW probe
OVBPI_DB_User_Name, 109
OVBPI_Flow_Name, 110
OVBPI_Identifier, 109
Port, 109
Target Host, 109
Username, 109
C_OVBPI_METRIC_PROBE
AVAILABILITY, 118
objective information, 118
RESPONSE_TIME, 118
SETUP_TIME, 118
TBN_DURATION_HIGH, 118
TBN_DURATION_LOW, 118
TRANSFER_TPUT, 118
C_OVBPI_METRIC - OVBPI_METRIC probe, 108
C_OVBPI_METRIC probe
OVBPI_DB_Password, 112
OVBPI_DB_User_Name, 111
OVBPI_Flow_Name, 112
OVBPI_Identifier, 111
OVBPI_Metric_Name, 112
Port, 111
Target Host, 111
Username, 111
C_OVBPI_NODE_PROBE
ACTIVE_INSTANCES_HIGH, 119
ACTIVE_INSTANCES_LOW, 119
ACTIVE_VALUES_HIGH, 120
ACTIVE_VALUES_LOW, 119
AVAILABILITY, 119
INSTANCE_TPUT_HIGH, 120
INSTANCE_TPUT_LOW, 120
objective information, 119
RESPONSE_TIME, 119
SETUP_TIME, 119
TRANSFER_TPUT, 119
Username, 113
VALUE_TPUT_LOW, 120
C_OVBPI_NODE - OVBPI_NODE probe, 108
C_OVBPI_NODE probe
OVBPI_DB_Password, 113
OVBPI_DB_User_Name, 113
OVBPI_Flow_Name, 114
OVBPI_Identifier, 113
OVBPI_Node_Name, 114
Port, 113
Target Host, 113
Case sensitivity
for expressions within progression rules, 216
Coercion
arithmetic expressions, 219
comparison expressions, 219
conditional expressions, 220
for expressions, 219
logical expressions, 220
rules for binding, filter and assignment expressions, 217
string expressions, 219
Comparison operator
coercion, 219
Complete condition
after(), 212
before(), 212
changed(), 212
created(), 213
in(), 213
methods for, 212
terminated(), 213
CompletedTime
Node_Instance_CompletedTimes table, 185
CompletedTimeLongMillis
Node_Instance_CompletedTimes table, 185

Complete on first assignment
grammar rules, 210
Complete on transition
grammar rules, 210
Component
Flow model Deployer, 42
Model Repository, 41
Conditional operator
coercion, 220
Configuring
an XML File Source adapter in the Business Availability Center, 80
HP Operations Dashboard integration, 60
OVIS alarms and SLOs, 115
Configuring a Business Availability Center
data sample destination, 78
Console
HPBPI Administration Console, 42
Notification Server Administration Console, 42
CountTotal
Metric_Fact_Statistics table, 149
CreatedDate
Metric_Dim_Metrics table, 152
Metric_Dim_Thresholds table, 166
Creating
an XML File Source adapter in the Business Availability Center, 80
Creating and MY BAC application for HPBPI, 84
Currency
data type definition, 135
CurrentAlertFromTime
Metric_Dim_Thresholds table, 164
CurrentAlertFromTimeLongMillis
Metric_Dim_Thresholds table, 164
CurrentAlertLevel
Metric_Dim_Thresholds table, 164
CurrentAlertStatus
Metric_Dim_Thresholds table, 164
Custom
fields and defining for HPSD, 128
probes and using with OVIS, 105
CustomMetricDescription
Metric_CustomTypes table, 167
CustomMetricName
Metric_CustomTypes table, 167
CustomSPName
Metric_CustomTypes table, 167

D

Dashboard
and business process metrics, 15
and HP Operations Dashboard, 60
architecture, 13
JSPs, 15
Data_Object-Instance table, 193
EndTime, 194
EndTimeLongMillis, 194
Id, 194
Model_ID, 194
StartTime, 194
StartTimeLongMillis, 194
Status, 194

Data_Objects table
ActiveInstances, 192
AvrgTime, 192
Business Entity schema, 192
Description, 192
FolderPath, 193
InstanceTable, 192
Model_ID, 192
Name, 192
RepositoryID, 192
RepositoryRevision, 193
SampleCount, 192
Status, 192
Subclass, 193
TotalInstances, 192
Database
schema for HPBPI data, 133
DataDefinition_ID
Metric_Dim_Flow_Instances table, 156
Metric_Dim_Flows table, 155
DataDefinitionInstance_ID
Metric_Dim_Flow_Instances table, 156
Data item names
Business Entity schema, 191
Data Sample
backlog, 86
business health, 86
duration, 86
throughput, 86
Data sample destination
and Business Availability Center, 78
Data type
assignment coercion, 218
Business Entity Schema, 191
Data type definition
Currency, 135
Date, 135
Float, 135
HPBPI Schema, 135
Integer, 135
Long, 135
String, 135
Text, 135
variable-length binary, 135
variable-length text, 135
Date
data type coercion rule, 218
data type definition, 135
functions, 208
Date_ID
Metric_Dim_Dates table, 153
Metric_Fact_Alerts table, 138
Metric_Fact_Statistics table, 147
Metric_Fact_Value table, 141
Day
Metric_Dim_Dates table, 153
DayOfWeek
Metric_Dim_Dates table, 153
DayOfWeek_Num
Metric_Dim_Dates table, 154
DBusMH_Patient
database table, 197

