

HP

Business Process Insight

For the Windows® Operating System

Software Version: 7.50

Reference Guide

Document Release Date: June 2008

Software Release Date: June 2008



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

© Copyright 2007-2008 Hewlett-Packard Development Company, L.P.

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://h20230.www2.hp.com/selfsolve/manuals>

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 Features	13
	Business Process Dashboard Structure	14
	Annotations	14
	The Dashboard and Business Process Monitors	15
	Email Notifications and the Dashboard	15
	Intervention Client	16
	HPBPI Database	18
	Process Data	19
	Business Process Monitor Data	20
	Monitor Threshold Definitions	21
	Event Hospital Data	21
	Model Repository Data	21
	Email Notification Data	22
	HPBPI Server	23
	Business Impact Engine	25
	Business Process Monitor Definer	29
	Monitor Engine	31
	Repository Explorer	34
	Notification Server	35
	Business Event Handler	38
	JMS Business Event Handler	42
	Modeler	44
	Repository Server	45
	Administration Consoles and Interfaces	46

HPBPI Security	47
Where to go Next	47
3 Integrating with HPBPI	49
HPBPI Integration Points	50
HPBPI Adapters for HP BTO Software Product's Integration	54
Service Source for Business Availability Center	54
HPBPI SOA Manager Service Source	56
HPBPI Operations Manager Adapter	56
HPBPI Accelerator for SAP Applications	59
iWay Integration	60
HP Operations Dashboard	61
Self-Healing Services	63
4 HPBPI and HP Business Availability Center	65
Overview of HPBPI and Business Availability Center Integration	66
Design-Time Integration	69
Run-Time Integration	71
Configuration Items Stored in the UCMDB	74
Business Availability Center and Load Balancing	76
Configuring HPBPI to Integrate with Business Availability Center	77
Business Availability Center General Settings	78
Configuring a Business Availability Center Data Sample Destination	80
Using Business Availability Center as a Source of Operational Service Status Data	82
Security and Single Sign on	84
Select Access Authorization	84
Servlet Engine Authorization	84
Using SSL for Connections with Business Availability Center	86
Business Availability Center System Configuration	87
Creating a My BAC Application for HPBPI	88
SLM and Reporting	89
Description of Data Samples Sent to Business Availability Center	90
5 HPBPI and TransactionVision	93
Overview of TransactionVision Integration	94
Design-Time Integration	96

Run-Time Integration	98
Configuring HPBPI to Integrate with TransactionVision	100
TransactionVision Configuration Tasks	100
HPBPI Configuration Tasks	104
Import TransactionVision Events to Model Repository	105
Configure a JMS Business Event Handler	113
Example Configuration for the TransactionVision System	117
Create a JMS Queue for the Integration	117
Set up the JNDI Entries on the TransactionVision System	118
Configure TransactionVision to Send Events to the JMS Queue	120
Example Configuration for the HPBPI System	121
Set up JNDI Entries on the HPBPI System	121
Configure a JMS Business Event Handler for HPBPI	123
6 HPBPI and SOA Manager	127
SOA Manager and HPBPI Integration	128
Business Events and SOA Manager	131
Configuring Access to the SOA Manager Adapter	132
A Expression Grammar in Process, Data and Filter Definitions	135
Grammar	136
Function Return Values	138
Functions for Dates and Times	138
Functions for Identifying String Values	138
Functions for All Property Types	139
Process Progression Rules	140
Methods for Progression Rules	142
Case Sensitivity for Expressions	146
Expression Properties	146
Expressions with String Constants	146
B Coercion Rules	147
Assignments	148
Expressions	149

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 process 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 Monitors and Monitor Threshold details through the Business Availability Center Dashboard.

- [Chapter 5, HPBPI and TransactionVision](#)

This chapter provides details of how to integrate HPBPI and TransactionVision. It describes how to import transaction event definitions into the HPBPI Server in order that you can link them into your business processes using the HPBPI Modeler.

- [Chapter 6, 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 HPBPI business processes.

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*.

- [Appendix A, Expression Grammar in Process, Data and Filter Definitions](#)

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

- [Appendix B, 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.

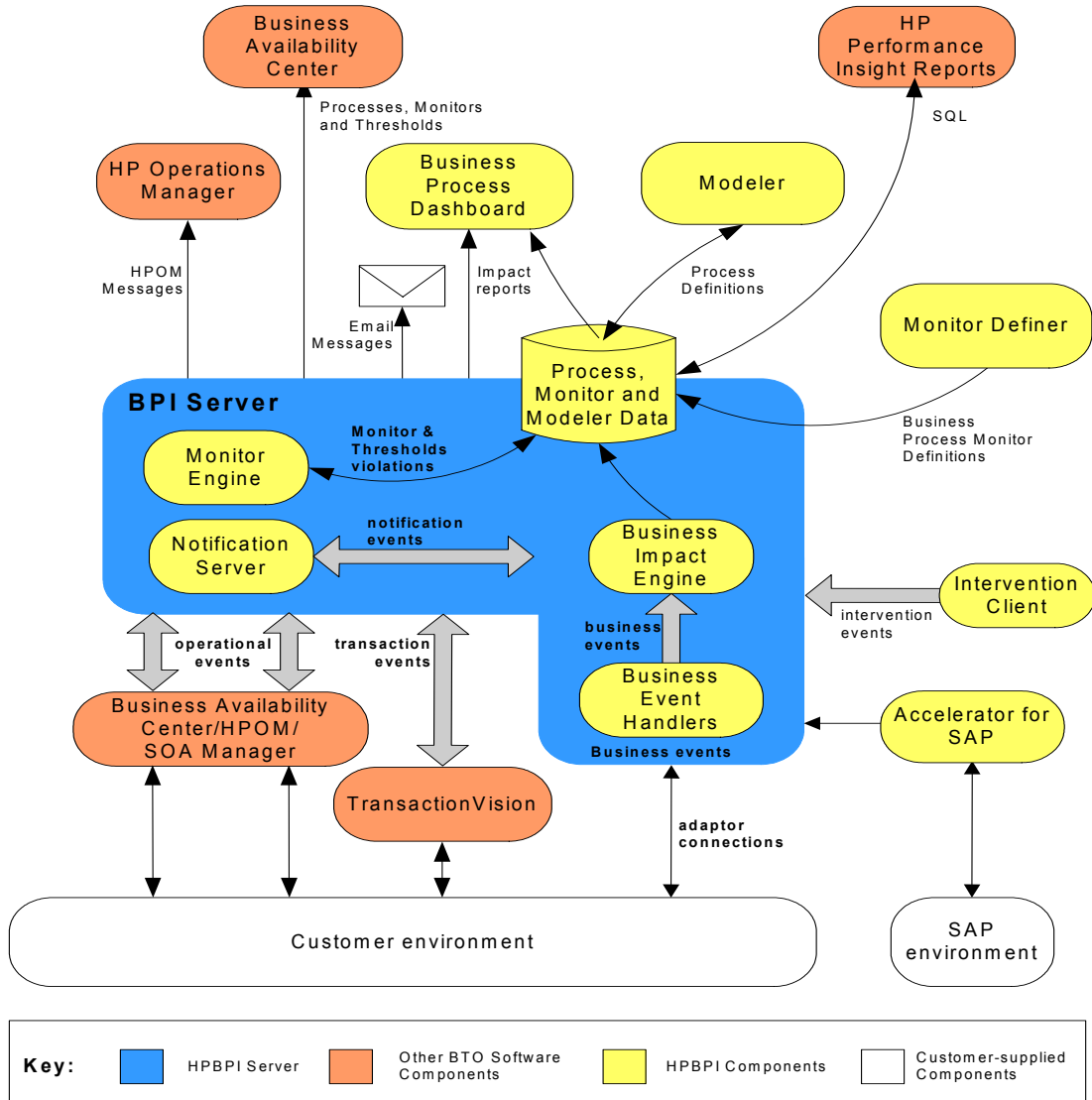
Specifically, the chapter covers the following topics:

- 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 Features](#) on page 13.
- The HPBPI Intervention Client, which provides you with access to active business processes and process 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 Monitor Definer, which you use to define business process monitors and monitor thresholds for your processes; see section [Business Process Monitor Data](#) on page 20 and section [Monitor 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 44.
- The administration GUIs and consoles used within the HPBPI system; see section [Administration Consoles and Interfaces](#) on page 46.

High-Level Component Architecture

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

Figure 1 HPBPI Architecture



Business Process Dashboard Features

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

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

- An overall business health scorecard, which provides an overall view of the health of business processes that you have deployed.
- Process instance information related to the processes that you have deployed.
- Information relating to the status of the operational Services that your processes are linked to within Business Availability Center, HP Operations Manager and SOA Manager.
- 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.



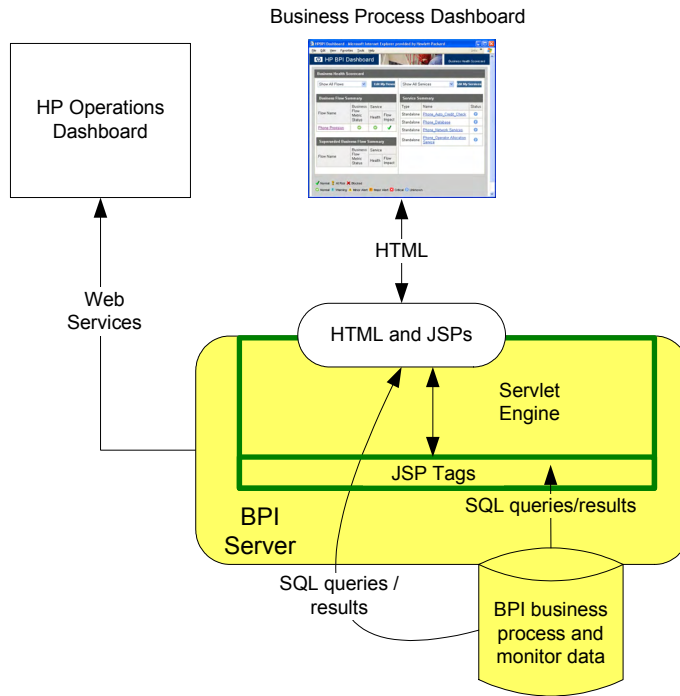
If you have installed the HPBPI Business Process Dashboard on a different system to the HPBPI Server, you do not get information about the status of your license.

If you have not enabled an integration between HPBPI and Business Availability Center, you view the Business process monitor information that you have defined using the Business Process Dashboard:

Business Process Dashboard Structure

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



Annotations

It is also possible to annotate your processes. This enables you to create queries and enhance the process data with existing business data.

Creating annotations is described in the *Business Process Insight System Administration Guide*.