DBusMH_Patient table
Admitted, 198
AdmittedUser, 199
DestAppName, 197
Discharged, 199
DischargedUser, 199
Examined, 198
ExaminedUser, 199
IndexParam1, 198
IndexParam2, 198
IndexParam3, 198
IndexParam4, 198
MarkedDead, 199
MarkedDeadUser, 199
Patient, 199
PatientId, 197
PatientStatus, 197
PubUniqueId, 198
RejectReason, 198
Retried, 199
SourceAppName, 198
Subject, 197
Treated, 198
TreatedUser, 199
Deadline
Metric_Fact_Value table, 143
DeadlineLongMillis
Metric_Fact_Value table, 143
DeadlineOverdue
Metric_Fact_Value table, 145
Defining
expressions for flows, 205
DeploymentStatus
Resources table, 188
DestAppName
DBusMH_Patient table, 197
Destination
Arcs table, 186
Developing applications
using HPBPI data, 134
Dimension
alert facts, 136
schema, 136
statistics facts, 146
tables in database, 151
value facts, 140
Discharged
DBusMH_Patient table, 199
DischargedUser
DBusMH_Patient table, 199
DOXML
EVENT_STORE table, 196
duration
Data Sample for Business Availability Center, 86

E

Email
notification data in HPBPI database, 22
Enable OVIS integration, 122
EndCondition
Metrics table, 169
EndNode
Metrics table, 169
EndTime
Data_Object-Instance table, 194
Flow_Instances table, 177
Metric_Dim_Flow_Instances table, 157
Metric_Values table, 170
Node_Instance table, 182
EndTimeLongMillis
Data_Object-Instance table, 194
Flow_Instances table, 177
Metric_Dim_Flow_Instances table, 158
Metric_Values table, 170
Node_Instance table, 182

EndTimeRecorded
Flow_Instances table, 177
Node_Instance table, 182
EndTimeRecordedLongMillis
Flow_Instances table, 178
Node_Instance table, 183
Engine
HPBPI Server component, 23
Entity model schema, 136
Entity relationship diagram
HPBPI schema, 190
Event
hospital and database table, 197
hospital data and HPBPI database, 21
propagation, 39
receivers and transmitters for the HPBPI Server, 28
EVENT_GROUP
EVENT_STORE table, 195
EVENT_NAME
EVENT_STORE table, 195
EVENT_STORE
table, 195
EVENT_STORE table
DOXML, 196
EVENT_GROUP, 195
EVENT_NAME, 195
GUID, 195
STATUS, 196
TIME_RX, 196
TIME_TX, 196
Event adapter
Business Event Handler, 38
Event Handler
See Business Event Handler
Event Store
definition, 195
schema, 195
Examined
DBusMH_Patient table, 198
ExaminedUser
DBusMH_Patient table, 199
Example
start condition for Start Node
Data definition created, 214
Data definition modified, 214
start condition using a method on a method, 215
Expression
coercion, 219
grammar, 205
properties
case sensitivity for Flow definitions, 216
using NULLs, 221

F

Filter expression
coercion rules, 217
Float
data type definition, 135
Flow
data and the HPBPI database, 20
expressions
functional return values, 208
progression rules
flow definitions, 210
schema, 133, 172