The Dashboard and Business Process Monitors

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

- overall health of the business processes relative to the business process monitors that are deployed within HPBPI.
- business process monitors and monitor thresholds that you have configured, including the violation status and individual violations.
- details of the individual process instances and monitor instances within the HPBPI system.
- historical statistical information relating to the business process monitors.

This business process monitor data is displayed automatically through the HPBPI Dashboard as a result of your defining the business process monitors and business thresholds for your processes.

Business process monitors and monitor threshold definitions are set up using the Monitor Definer, which is described in more detail in the *HP Business Process Insight Integration Training Guide - Defining Business Process Monitors*.

Email Notifications and the Dashboard

In addition to proactively monitoring business processes through the Business Process Dashboard, you can configure HPBPI to send email notifications relating to your processes through the Notification Server (see section [Notification Server](#) on page 35). These email notifications contain information relating to the business processes 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 processes.

If you have configured the HPBPI integration with Business Availability Center to be enabled, email notifications are configured using Business Availability Center and not the Email Notification Server. Refer to the Business Availability Center documentation for details of sending violation notifications using email.

Intervention Client

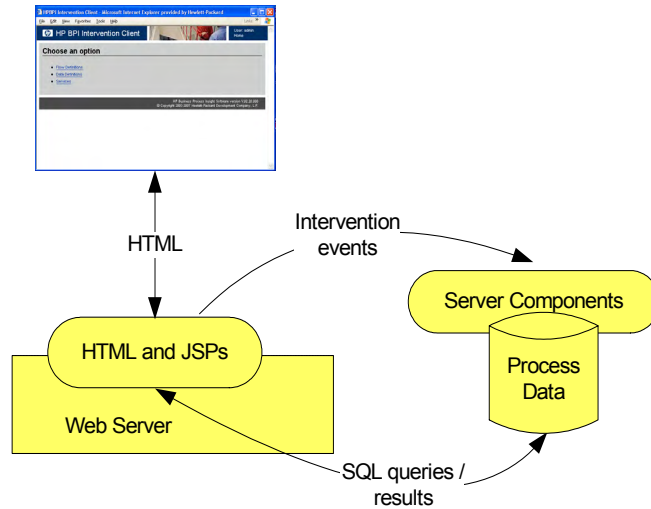
The Intervention Client enables you to access processes that you have deployed in order to modify or delete Process instances and their associated Data instances. You might need to do this to resolve problems with the process, or its data, usually after a period of new development or where progression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

- manually progress and delete process instances.
- update and delete Data Definition 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 Business Availability Center or HP Operations Manager; it makes changes only to the recorded status within HPBPI. In addition, if you make changes to HP Operations Manager, SOA Manager, or Business Availability service statuses through the Intervention Client, these changes are overwritten when HPBPI next polls the source of the service for updates.

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

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 processes. 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:

- Processes and any related data
- Business process monitors
- Monitor threshold definitions
- Monitor threshold violations
- Monitor statistics
- Event hospital data, including the business event data
- Operational service status data
- Model Repository data
- Email notification data

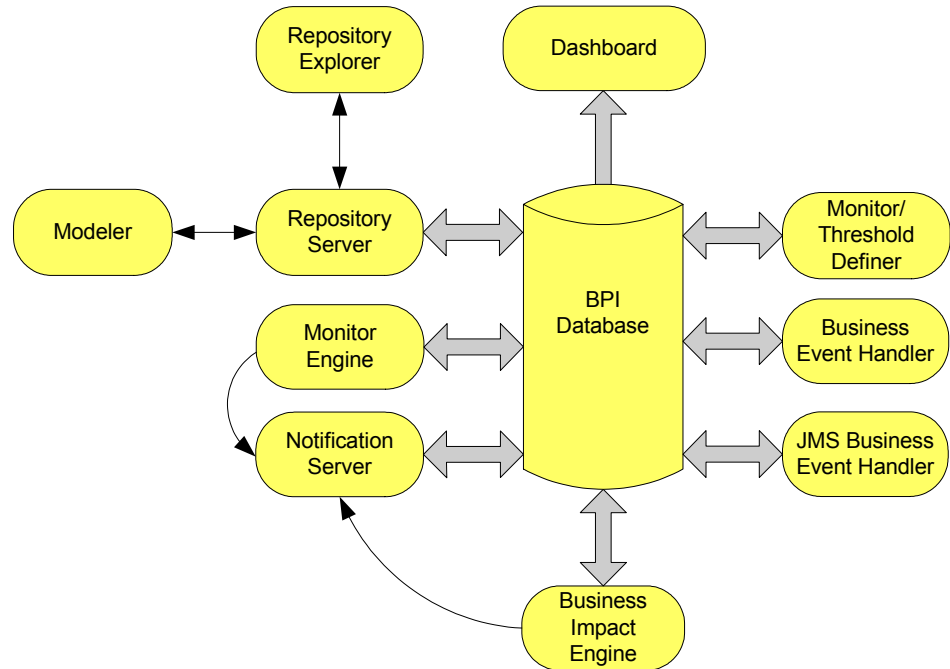
This information is used by the HPBPI components for monitoring and progressing process instances, plus reporting process impact data, threshold violations and business process monitors through the Dashboard. It is also used by the Business Impact Engine to identify the Process, Data and Service definitions 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.

[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 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.

Process 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 Process definitions, within the HPBPI system. These definitions comprise status information required by the Business Impact Engine for:

- Processes
- Data
- Services

These Process, 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 processes is described in more detail in the *HP Business Process Insight Integration Training Guide - Modeling Processes*.

Business Process Monitor Data

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

You can also use reporting applications, such as HP Performance Insight, to access the monitor and statistical information in the database tables and generate reports from the data that is collected.

You can read more about the Monitor Definer and the monitor data in the *Business Process Insight Integration Training Guide - Defining Business Process Monitors*.

Monitor Threshold Definitions

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

Monitor threshold violations are shown:

- using the Business Process Dashboard
- using the Business Availability Center
- as email notifications using your email system
- as HP Operations Manager violations

You can read more about monitor thresholds and creating them using the Monitor Definer in the *Business Process Insight Integration Training Guide - Defining Business Monitors*.

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 Process 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 processes evaluated. The HPBPI Server is responsible for:

- Maintaining the Process, Data, Event and business process monitor 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 process context for the business.
- Monitoring services and reporting any change that causes an impact on the processes.
- Defining business process monitors and monitor thresholds.
- Monitoring monitor thresholds and reporting on the threshold violations.
- Validating the status of processes according to progression rules that you have specified.
- Sending email notifications for process impact violations that you have subscribed to.
- Sending email notifications for monitor threshold violations that have been subscribed to.

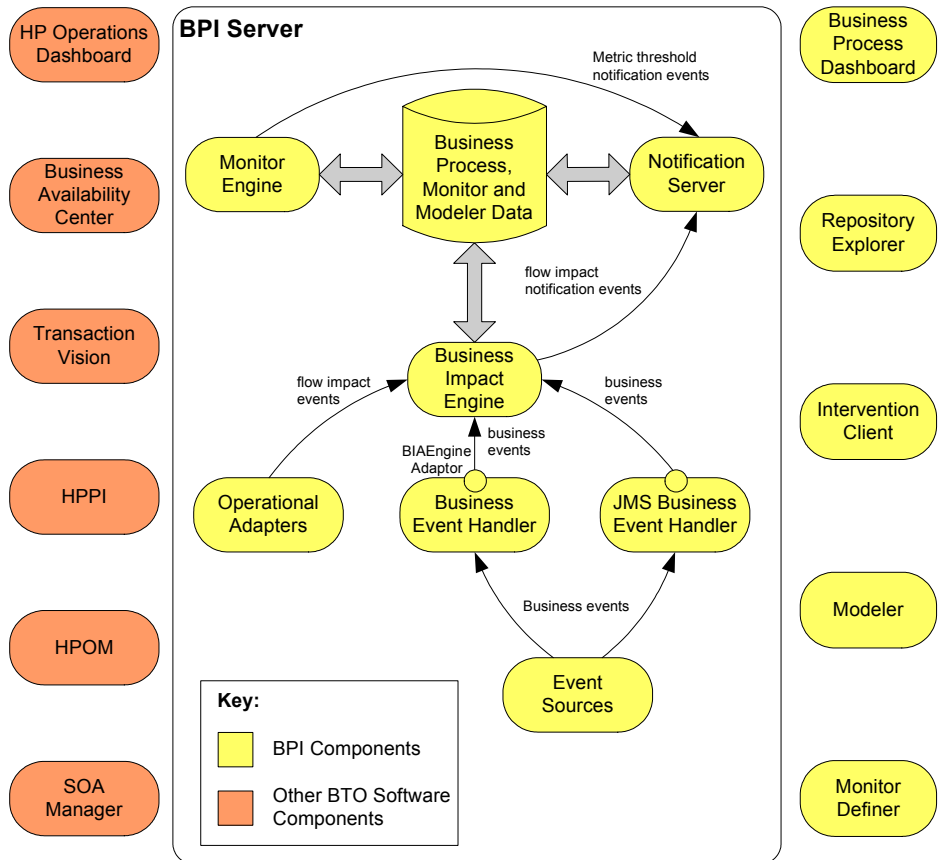
The components of the HPBPI Server that provide these functions are the:

- Business Impact Engine
- Monitor Engine
- Business Process Monitor Definer
- Repository Explorer
- Notification Server
- Business Event Handler
- JMS Business Event Handler

HPBPI can also accept service status 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