Flow_ID
Arcs table, 186
Flow_Instances table, 175
Flows table, 173
Metric_Dim_Flow_Instances table, 156
Metric_Dim_Flows table, 155
Metric_Dim_Metrics table, 151
Metric_Fact_Alerts table, 138
Metric_Fact_Statistics table, 147
Metric_Fact_Value table, 141
Metrics table, 168
Nodes table, 179
Flow_Instance
Metric_Values table, 170
Flow_Instances table, 178
EndTime, 177
EndTimeLongMillis, 177
EndTimeRecorded, 177
EndTimeRecordedLongMillis, 178
Flow_ID, 175
FlowInstance_ID, 175
Identifier, 175
Primary_entity, 176
Primary_entity_ inst, 176
schema, 175
StartTime, 177
StartTimeLongMillis, 177
Status, 178
Weight, 176
Weight_type, 177
Flow definition, 205
expression grammar, 205
expression properties
case sensitivity, 216
functions for dates and times, 208
Modeler
date and time functions for Flow definitions, 208
Progression rules, 210
String values, 208
Time functions, 208
FlowDescription
Flows table, 173
FlowInstance_ID
Flow_Instances table, 175
Metric_Dim_Flow_Instances table, 156
Metric_Fact_Alerts table, 138
Metric_Fact_Value table, 141
Node_Instance table, 182
Flow Instances
C_OVBPI_FLOW - OVBPI_FLOW probe, 108
Flow Instances probe
OVBPI_DB_Password, 110
OVBPI_DB_User_Name, 109
OVBPI_Flow_Name, 110
OVBPI_Identifier, 109
Port, 109
Target Host, 109
Username, 109
Flow Metric probe
OVBPI_DB_Password, 112
OVBPI_DB_User_Name, 111
OVBPI_Flow_Name, 112
OVBPI_Identifier, 111
OVBPI_Metric_Name, 112
Port, 111
Target Host, 111
Username, 111
Flow Metrics
C_OVBPI_METRIC - OVBPI_METRIC probe, 110
FlowName
Flows table, 173
Metric_Dim_Flows table, 155

Flow probe
ACTIVE_INSTANCES_HIGH, 116
ACTIVE_INSTANCES_LOW, 116
ACTIVE_VALUES_HIGH, 117
ACTIVE_VALUES_LOW, 116
AVAILABILITY, 116
INSTANCE_TPUT_HIGH, 117
INSTANCE_TPUT_LOW, 117
RESPONSE_TIME, 116
SETUP_TIME, 116
TRANSFER_TPUT, 116
VALUE_TPUT_HIGH, 117
VALUE_TPUT_LOW, 117
Flows table, 173
ActiveFlows, 173
AvrgTime, 173
Flow_ID, 173
FlowDescription, 173
FlowName, 173
FolderPath, 175
Primary_entity, 174
RepositoryID, 174
RepositoryRevision, 175
SampleCount, 173
Status, 174
Subclass, 174
TotalFlows, 173
FolderPath
Data_Objects table, 193
Flows table, 175
Functional return values
flow expressions, 208

G

Grammar rules, 205
Data definitions, 205
Flow definitions, 205
Metric definitions, 205
Group_ID
Metric_Dim_Groups table, 159
Metric_Fact_Statistics table, 148
Metric_Fact_Value table, 143
GroupName
Metric_Dim_Groups table, 159
Metric_Dim_Metrics table, 152
GroupValue
Metric_Dim_Groups table, 159
GUID
EVENT_STORE table, 195

H

Hour
Metric_Dim_Dates table, 154

HPBPI
Administration Console, 42
and Business Availability Center data sampling, 75
and OVIS integration, 97
and SOA Manager integration, 89
architecture, 11, 45
architecture diagram, 12, 46
Business Availability Center portlet server, 70
Business Event Handler, 37
custom probe configuration for multiple HPBPI Servers, 123
database schema, 133
data sample destination in Business Availability Center, 78
HPSD service calls, 126
importing SOA Manager business events, 90, 93
importing SOA Manager status events, 90
monitoring Business Availability Center, 51
monitoring HPSD processes, 56
monitoring SOA Manager, 52
MY BAC, 81
Notification Server Administration Console, 42
openadaptor, 37
OvbpiProbe.cfg, 123
Probe locations, 122
schema data type definitions, 135
schema entity relationship diagram, 190
Server components, 23
SLAs, 121

HPBPI Accelerator
for SAP integration, 58

HPBPI Adapters, 27

HPBPI and OVIS
design-time integration, 99
run-time integration, 101

HPBPI data
used for developing applications, 134

HPBPI database
business process metric data, 20
diagram of relationship with HPBPI components, 19
email notification data, 22
event hospital data, 21
flow data, 20
metric threshold data, 21
model repository data, 21
part of architecture, 18

HPBPI Intervention Client, 43

HPBPI Modeler
and BPEL, 41
grammar description, 206
grammar rules, 205

HPBPI Notification Server
Administration Console, 43

HPBPI Operations Manager Adapter
architecture, 55

HPBPI Probes and alarms
architecture, 53

HPBPI Server
architecture, 23
Business Availability Center Adapter Colleague, 28
business object manager, 27
Business Process Metric definer component, 29
configuring custom probes, 123
event receivers and transmitters, 28
Metric Engine, 31
Notification Server, 34
OVIS event receiver, 28
Repository Explorer, 33
SOA Manager Client Colleague, 27