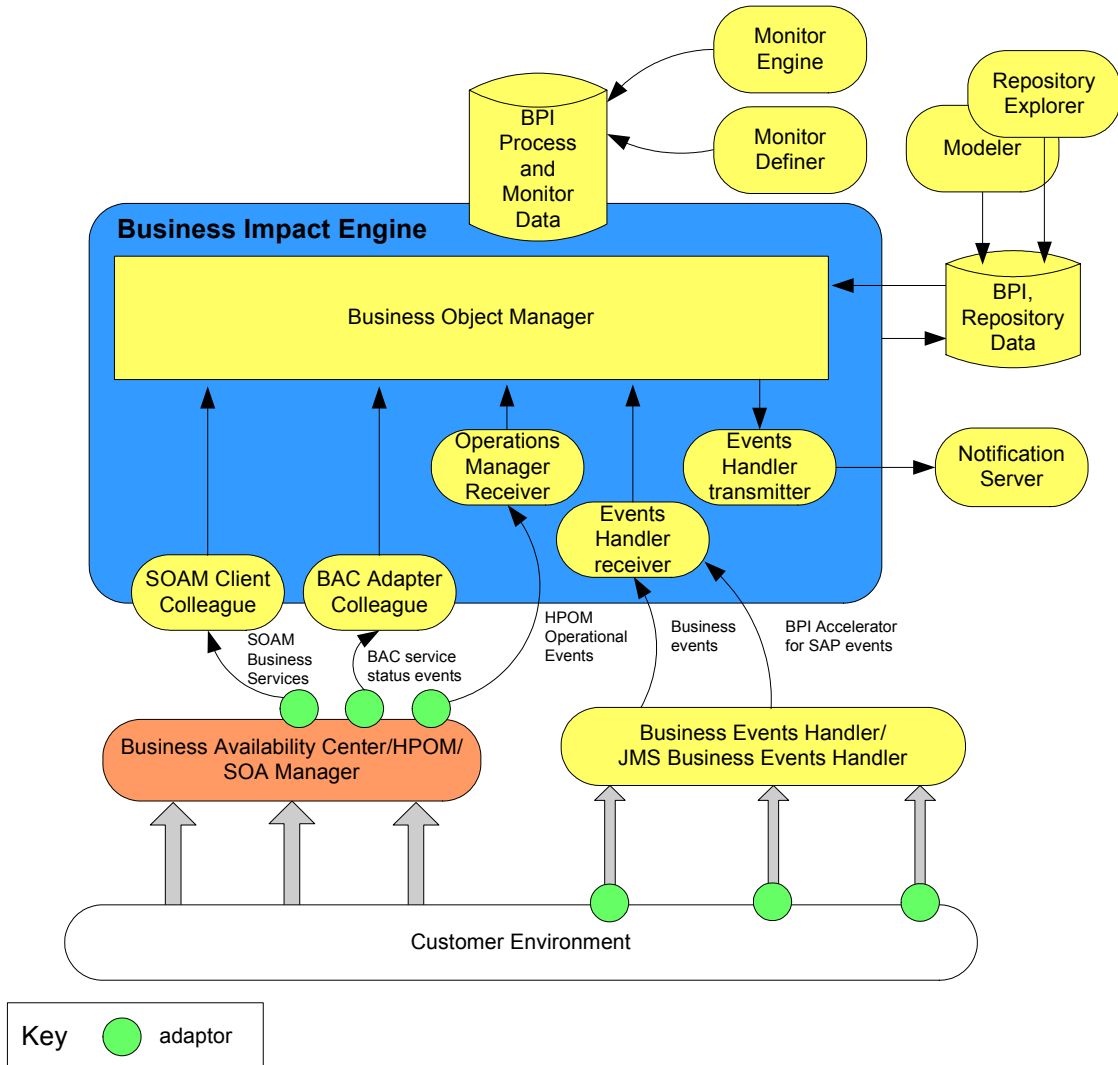


Business Impact Engine

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 process definitions, the Business Impact Engine raises process impact violations when an underlying operational service or business event, which is relied on by the process, falls outside defined thresholds.

Business Monitor Threshold violations are managed by the Monitor Engine as described in section [Monitor Engine](#) on page 31.

Figure 6 Business Impact Engine Architecture



When a process impact threshold violation is identified, the Business Impact Engine sends an impact violation to the Notification Server. In addition, the Business Process Dashboard checks the database for status changes resulting from these impact violations to present the violations to the business

manager. This enables you either to be notified, or to proactively monitor the status of your business processes, 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 Process definitions and instances. The business object manager receives:

- operational events from Business Availability Center and HP Operations Manager
- business events from the Business Event Handler
- internal events, such as process 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 BIE Model Cleaner and the BIE Process 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 process using the HPBPI Modeler.

- SOA Manager Client Colleague

Not strictly a component of the Business Impact Engine, this adapter 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

Not strictly a component of the Business Impact Engine, 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 steps in your HPBPI Business Processes.

- Event Receivers and Transmitters

The Event Receiver accepts business events, using RMI, as Java objects from the Business Event Handler, or the JMS Business Events 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 both the Business Event Handlers. This means that the Business Event Handlers do 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 Handlers wait 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 violations from the Business Object Manager, converts them into an RMI message and sends them to the Notification Server.

Business Process Monitor Definer

The Monitor Definer is a Web-based GUI that enables you to define business process monitors and monitor threshold definitions for deployed processes.

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



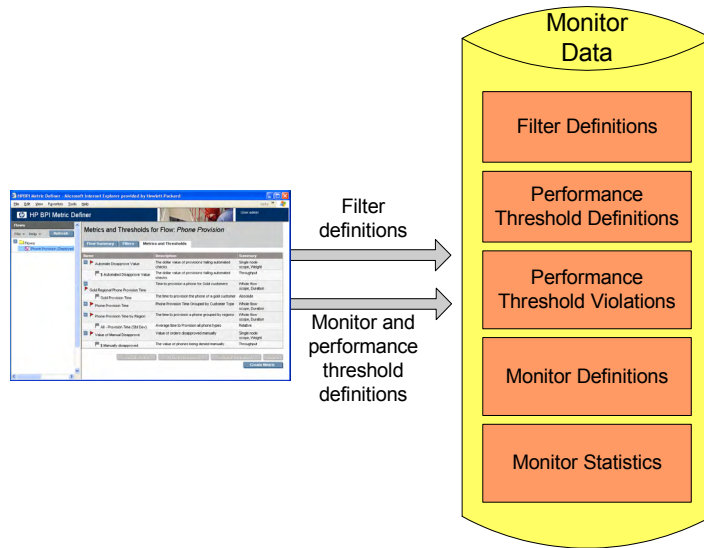
If you have configured HPBPI to integrate with Business Availability Center, the monitor and threshold statistics recorded by the Monitor Engine are not viewed using the Business Process Dashboard, they are viewed using Business Availability Center.

Within HPBPI, the tool for defining business processes is separated from the tool for defining business process monitors for these processes. You use the HPBPI Modeler to define your business processes and you use the Monitor Definer to define the business process monitors and monitor thresholds for these business processes. You define business process monitors for a business process after the process has been deployed; this provides more flexibility for defining and modifying business process monitors without the need to redeploy your business processes.

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 Monitor Definer relative to the Business Monitor data stored in the HPBPI database.

Figure 7 HPBPI Business Process Monitor Definer



You can define monitors for:

- Single steps
- Multiple steps
- The whole process

These monitors can be duration based or Process Value based. In addition, you can define a custom monitor.

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

You can also use the Monitor Definer to export Process definitions, including Business Process Monitors, to an XML file, and subsequently import these definitions.

How the Business Process Monitors definer interworks with the Monitor Engine is described in section [Monitor Engine](#) on page 31.

Monitor Engine

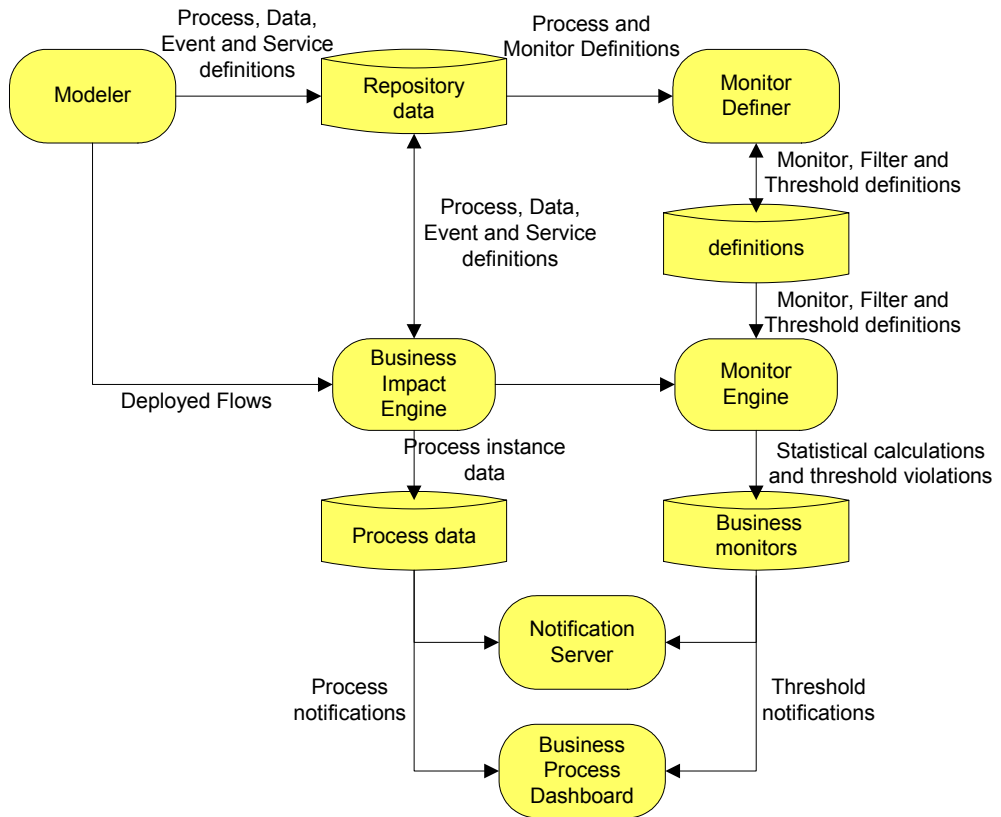
The Monitor Engine is the component of the HPBPI Server that analyzes and provides the statistical results from the business process monitor and monitor threshold data that you define. The Monitor Engine takes data about the individual instances from the Business Impact Engine and combines this data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



If you have configured HPBPI to integrate with Business Availability Center, no statistical monitors and thresholds can be defined in HPBPI. Instead, this data is sent as data samples to Business Availability Center where the statistics can be displayed and analyzed.

Figure 8 shows the relationship between the process impact data and the monitor data that you define.

Figure 8 Process Impact and Monitor Data Relationship



The business process monitors and monitor thresholds are configured through the Business Process Monitor Definer; see section [Business Process Monitor Definer](#) on page 29.

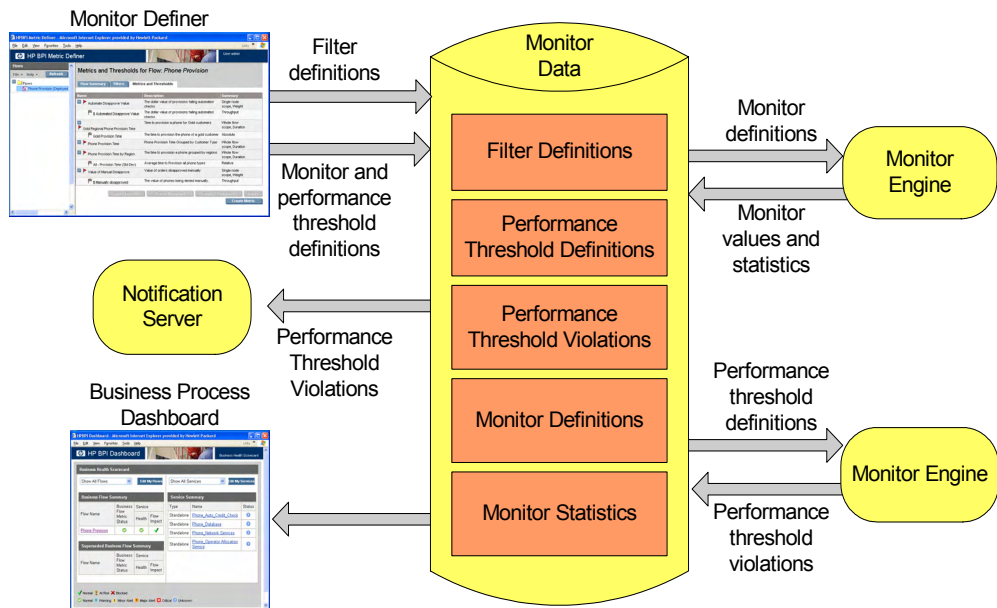
The Monitor Engine maintains business data for the business processes that are deployed according to the business process monitors that you have defined. It also maintains statistical information about the monitor thresholds that are defined.

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

▶ If you have enabled the integration between HPBPI and Business Availability Center, statistical thresholds are reported through the Business Availability Center Dashboard and not through the Business Process Dashboard. Instance thresholds continue to be reported through the Business Process Dashboard.

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

Figure 9 HPBPI Business Process Monitor Engine



The statistics that the Monitor Engine calculates are stored in HPBPI database tables and are used by the Business Process Dashboard. The database tables also hold information used to determine if any monitor thresholds have been violated. These violations are reported through the Business Process 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 process within the HPBPI Modeler, but not yet redeployed the process.

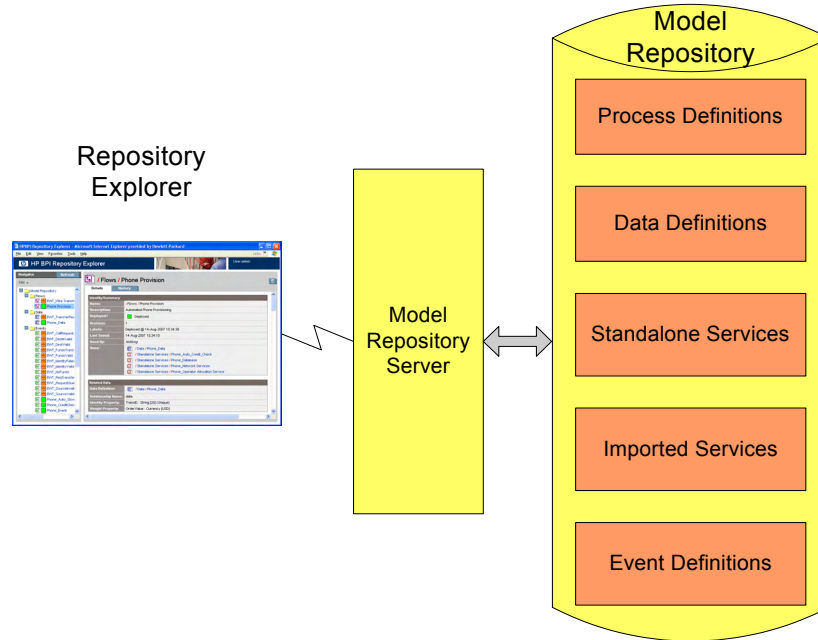
The Repository Explorer also provides access to information about all the superseded versions of the business process that are currently deployed, enabling you to view and export details of the superseded processes.

Using the Repository Explorer you can:

- View all the details of your currently deployed and superseded processes 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



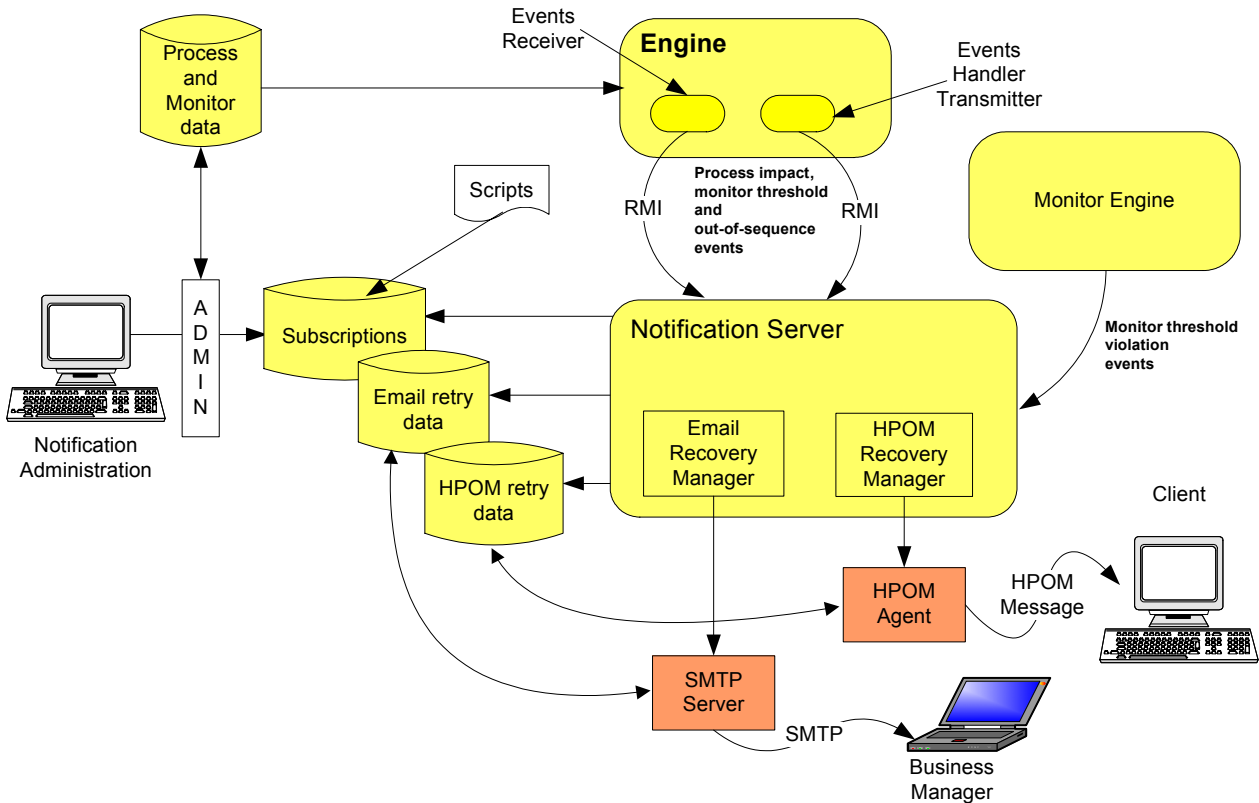
Notification Server

This component of the HPBPI Server is responsible for notifying you of the process-impact, out-of-sequence and monitor-threshold violations that you have configured through the Notification Server Administration Console. The Notification Server Administration is described in section [Administration Consoles and Interfaces](#) on page 46.

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

- process-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.
- monitor threshold violation notification events from the Monitor Engine.

The Notification Server sends these events as business violations 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 process. SLAs are filtered according to Customer SLA name, and SLOs are filtered according to Customer, Service Group and SLO name.

In addition to sending violations 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 processes 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.

- monitor 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 violations into the messages and configure how many email notifications that you receive for a particular monitor threshold violation.

The default templates are provided as part of the HPBPI installation and instructions for creating your own templates and configuring monitor 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 notifications to HP Operations Manager, you enable HP BTO software users to receive these notifications in the clients of their choice. 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.

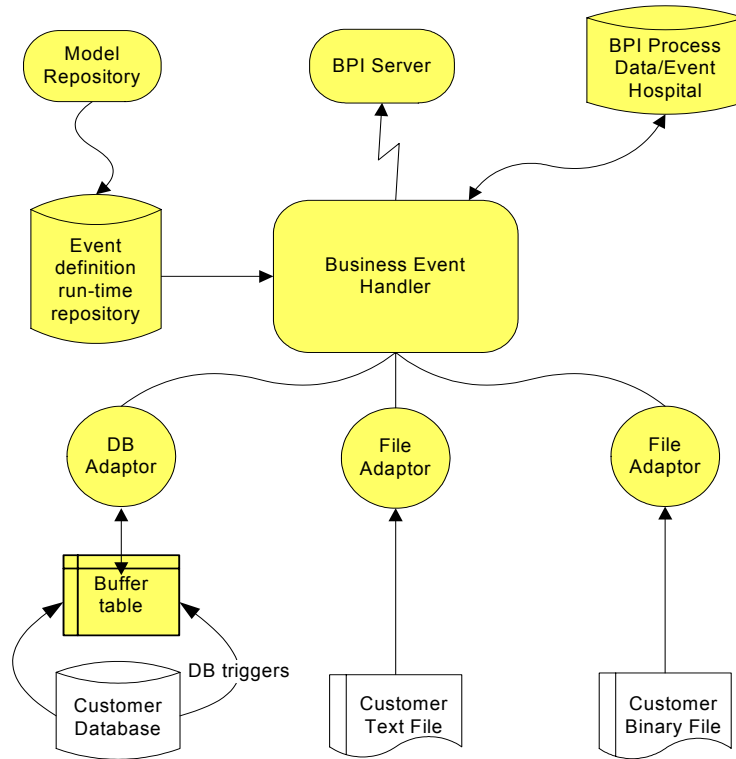
Business Event Handler

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 process 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 Processes. 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 messaging 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 processes to be re-evaluated, and where appropriate the processes are progressed.

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

At the same time, the Business Process Dashboard is polling the HPBPI Process 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 process, or processes, 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.

JMS Business Event Handler

The JMS Business Event Handler is an extension to the Business Events Handler described in section [Business Event Handler](#) on page 38.

The JMS Business Event Handler provides the Business Impact Engine with business events, in the form of XML documents, from a JMS-enabled system.

As described above the Business Event Handler is built using the openadaptor adaptor framework. An additional JMS Source has been added to the Business Event Handler in order to provide the additional capability of receiving business events from a JMS-enabled system.



HPBPI supports the JMS message type `javax.jms.TextMessage`, where the content of the text is an XML file in a particular format. This means you must be integrating with an application that sends and receives this message type if you want to use the JMS Business Event Handler. One example of such an application is TransactionVision.

Configuring the JMS Business Event Handler is described in the *Business Process Insight System Administration Guide*.