HPBPI SLO and SLA violations, 54

HPBPI SOA Manager
configuring access to adapter, 94
HP Operations Dashboard
and HPBPI Dashboard, 60
single signon, 61
HP Operations Dashboard integration, 60
HPSD
custom field definition, 128
integrating with HPBPI, 126
mapping services, 128
monitoring by HPBPI, 56
service calls, 126

I

Id
Data_Object-Instance table, 194
Identifier
Flow_Instances table, 175
Metric_Dim_Flow_Instances table, 156
Idx
Metric_Fact_Value table, 144
Metric_Value table, 171
Node_Instance_CompletedTimes table, 185
Node_Instance_StartedTimes table, 184
IndexParam1
DBusMH_Patient table, 198
IndexParam2
DBusMH_Patient table, 198
IndexParam3
DBusMH_Patient table, 198
IndexParam4
DBusMH_Patient table, 198
INSTANCE_TPUT_HIGH, 117, 120
INSTANCE_TPUT_LOW, 117, 120
InstanceRate
Nodes table, 180
InstanceTable
Data_Objects table, 192
Integer
date type definition, 135
Integrating with iWay, 59
Integrating with SAP, 58
Integrating with the Business Availability Center, 65
Integration options
for Business Availability Center, 68
Intervention Client
architecture, 16
IsAcknowledged
Metric_Fact_Alerts table, 139
IsDeleted
Metric_Dim_Metrics table, 152
IsLatest
Metric_Fact_Statistics table, 148
iWay integration, 59

J

JSPs
used within Dashboard, 15

K

Keys
Business Entity Schema, 191

L

LastChange
Resources table, 188
LastChangeLongMillis
Resources table, 188
LastCheckTime
Metric_Dim_Thresholds table, 163

LastCheckTimeLongMillis
Metric_Dim_Thresholds table, 163
LastRateUpdate
Nodes table, 181
LastRateUpdateLongMillis
Nodes table, 181
LastStatisticsRecordLongMillis
Metric_Dim_Metrics table, 152
LastStatisticsRecordTime
Metric_Dim_Metrics table, 152
LastUpdateTime
Metric_Fact_Value table, 145
LastUpdateTimeLongMillis
Metric_Fact_Value table, 145
Logical operator
coercion, 220
Long
data type definition, 135

M

MarkedDead
DBusMH_Patient table, 199
MarkedDeadUser
DBusMH_Patient table, 199
MaximumValue
Metric_Fact_Statistics table, 149
MeasurementPeriod
Metric_Dim_Metrics table, 152
Metric_Fact_Statistics table, 148
Metric_Custom_Types table, 166
Metric_CustomTypes table
CustomMetricDescription, 167
CustomMetricName, 167
CustomSPName, 167
ValueUnits, 167
Metric_Dim_Dates table, 153
Date_ID, 153
Day, 153
DayOfWeek, 153
DayOfWeek_Num, 154
Hour, 154
Minute, 154
Month, 153
Quarter, 153
Time, 154
Week, 153
Year, 153
Metric_Dim_Flow_Instances table, 156
DataDefinition_ID, 156
DataDefinitionInstance_ID, 156
EndTime, 157
EndTimeLongMillis, 158
Flow_ID, 156
FlowInstance_ID, 156
Identifier, 156
StartTime, 157
StartTimeLongMillis, 157
Status, 158
WeightType, 156
Metric_Dim_Flows table, 155
Data_Definition_ID, 155
Flow_ID, 155
FlowName, 155
Status, 155
Metric_Dim_Groups table, 159
Group_ID, 159
GroupName, 159
GroupValue, 159