Figure 13 JMS Business Event Handler

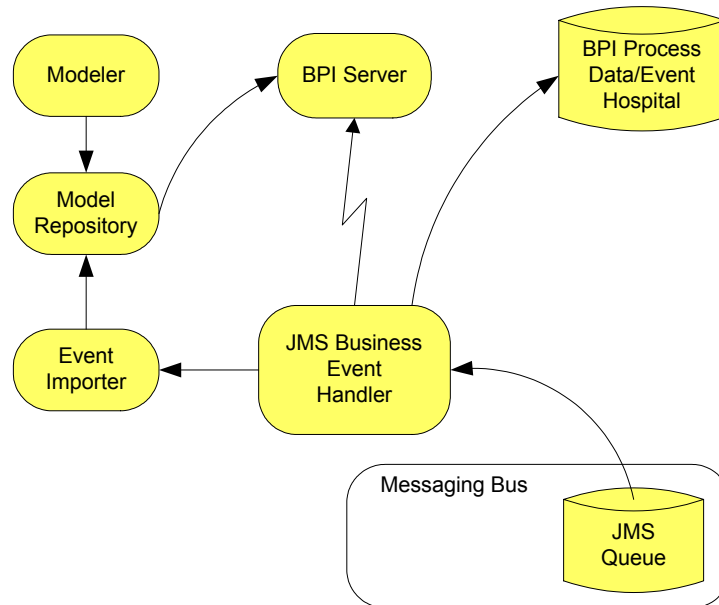


Figure 13 shows how Event definitions are imported from the JMS-enabled system through an Event Importer into the Model Repository. You can then access these Event Definitions from the Modeler, and add Data Definitions to reference these events. When the process, including the Event and Data definitions are deployed, the Business Events Manager uses the data to convert the incoming events into the format required. The Business Events Manager also makes sure that the correct information is added to the event, so it can be used by the Business Impact Engine.

If, for any reason, the Business Impact Engine cannot receive an event from the JMS Business Event Handler, the JMS Business Event Handler rolls back the event transaction in exactly the same way as the Business Events Handler.

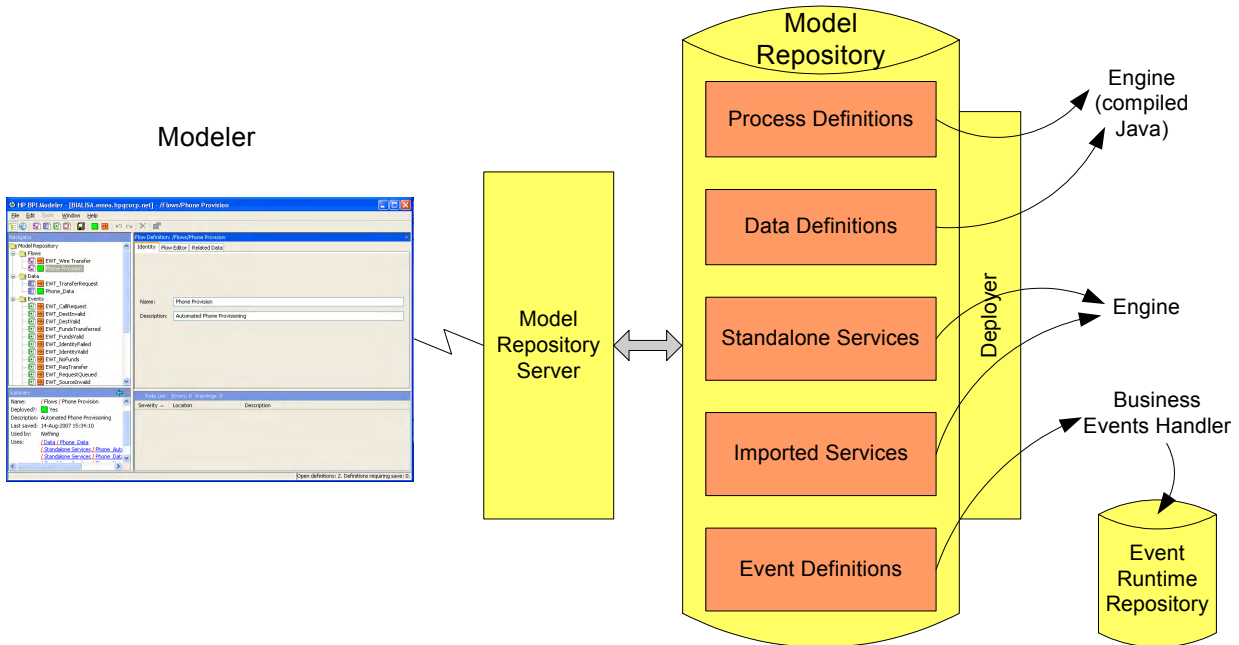
Modeler

The HPBPI Modeler is a client-side HPBPI component that you use to create your business Process 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 processes 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 Process definition.

A local repository cache, in the HPBPI Modeler, holds Process 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 45.

Figure 14 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:

- Process 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 Processes*.

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 *Business Process Insight 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 *Business Process Insight System Administration Guide*.
- Intervention Client, which is a Web-based interface that enables you to access, or intervene in, business process instances. Using the Intervention Client is described in the *Business Process Insight 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.

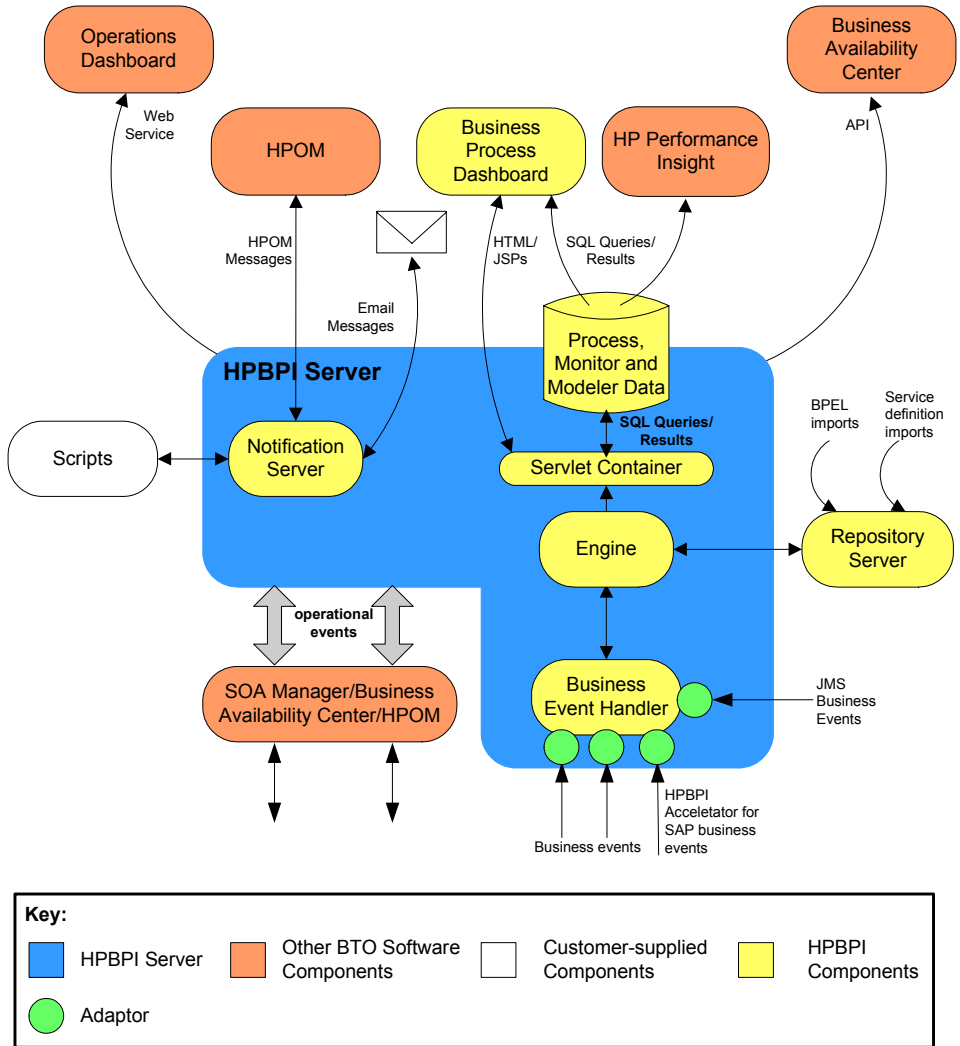
The chapter covers the following topics:

- [HPBPI Integration Points](#) on page 50
- [HPBPI Adapters for HP BTO Software Product's Integration](#) on page 54
- [HPBPI Accelerator for SAP Applications](#) on page 59
- [iWay Integration](#) on page 60
- [HP Operations Dashboard](#) on page 61
- [Self-Healing Services](#) on page 63

HPBPI Integration Points

Figure 15 shows a high-level diagram of how components can integrate with HPBPI; these include both design-time and run-time integrations.

Figure 15 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 processes and report on the status of these services and how the services are impacting your business processes. Section [HPBPI Adapters for HP BTO Software Product's Integration](#) on page 54 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 processes. You can also use these business events to report the business impact where processes are not progressing as they should be. The Business Event Handler is described in [Chapter 2, HPBPI Architecture](#).

3. Configure JMS data sources through the JMS Business Event Handler in order to import JMS-based Business Events. The JMS Business Event Handler is described in [Chapter 2, HPBPI Architecture](#).

4. Configuring notifications through the Notification Server:

- You can configure email notifications that can be sent when a notification event is received by the Notification Server.
- You can create scripts that are executed when a notification event is received by the Notification Server.

This enables you to integrate the data from 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 a 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 process. 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*.

5. Using a reporting tool such HP Performance Insight to report on data in the HPBPI Database.

You can use the data created by HPBPI in the database to report on your business processes and analyze the behavior of your processes as required. There is an HPBPI-specific report pack provided with Performance Insight.

6. Configure HPBPI to integrate with Business Availability Center. This enables Business Availability Center to report on HPBPI business processes, business process monitors and monitor thresholds from within the Business Availability Center Dashboard and My BAC portlets. The integration with Business Availability Center is described in detail in [Chapter 4, HPBPI and HP Business Availability Center](#).

7. Configure HPBPI to integrate with TransactionVision. This enables you to view the life cycle and associated data for specific business transactions within your organization. TransactionVision discovers transaction flows within your business, analyses the steps within these flows and the displays reports based on the results of the analysis. The integration with TransactionVision is described in detail in section [Chapter 5, HPBPI and TransactionVision](#).
8. 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 process definitions. Refer to the *Business Process Insight Integration Training Guide - Importing BPEL*, for more information on the options for importing BPEL definitions.

HPBPI Adapters for HP BTO Software Product's 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 54.
 - HP Service Oriented Architecture Manager (SOA Manager) service source; see section [HPBPI SOA Manager Service Source](#) on page 56.
- HP Operations Manager (HPOM) adapter; see section [HPBPI Operations Manager Adapter](#) on page 56.

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

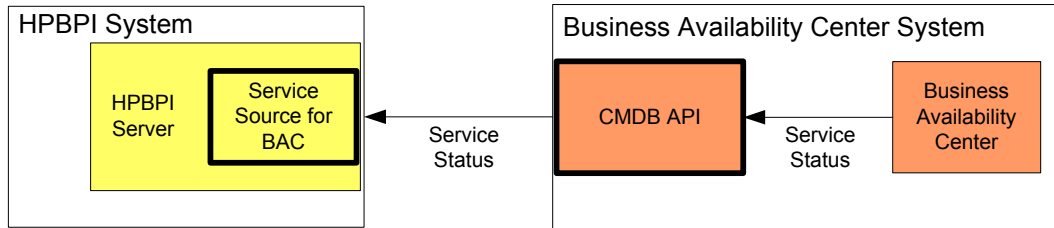
Service Source for Business Availability Center

Business Availability Center is one of the HP BTO Software products that you can integrate with. From within HPBPI, you create an Operational Service Source from the Business Availability Center server that you want to integrate with. You can then configure Business Availability Center to link Service status events to the business processes and process steps that you

define within HPBPI. You can link a number of HPBPI Servers to one Business Availability Center server; however, you cannot link one HPBPI Server to multiple Business Availability Center Servers.

You use Business Availability Center to model the Service definitions that you want to reference from within HPBPI, and the HPBPI Adapter for Business Availability Center (created as a result of defining an Operational Service Source) retrieves the status of these Services at intervals.

Figure 16 HPBPI Service Source for Business Availability Center



Many HP BTO software products store their service status data in the Business Availability Center configuration management database (UCMDB) as configuration items (CIs). This enables HPBPI access to all this service status data, without the need to model the data within the Modeler, or import the data into the Model Repository. You need only create a link to the required Service from the business process and the data is automatically obtained when the integration to Business Availability Center is configured.

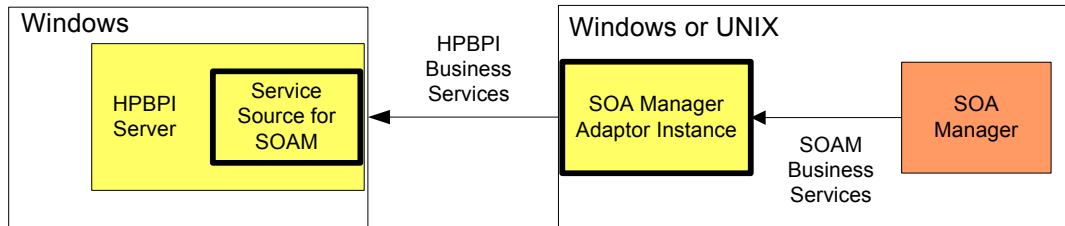
Details of you can integrate HPBPI and Business Availability Center can be found in [Chapter 4, HPBPI and HP Business Availability Center](#).

HPBPI SOA Manager Service Source

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 processes that you define.

HP SOA Manager enables you to manage 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 17 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 6, HPBPI and SOA Manager](#).

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.

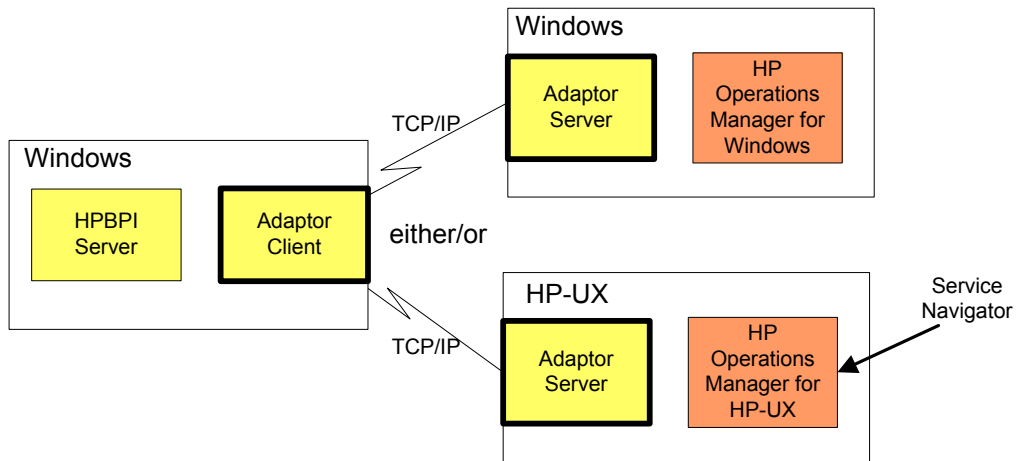


When HPBPI is configured to integrate with Business Availability Center, you can link HP Operations Manager services to HPBPI business process steps from within Business Availability Center. This avoids the need to configure the adapter and import the service definitions into the HPBPI Repository.

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 Process definitions as HPBPI events.

Figure 18 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 18](#) on page 57), 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.

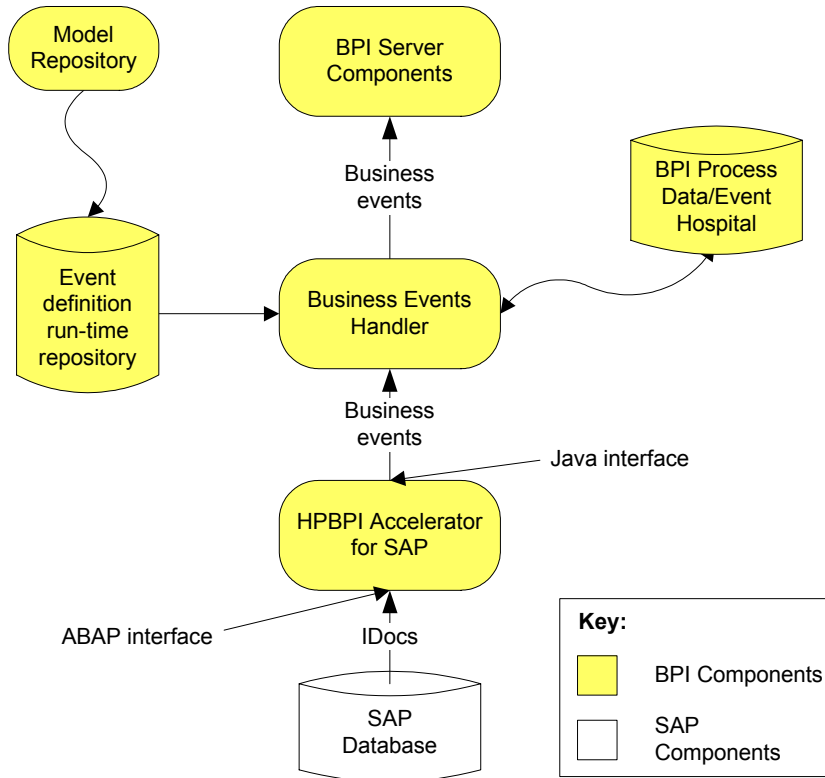
Details of how to install the HP Operations Manager Adapter for HPBPI are provided in the *Business Process Insight Installation Guide*.

HPBPI Accelerator for SAP Applications

In addition to the integrations with HP BTO Software products, HPBPI provides an Accelerator for SAP applications. The HPBPI Accelerator for SAP applications is specifically designed to integrate with 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 19 shows the integration between the HPBPI Accelerator and SAP.

Figure 19 HPBPI and SAP Integration Using HPBPI Accelerator



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

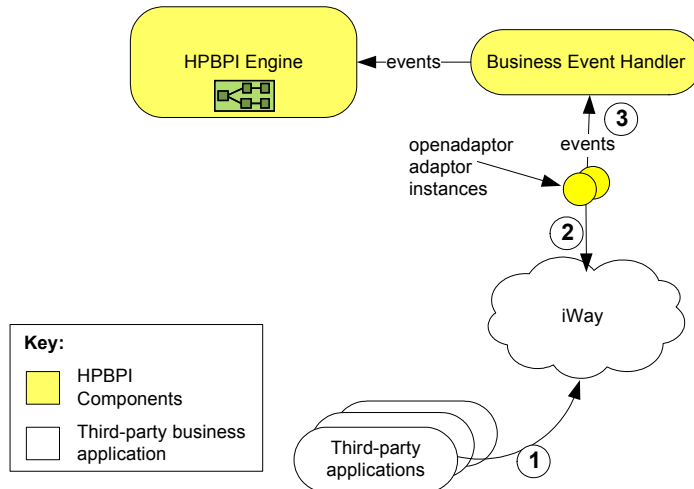
iWay Integration

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 20](#).

Figure 20 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 status 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 processes, monitors 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 monitors, 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 `OVBPPI`, 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, monitors 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.

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:

- [Overview of HPBPI and Business Availability Center Integration](#) on page 66
- [Configuring HPBPI to Integrate with Business Availability Center](#) on page 77
- [Security and Single Sign on](#) on page 84
- [Using SSL for Connections with Business Availability Center](#) on page 86
- [Business Availability Center System Configuration](#) on page 87
- [Creating a My BAC Application for HPBPI](#) on page 88
- [SLM and Reporting](#) on page 89
- [Description of Data Samples Sent to Business Availability Center](#) on page 90

Overview of 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.

You can configure a Service Source for the HPBPI and Business Availability Center integration, which enables you to map IT operational services to BPI Steps within Business Availability Center.

When you configure this integration between HPBPI and Business Availability Center, deployed business process definitions (including Steps) are automatically synchronized with Business Availability Center, where they are represented as CIs within the Business Processes View. There is an ongoing synchronization process between HPBPI and Business Availability Center following the initial export.



Superseded Process definitions are not exported.

When this integration is configured, 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 links through to appropriate Business Availability Center service definitions (in the Business Availability Center Dashboard) for impacted instances.

In addition to synchronizing Business Processes, you can additionally configure HPBPI to send data samples to Business Availability Center. These data samples provide data relating to the Business Process, Monitors and Steps defined for the Processes.



If you have defined Groups for your Monitors, the Groups are not exported to Business Availability Center. The data samples that HPBPI sends to Business Availability Center are for the entire Monitor definition.