Metric_Dim_Metrics table, 151
CreatedDate, 152
Flow_ID, 151
GroupName, 152
IsDeleted, 152
LastStatisticsRecordLongMillis, 152
LastStatisticsRecordTime, 152
MeasurementPeriod, 152
Metric_ID, 151
MetricName, 151
ValueUnits, 151
Metric_Dim_Thresholds table, 159
CreatedDate, 166
Metric_Dim_Threshold table
AlertCriticalLevel, 163
AlertCriticalLevelDefnUnit, 165
AlertMajorLevel, 163
AlertMajorLevelDefnUnit, 165
AlertMinorLevel, 163
AlertMinorLevelDefnUnit, 165
AlertWarningLevel, 163
AlertWarningLevelDefnUnit, 166
CurrentAlertFromTime, 164
CurrentAlertFromTimeLongMillis, 164
CurrentAlertLevel, 164
CurrentAlertStatus, 164
LastCheckTime, 163
LastCheckTimeLongMillis, 163
Metric_ID, 160
MetricThreshold_ID, 160
StagingAlertLevel, 165
ThresholdColumnName, 161
ThresholdDescription, 160
ThresholdMessage, 160
ThresholdName, 160
ThresholdTest, 162
ThresholdType, 161
Metric_Fact_Alerts table, 138
AlertLevel, 139
AlertStatus, 139
Date_ID, 138
Flow_ID, 138
FlowInstance_ID, 138
IsAcknowledged, 139
MetricAlert_ID, 138
MetricThreshold_ID, 138
RaisedTime, 139
RaisedTimeLongMillis, 139
Time, 138
TimeLongMillis, 138
Value, 139
Metric_Fact_Statistics table, 147
AverageValue, 149
CountTotal, 149
Date_ID, 147
Flow_ID, 147
Group_ID, 148
IsLatest, 148
MaximumValue, 149
MeasurementPeriod, 148
Metric_ID, 147
MetricStatistic_ID, 147
MinimumValue, 149
StatisticsType, 148
StdDevValue, 149
SumSquareValue, 149
SumValue, 149
Throughput, 149
Time, 147
TimeLongMillis, 148
WeightAverageValue, 150
WeightStdDevValue, 150
WeightSumSquareValue, 150
WeightSumValue, 150
WeightThroughput, 150
WeightTotal, 149
Metric_Fact_Values table, 141
MetricValue_ID, 141

Metric_Fact_Value table
Date_ID, 141
Deadline, 143
DeadlineLongMillis, 143
DeadlineOverdue, 145
Flow_ID, 141
FlowInstance_ID, 141
Group_ID, 143
Idx, 144
LastUpdateTime, 145
LastUpdateTimeLongMillis, 145
Metric_ID, 141
StartTime, 144
StartTimeLongMillis, 144
Status, 142
TimeCompleted, 142
TimeCompletedLongMillis, 142
Value, 143
Weight, 142
Metric_Generate_Errors table, 167
Metric_ID
Metric_Dim_Metrics table, 151
Metric_Dim_Thresholds table, 160
Metric_Fact_Statistics table, 147
Metric_Fact_Value table, 141
Metric_Values
Flow_Instance, 170
table
EndTime, 170
EndTimeLongMillis, 170
Idx, 171
Name, 169
StartTime, 170
StartTimeLongMillis, 170
Time, 171
Type, 169
Value, 170
Metric_Values table, 169
MetricAlert_ID
Metric_Fact_Alerts table, 138
Metric data
HPBPI database, 20
Metric Definer
creating an entities file for the Business Availability Center, 81
Metric definer
architecture, 29
Metric Engine
architecture, 31
HPBPI Server, 31
MetricName
Metric_Dim_Metrics table, 151
Metrics table, 168
Metric probe
AVAILABILITY, 118
RESPONSE_TIME, 118
SETUP_TIME, 118
TBN_DURATION_HIGH, 118
TBN_DURATION_LOW, 118
TRANSFER_TPUT, 118
Metrics table
EndCondition, 169
EndNode, 169
Flow_ID, 168
MetricName, 168
MetricType, 168
StartCondition, 169
StartNode, 168
MetricStatistics_ID
Metric_Fact_Statistics table, 147
MetricThreshold_ID
Metric_Dim_Thresholds table, 160
Metric_Fact_Alerts table, 138
Metric threshold data
HPBPI database, 21
Metric thresholds
grammar rules, 205

MetricType
Metrics table, 168
MetricValue_ID
Metric_Fact_Values table, 141
MinimumValue
Metric_Fact_Statistics table, 149
Minute
Metric_Dim_Dates table, 154
Model_ID
Data_Object-Instance table, 194
Data_Objects table, 192
Modeler
Advanced conditions grammar, 210
Complete on first assignment grammar, 210
Complete on transition grammar, 210
Flow definition language, 205
functional return values for flow definitions, 208
grammar description, 206
grammar rules, 205
progression rules
defining for flows, 210
Start and complete on transition grammar, 210
String values for Flow definitions, 208
Modeling IT processes
using the Business Availability Center, 66
Model Repository
data and the HPBPI database, 21
description, 41
Monitoring Business Availability Center, 51
Monitoring HPSD processes, 56
Monitoring SOAM Manager, 52
Month
Metric_Dim_Dates table, 153
MY BAC
and HPBPI, 81
creating for BPI, 84
for HPBPI, 70

N

Name
Data_Objects table, 192
Metric_Values table, 169
Node2Resources
Resource_ID, 189
table
Node_ID, 189
Node2Resources table, 189
Node_ID
Node2Resources table, 189
Node_Instance table, 182
Nodes table, 179
Node_Instance_CompletedTimes table, 185
CompletedTime, 185
CompletedTimeLongMillis, 185
Idx, 185
NodeInstance_ID, 185
Node_Instance_StartedTimes
NodeInstance_ID, 184
table, 184
Idx, 184
StartedTime, 184
StartedTimeLongMillis, 184

Node_Instance table
EndTime, 182
EndTimeLongMillis, 182
EndTimeRecorded, 182
EndTimeRecordedLongMillis, 183
FlowInstance_ID, 182
Node_ID, 182
NodeInstance_ID, 182
ResourceStatus, 183
StartTime, 182
StartTimeLongMillis, 182
Status, 183
Table
Node_Instance, 182
NodeDescription
Nodes table, 179
NodeInstance_ID
Node_Instance_CompletedTimes table, 185
Node_Instance_StartedTimes, 184
Node_Instance table, 182
Node instance probe
ACTIVE_INSTANCES_HIGH, 119
ACTIVE_INSTANCES_LOW, 119
ACTIVE_VALUES_HIGH, 120
ACTIVE_VALUES_LOW, 119
AVAILABILITY, 119
INSTANCE_TPUT_HIGH, 120
INSTANCE_TPUT_LOW, 120
RESPONSE_TIME, 119
SETUP_TIME, 119
TRANSFER_TPUT, 119
Username, 113
VALUE_TPUT_LOW, 120
Node instances
C_OVBPI_NODE - OVBPI_NODE probe, 112
Node instances probe
OVBPI_DB_Password, 113
OVBPI_DB_User_Name, 113
OVBPI_Flow_Namet, 114
OVBPI_Identifier, 113
OVBPI_Node_Name, 114
Port, 113
Target Host, 113
NodeName
Nodes table, 179
Nodes table, 179
ActiveCount, 179
ActiveWeight, 180
AvrgTime, 180
Flow_ID, 179
InstanceRate, 180
LastRateUpdate, 181
LastRateUpdateLongMillis, 181
Node_ID, 179
NodeDescription, 179
NodeName, 179
NodeType, 179
ResourceStatus, 181
SampleCount, 180
TotalCount, 179
TotalTime, 180
TotalWeight, 180
WeightRate, 180
X_Pos, 179
NodeType
Nodes table, 179
Notification Server
Administration Console, 42
architecture, 34
component of BPI Server, 34
email alerts and HP Service Desk, 37
HPBPI Server component, 23
Velocity email templates, 36
XSLT email templates, 36

NULL
when used within an expression, 221
Numeric
data type coercion rule, 218

O

Objective information
C_OVBPI_FLOW_PROBE, 116
C_OVBPI_METRIC_PROBE, 118
C_OVBPI_NODE_PROBE, 119
for OVIS configuration, 115
openadaptor
event store, 195
handling HPBPI business events, 37
Operation Service Source
configuring for Business Availability Center
HPBPI
Business Availability Center Operation Service Source, 76
OVBPI_DB_Password
Flow Instances probe, 110
Flow Metric probe, 112
Node instances probe, 113
OVBPI_DB_User_Name
Flow Instances probe, 109
Flow Metric probe, 111
Node instances probe, 113
OVBPI_Flow_Name
Flow Instances probe, 110
Flow Metric probe, 112
Node instances probe, 114
OVBPI_Identifier
Flow Instances probe, 109
Flow Metric probe, 111
Node instances probe, 113
OVBPI_Metric_Name
Flow Metric probe, 112
OVBPI_Node_Name
Node instances probe, 114
OVIS
alarms and SLOs, 115
alarms database tables, 122
custom probes, 97, 104, 105
enabling integration, 122
event receiver for HPBPI Server, 28
integration with HPBPI, 97
Objective information, 115
Probes custom dialogs, 43
reporting on operational services, 97, 104
reporting SLO and SLA violations, 97, 104
OVIS and HPBPI
design-time integration, 99
integration, 99
run-time integration, 101
OVIS custom probe, 53
C_OVBPI_FLOW - OVBPI_FLOW probe, 108
C_OVBPI_METRIC - OVBPI_METRIC probe, 108
C_OVBPI_NODE - OVBPI_NODE probe, 108
Flow Instances probe, 108
Flow Metric probe, 110
Node instances probe, 112
OvpbiProbe.cfg
configuring for HPBPI, 123

P

Patient
DBusMH_Patient table, 199
PatientId
DBusMH_Patient table, 197

PatientStatus
DBusMH_Patient table, 197
Port
Flow Instances probe, 109
Flow Metric probe, 111
Node instances probe, 113
Primary_entity
Flow_Instances table, 176
Flows table, 174
Primary_entity_ inst
Flow_Instances table, 176
Probe locations
for HPBPI custom probes, 122
Progression rule
after(), 212
before(), 212
case sensitivity for expressions, 216
changed(), 212
created(), 213
in(), 213
methods for, 212
terminated(), 213
Propagating events, 39
PubUniqueId
DBusMH_Patient table, 198