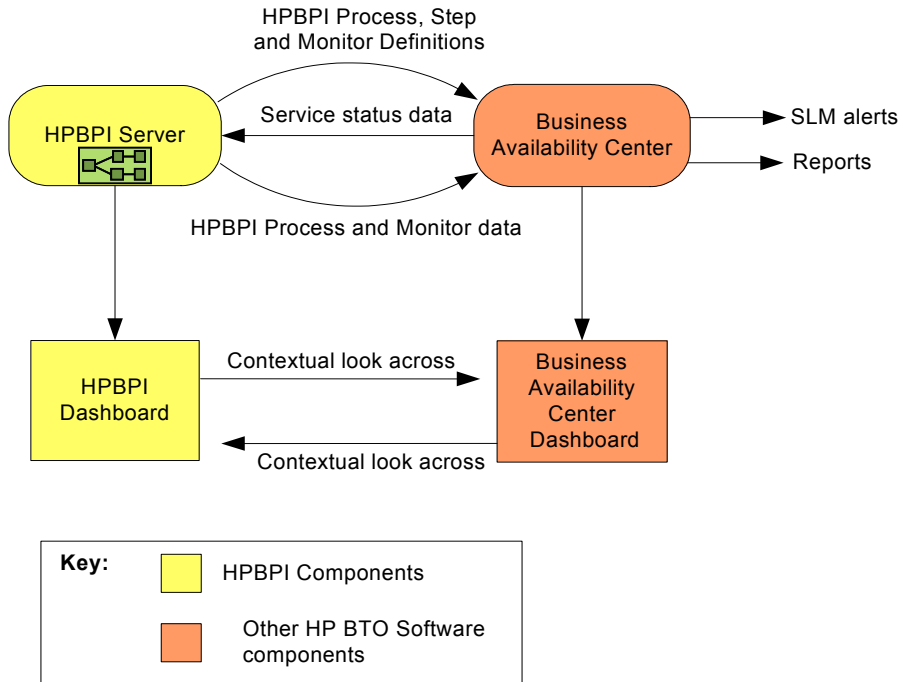
A data sample, in Business Availability Center terms, is application-specific data sent from an application, in this case, HPBPI. In the case of HPBPI, these data samples can be used by Business Availability Center to show key performance indicators (KPIs), relating to the imported Process and Monitor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

When the Business Monitors are available within Business Availability Center, you can then use the Business Availability Center Dashboard to monitor HPBPI Business Processes and Monitors.

The Business Availability Center Dashboard can also provide contextual links to the details of HPBPI-specific CIs; that is, Process, Steps and Monitor definitions, within the HPBPI Business Process Dashboard.

Figure 21 shows the high-level architecture diagram.

Figure 21 HPBPI and Business Availability Center High-Level Architecture



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

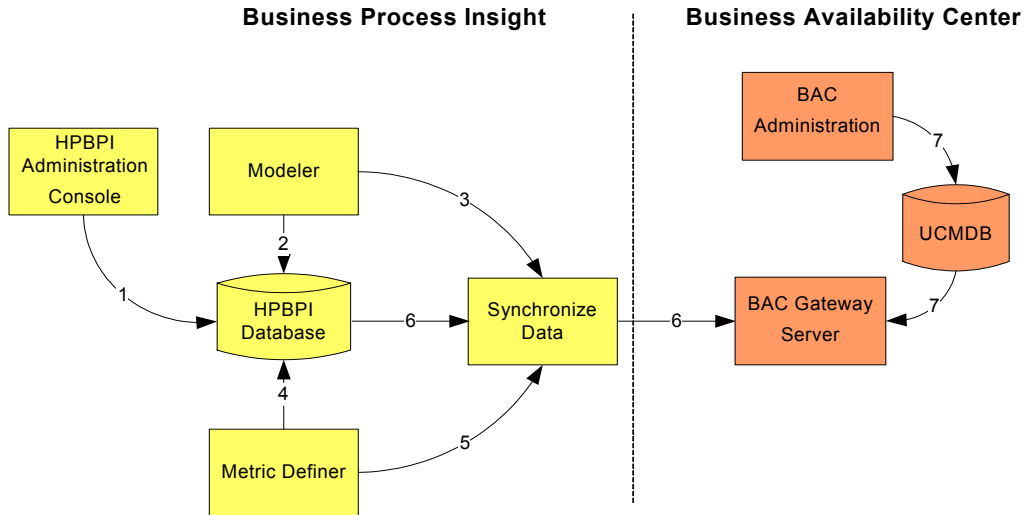
- use Business Availability Center to report on the status of operational services and how they impact HPBPI business processes.
- view HPBPI Process and Monitor definitions from within the Business Availability Center Dashboard.
- link from the Business Availability Center Dashboard to the HPBPI Business Process Dashboard to monitor more detail of the business impact for a selected business process.
- enable service level objectives (SLOs) to be set for the HPBPI configuration items (CIs), which are managed within Business Availability Center.

- configure alerts, within the Business Availability Center SLM component, based on the SLOs that have been set on the HPBPI CIs.
- enable reports to be generated with Business Availability Center based on the HPBPI CIs.
- use MY BAC portlets for views of the HPBPI Process and Monitor data.

Design-Time Integration

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

Figure 22 HPBPI and Business Availability Center Design-Time Configuration



The flow of configuration information, as shown in [Figure 22](#), is described in the following steps:

Step 1: Using the HPBPI Administration Console, configure a Gateway Server name and connection information for the Business Availability Center Server that you intend to integrate with.

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

Step 3: At the same time as the new Process definitions are created in the HPBPI database tables, the appropriate Business Process and Business Process Step CIs are automatically created in the UCMDB.

Step 4: Create Business Process Monitors definitions using the HPBPI Monitor Definer. These definitions are written to the HPBPI database tables.

Step 5: At the same time as each new Monitor definition is created in the HPBPI database tables, an BPI Monitor CI is created in the UCMDB.

Step 6: Whenever changes are made to any of the appropriate Business Availability Center settings in the Administration Console, all HPBPI Processes and Monitors are synchronized between the HPBPI database tables and the UCMDB.



Superseded Process definitions are not synchronized.

Ongoing changes made to the Processes and Monitors in the HPBPI database tables are automatically synchronized with the UCMDB.

Each time the HPBPI data is synchronized with the Business Availability Center data, the identifiers created for the HPBPI CIs in the UCMDB are returned to HPBPI and stored in the HPBPI database.

Step 7: You map the IT operation entities within the Business Availability Center environment to the BPI Processes and Monitors imported from HPBPI.

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

Run-Time Integration

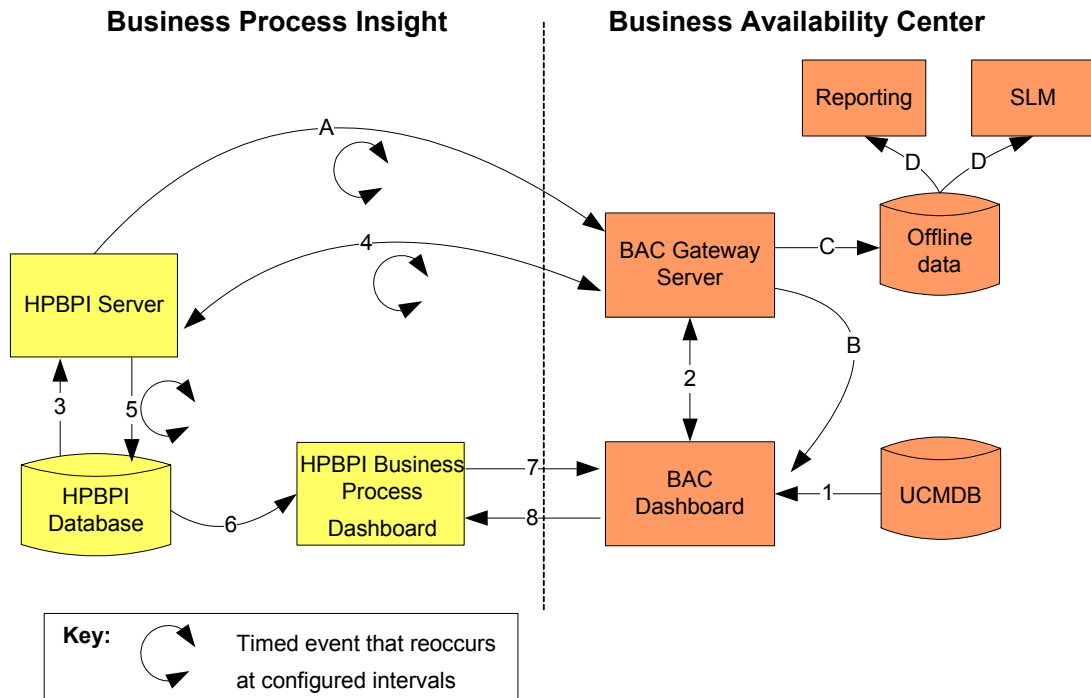
This section describes the behavior of the HPBPI and the Business Availability Center system following the initial design of the systems, as described in [Design-Time Integration](#) on page 69.

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 operational status of BPI Steps; see section [Request for Service Status from Business Availability Center](#) on page 72.
- HPBPI sends data samples to Business Availability Center; see section [HPBPI and Business Availability Center Data Sampling](#) on page 72.

These logical areas are shown in [Figure 23](#).

Figure 23 HPBPI and Business Availability Center Run-Time Behavior



Request for Service Status from Business Availability Center

The following steps describe the behavior of the HPBPI integration with Business Availability Center when determining the status of the HPBPI CIs. These steps occur only when you configure `Service Source` settings within the `HP Business Availability Center settings` options in the `Administration Console`.

The steps in the list relate to the numbers in [Figure 23](#):

Step 1: Business Availability Center determines the status of the KPIs configured for Business Process Insight.

Step 2: The current status is displayed in the Business Availability Center Dashboard and passed to the Business Availability Center Gateway Server.

Step 3: The 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 Gateway 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 Dashboard 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.

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.

These steps occur only when you configure Data Sample settings within the HP Business Availability Center settings options in the Administration Console.

The steps in the list relate to the alpha characters in [Figure 23](#).

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 Gateway Server. The data samples are held on the HPBPI Server for a configurable length of time, until the 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 used to evaluate 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 Business Availability Center offline database are made available for service level monitoring using the Service Level Management (SLM) component and Reporting.

Configuration Items Stored in the UCMDB

Table 1 describes the HPBPI data that is stored in the UCMDB as a result of the synchronization. This sharing of information between HPBPI and Business Availability Center enables Business Availability Center to access the HPBPI data from the UCMDB. The HPBPI data is stored in the UCMDB as configuration items (CIs).

Table 1 Configuration Items Stored in the UCMDB for HPBPI Process Components

CI Type	Description
business_process	Stores the Process Name and Process Description as entered using the HPBPI Modeler. A business_process CI is created for each Business Process in HPBPI. business_process CIs are logical CIs.
bpi_step	Stores the Step Name and Description as entered using the HPBPI Modeler. A bpi_step CI is created for each Step in all Business Processes. bpi_step CIs are logical CIs.
bpi_monitor	Stores the Monitor Name, Description, Type, Units, Collection Interval, Threshold Operator, Server Name and Instance Threshold limits (where specified) as defined with HPBPI. Monitor data samples are sent for each bpi_monitor CI. bpi_monitor CIs are Monitor CIs.