Q

Quarter
Metric_Dim_Dates table, 153

R

RaisedTime
Metric_Fact_Alerts table, 139
RaisedTimeLongMillis
Metric_Fact_Alerts table, 139
RejectReason
DBusMH_Patient table, 198
Reporting operational status using OVIS, 104
Reporting SLO and SLA violations using OVIS, 104
Repository Explorer
architecture, 33
component of HPBPI Server, 33
RepositoryID
Data_Objects table, 192
Flows table, 174
RepositoryRevision
Data_Objects table, 193
Flows table, 175
Resource_ID
Node2Resources, 189
Resources table, 187
ResourceDescription
Resources table, 187
ResourceName
Resources table, 187
Resources table, 187
DeploymentStatus, 188
LastChange, 188
LastChangeLongMillis, 188
Resource_ID, 187
ResourceDescription, 187
ResourceName, 187
ResourceType, 187
RootCause, 187
ResourceStatus
Node_Instance table, 183
Nodes table, 181
ResourceType
Resources table, 187
RESPONSE_TIME, 116, 118, 119
Retried
DBusMH_Patient table, 199

RootCause
Resources table, 187
Rules
for Modeler grammar, 205
for thresholds, 205

S

SampleCount
Data_Objects table, 192
Flows table, 173
Nodes table, 180
SAP integration, 58
Schema
Arcs
table in schema, 186
Business Entity, 191
Business Entity Data Type, 191
Business Entity Keys, 191
Business Event Handler, 195
Business process metrics, 136
definition, 133
dimensional versus entity model, 136
Flow_Instances, 175
Flows, 172
Metric_Custom_Types, 166
Metric_Dim_Dates, 153
Metric_Dim_Flow_Instances, 156
Metric_Dim_Flows, 155
Metric_Dim_Groups, 159
Metric_Dim_Metrics, 151
Metric_Dim_Thresholds, 159
Metric_Fact_Alerts, 138
Metric_Fact_Statistics, 147
Metric_Fact_Values, 141
Metric_Generate_Errors, 167
Metric_Values, 169
Node2Resources
table, 189
Node_Instance_CompletedTimes, 185
Node_Instance_StartedTimes, 184
Node_Instance table, 182
Nodes table, 179
Resources table, 187
Schema for HPBPI data, 133
Security
architecture, 44
Select Access, 44
Tomcat Servlet Engine, 44
Select Access
security, 44
Self-Healing Services
architecture, 62

Service Desk
and Notification Server, 37
and Notification Server email alerts, 37
custom field definition, 128
integrating with HPBPI, 126
integration
through adapter, 125
integration through Business Process Dashboard, 125
mapping services, 128
monitoring by HPBPI, 56
Service Desk processes
BPI monitoring, 56
Service Level Agreements
for HPBPI, 121
SETUP_TIME, 116, 118, 119
Single signon
for HP Operations Dashboard links to HPBPI, 61
SLA
for BPI, 121
SLO and SLA violations
using OVIS, 54
SLO information, 116, 117, 118, 119, 120
C_OVBPI_FLOW_PROBE, 116
C_OVBPI_METRIC_PROBE, 118
C_OVBPI_NODE_PROBE, 119
SOAM
monitoring by HPBPI, 52
SOA Manager
business events, 90, 93
client colleague for HPBPI Server, 27
configuring access to adapter, 94
integration with HPBPI, 89, 90
status of Business Services, 90
Source
Arcs table, 186
SourceAppName
DBusMH_Patient table, 198
StagingAlertLevel
Metric_Dim_Thresholds table, 165
Start and complete on transition
grammar rules, 210
StartCondition
Metrics table, 169
Start condition
after(), 212
before(), 212
changed(), 212
created(), 213
in(), 213
methods for, 212
terminated(), 213
StartedTime
Node_Instance_StartedTimes table, 184
StartedTimeLongmillis
Node_Instance_StartedTimes table, 184
StartNode
Metrics table, 168
StartTime
Data_Object-Instance table, 194
Flow_Instances table, 177
Metric_Dim_Flow_Instances table, 157
Metric_Fact_Value table, 144
Metric_Values table, 170
Node_Instance table, 182
StartTimeLongMillis
Data_Object-Instance table, 194
Flow_Instances table, 177
Metric_Dim_Flow_Instances table, 157
Metric_Fact_Value table, 144
Metric_Values table, 170
Node_Instance table, 182
Statistics facts dimension, 146

StatisticsType
Metric_Fact_Statistics table, 148
STATUS
EVENT_STORE table, 196
Status
Data_Object-Instance table, 194
Data_Objects table, 192
Flow_Instances table, 178
Flows table, 174
Metric_Dim_Flow_Instances table, 158
Metric_Dim_Flows table, 155
Metric_Fact_Value table, 142
Node_Instance table, 183
Resources table
Resources table
Status, 187
StdDevValue
Metric_Fact_Statistics table, 149
String
data type coercion rule, 218
data type definition, 135
operator for coercion, 219
values for flow definitions, 208
Subclass, 178
Data_Objects table, 193
Flow_Instances table, 178
Flows table, 174
Subject
DBusMH_Patient table, 197
SumSquareValue
Metric_Fact_Statistics table, 149
SumValue
Metric_Fact_Statistics table, 149

T

Table
Arcs, 186
Data_Object-Instance, 193
EVENT_STORE, 195
Flow_Instances, 175
Flows, 173
Metric_Custom_Types, 166
Metric_Dim_Dates, 153
Metric_Dim_Flow_Instances, 156
Metric_Dim_Flows, 155
Metric_Dim_Groups, 159
Metric_Dim_Metrics, 151
Metric_Dim_Thresholds, 159
Metric_Fact_Alerts, 138
Metric_Fact_Statistics, 147
Metric_Fact_Values, 141
Metric_Generate_Errors, 167
Metric_Values, 169
Node2Resources, 189
Node_Instance_CompletedTimes, 185
Node_Instance_StartedTimes, 184
Nodes, 179
Resources, 187
Target Host
Flow Instances probe, 109
Flow Metric probe, 111
Node instances probe, 113
TBN_DURATION_HIGH, 118
TBN_DURATION_LOW, 118
Text
data type definition, 135
Threshold
grammar rules, 205
ThresholdColumnName
Metric_Dim_Thresholds table, 161
Threshold definitions, 205
ThresholdDescription
Metric_Dim_Thresholds table, 160

ThresholdMessage
Metric_Dim_Thresholds table, 160
ThresholdName
Metric_Dim_Thresholds table, 160
ThresholdTest
Metric_Dim_Thresholds table, 162
ThresholdType
Metric_Dim_Thresholds table, 161
Throughput
Metric_Fact_Statistics table, 149
throughput
Data Sample for Business Availability Center, 86
Time
Metric_Dim_Dates table, 154
Metric_Fact_Alerts table, 138
Metric_Fact_Statistics table, 147
Metric_Values table, 171
TIME_RX
EVENT_STORE table, 196
TIME_TX
EVENT_STORE table, 196
TimeCompleted
Metric_Fact_Value table, 142
TimeCompletedLongMillis
Metric_Fact_Value table, 142
TimeLongMillis
Metric_Fact_Alerts table, 138
Metric_Fact_Statistics table, 148
Tomcat Servlet Engine
security, 44
TotalCount
Nodes table, 179
TotalFlows
Flows table, 173
TotalInstances
Data_Objects table, 192
TotalTime
Nodes table, 180
TotalWeight
Nodes table, 180
TRANSFER_TPUT, 116, 118, 119
Treated
DBusMH_Patient table, 198
TreatedUser
DBusMH_Patient table, 199
Type
Arcs table, 186
Metric_Values table, 169

U

Username
Flow Instances probe, 109
Flow Metric probe, 111
Node instance probe, 113

V

Value
Metric_Fact_Alerts table, 139
Metric_Fact_Value table, 143
Metric_Values table, 170
VALUE_TPUT_HIGH, 117
VALUE_TPUT_LOW, 117, 120
Value facts dimension, 140
ValueUnits
Metric_CustomTypes table, 167
Metric_Dim_Metrics table, 151
variable-length binary
data type definition, 135
variable-length text
data type definition, 135

W

Week
Metric_Dim_Dates table, 153

Weight
Flow_Instances table, 176
Metric_Fact_Value table, 142

Weight_type
Flow_Instances table, 177

WeightAverageValue
Metric_Fact_Statistics table, 150

WeightRate
Nodes table, 180

WeightStdDevValue
Metric_Fact_Statistics table, 150

WeightSumSquareValue
Metric_Fact_Statistics table, 150

WeightSumValue
Metric_Fact_Statistics table, 150

WeightThroughput
Metric_Fact_Statistics table, 150

WeightTotal
Metric_Fact_Statistics table, 149

WeightType
Metric_Dim_Flow_Instances table, 156

X

X_Pos
Nodes table, 179

Y

Y_Pos, 179
Nodes table, 179

Year
Metric_Dim_Dates table, 153