Table 1 Configuration Items Stored in the UCMDB for HPBPI Process Components

CI Type	Description
bpi_process_monitor	<p>A bpi_process_monitor CI is created in the UCMDB for every business_process CI stored. This CI stores the process monitor data collected implicitly by HPBPI for each process; for example:</p> <ul style="list-style-type: none"> • current numbers of active instances • numbers of blocked instances • number of completed instances <p>Data samples are sent to Business Availability Center for each bpi_process_monitor CI.</p> <p>bpi_process_monitor CIs are Monitor CIs.</p>
bpi_step_monitor	<p>A bpi_step_monitor is created in the UCMDB for every bpi_step CI stored. This CI stores the process monitor data collected implicitly by HPBPI for each process step; for example:</p> <ul style="list-style-type: none"> • current numbers of active instances • number of completed instances <p>Data samples are sent to Business Availability Center for each bpi_step_monitor.</p> <p>bpi_step_monitor CIs are Monitor CIs.</p>

As with any synchronization, or sharing of data, there must be a master copy of the data to synchronize from. In this case, HPBPI holds the master copy of the data. This means that any changes to the data within Business Availability Center UI Framework are overwritten by HPBPI when the data models are synchronized.

Data synchronization occurs as a result of any one of the following:

- a business process is deployed, re-deployed, or un-deployed.
- a business process monitor definition is created, modified or deleted.
- related configuration changes within the HPBPI Administration Console.

As part of the synchronization process, HPBPI creates mapping tables, which are used to provide the references between the HPBPI data and the corresponding CI within the UCMDB. The HPBPI unique identifiers for the data (GUIDs) are mapped to the UCMDB GUIDs for the HPBPI-related CIs. This enables HPBPI to send data samples to Business Availability Center as well as manage the synchronization process.

Business Availability Center and Load Balancing

You can configure Business Availability Center to utilize load balancing. If load balancing is configured for Business Availability Center, there are two Gateway Servers that can be configured for HPBPI:

- A Gateway Server for the HPBPI Data Collector.

This is the Gateway Server that receives data samples, is used for obtaining the status of services and is used for UCMDB synchronization. It is referenced as the `Hostname of Gateway server for Data Collectors` in the HPBPI Administration Console.

- A Gateway Server for users of Business Availability Center when accessing data, including HPBPI users.

This is the Gateway Server that is used from the Business Process Dashboard to access the Business Availability Center Dashboard and is referenced as the `Hostname of Gateway Server for Application Users` in the HPBPI Administration Console.

If there is no load balancing configured for Business Availability Center, you can use the same values for both Gateway Server parameters within the HPBPI Administration Console.

Configuring HPBPI to Integrate with Business Availability Center

You configure the integration with Business Availability Center through the HPBPI Administration Console.

There are a number of properties that you need to complete, according to the how you want to interoperate.

The Business Availability Center General settings properties need to be completed in all cases; these are described in section [Business Availability Center General Settings](#) on page 78.

If you want to send data samples to Business Availability Center in order that the HPBPI data can be displayed and managed using the Business Availability Center components, you need to configure Data Sample settings as described in section [Configuring a Business Availability Center Data Sample Destination](#) on page 80.

If you want 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 configure Service Source settings as described in section [Using Business Availability Center as a Source of Operational Service Status Data](#) on page 82.

Complete the sections appropriate to your requirements and then continue at section [Business Availability Center System Configuration](#) on page 87 to configure the Business Availability Center system.

Business Availability Center General Settings

You configure general settings that apply to both the Data Sample Destination settings and the Operational Service Source settings using the Administration Console as follows:

1. Start the Administration Console:

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

2. Click the HP Business Availability Center settings option in the left-hand navigation pane.

A configuration page is opened in the right-hand pane, which contains all the configuration options for the Business Availability Center integration. You need to set the parameters for the Business Availability Center General settings.

3. Enter values for the properties of the General settings. The properties are fully described in the *Business Process Insight Administration Guide*:

- Hostname of Gateway server for Data Collectors

The fully qualified DNS name of the machine where the Business Availability Center Gateway Server is installed. This is the server that you want HPBPI to use as a source for Service Status information, as the destination for data samples and for UCMDB synchronization.

- HTTP port of Gateway server for Data Collectors

The port number used by the Web Server on the Business Availability Center Gateway Server for Data Collectors. Usually this is port 80, or port 443 if you have configured SSL.

- Use SSL for Data Collectors?

Enables or disables use of SSL when connecting to the Gateway Server for Data Collectors; see also section [Using SSL for Connections with Business Availability Center](#) on page 86.

- Hostname of Gateway Server for Application Users

The fully qualified DNS name of the machine where the Business Availability Center Gateway Server is installed. This is the Gateway server that provides access to the Business Availability Center

Dashboard for Web Browser users, such as users of the Business Process Dashboard. Can be the same Gateway Server name as that defined for Data Collectors.

- HTTP port of Gateway server for Application Users

The port number used by the Web Server on the Business Availability Center Gateway Server for Application Users.

- Use SSL for Application Users?

Enables or disables use of SSL when connecting to the Gateway Server for Application Users; see also section [Using SSL for Connections with Business Availability Center](#) on page 86.

- User name

The user name that you have defined to be used by HPBPI server components when connecting to Business Availability Center. This is the account used by HPBPI to obtain the Service Status information from Business Availability Center and for synchronization.

This can be an existing user, or a new user, as long as it has permissions to access the View name as specified in the Business Availability Center settings in the HPBPI Administration Console. For a new installation the view name is Business Processes.

- Password

The password set for User name.

- Maximum retries for synchronization deadlock

For a new configuration accept the value offered.

- Hostname of HPBPI Dashboard

The fully-qualified hostname of the machine where the HPBPI Business Process Dashboard is installed. This is the Dashboard that Business Availability Center users are directed to when they select the go to BPI menu option.

- HTTP port number of HPBPI Dashboard

Port number of the machine identified under Hostname for HPBPI Dashboard.

You do not need to restart any HPBPI components for these settings to take effect. HPBPI automatically synchronizes its data with Business Availability Center when changes are applied to the General settings.

You can now continue and complete your configuration for one or both of the following:

- Data samples settings; see section [Configuring a Business Availability Center Data Sample Destination](#) on page 80
- Service source settings; see section [Using Business Availability Center as a Source of Operational Service Status Data](#) on page 82.

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. Sending data samples to Business Availability Center enables you to view HPBPI Business Process Monitor data from within the Business Availability Center Dashboard.

Business Availability Center displays the sample information based on the time period specified in the `Data sample send interval` and until the next sample is sent by HPBPI. In order to make sure that there are data samples to display, there is an additional decay period that you can set within Business Availability Center. Unless modified, the value of the decay period is set at 90 seconds. The decay period allows for cases where samples are not sent, due to delays or out-of-sequence events. If no decay period is set, or is no new sample is received within the decay period, then Business Availability Center displays a status of `No Data`.

The Data Sample settings are configured using the Administration Console as follows:

1. Select `HP Business Availability Center settings` from the Navigation tree in the Administration Console.

The right-hand pane shows the configuration options for the Business Availability Center integration.

2. Select the `Data Samples enabled` check box to enable the Data Samples. If you do not do this, you cannot modify the parameter values for any of the settings.

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

- Data samples send interval (minutes)

For a new configuration accept the value offered. This interval becomes the collection period for all HPBPI monitors for as long as you enable sending data samples to Business Availability Center.

- Connection timeout (seconds)

For a new configuration accept the value offered.

- Guaranteed retention period (minutes)

For a new configuration accept the value offered.

- Minimum Send Delay for Monitor Samples (seconds)

Delay following the Data samples send interval to allow time for the Monitor Engine to collate data to be sent to Business Availability Center.

4. Click the Apply button to apply your changes.

You do not need to restart any HPBPI components for these settings to take effect. HPBPI automatically synchronizes its data with Business Availability Center when changes are applied to the Data Sample settings.

When you have configured your Data Sample settings, you can continue at section [Using Business Availability Center as a Source of Operational Service Status Data](#) on page 82, if you want to receive status information for operational services from Business Availability Center.

If you do not want to create a service source, continue at section [Business Availability Center System Configuration](#) on page 87 to configure the Business Availability Center system.

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

If you want HPBPI to receive status information relating to operational services modeled within Business Availability Center, you need to configure an Operational Service Source for Business Availability Center; these are operational services that have been configured for BPI Step CIs within Business Availability Center.

You configure the Service Source settings using the Administration Console as follows:

1. Start the Administration Console as follows:

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

2. Click the HP Business Availability Center settings option in the left-hand navigation pane.

A configuration page is opened in the right-hand pane, which contains all the configuration options for the Business Availability Center integration.

3. Select the Service Source check box to enable the service source. If you do not do this, you cannot modify the parameter values for the settings.
4. Enter values for the properties of the Service Source settings. The properties are fully described in the Business Process Insight Administration Guide:

- View name

Unless you have modified this on the Business Availability Center system, accept the value offered (Business Process)

- Status event poll interval (seconds)

For a new configuration accept the value offered.

- Connection timeout (seconds)

For a new configuration accept the value offered.

- Override Service Adapters' log level?
For a new configuration accept the value offered.
- Log level for this adapter
For a new configuration accept the value offered.

5. Click the Apply button to apply your changes.

You do not need to restart any HPBPI components for these settings to take effect. HPBPI automatically synchronizes its data with Business Availability Center when changes are applied to the Service Source.

You should now continue at section [Business Availability Center System Configuration](#) on page 87 to configure the Business Availability Center system.

Security and Single Sign on

You can configure your HPBPI and Business Availability Center integration such that only one set of login credentials is required to navigate between the HPBPI and the Business Availability Center Dashboards.

Within HPBPI, the following are the different methods that you can use to secure the Business Process Dashboard:

- Select Access
- Servlet Engine
- open access with no security. In this case, no additional configuration is required and the Business Availability Center logon credentials are provided when required.

Select Access Authorization

You can enable Select Access authorization for HPBPI using the Administration Console. You also need to configure Select Access, using the configuration tools for Select Access, to add HPBPI and Business Availability Center. The *Business Process Insight Administration Guide* describes how to configure HPBPI to use Select Access. If you similarly enable Business Availability Center to be authorized using Select Access, you need only logon once to navigate between the Business Process Dashboard and Business Availability Center.

Servlet Engine Authorization

To use the Servlet Engine security, you need to configure the Business Process Dashboard to be authorized using the Servlet Engine. If you enable Servlet Engine authorization, then when you log on to the Business Process Dashboard, HPBPI creates the cookie. If you then attempt to access Business Availability Center pages from within the Business Process Dashboard, HPBPI passes the cookie to Business Availability Center, which in turn validates it.

In the same way, if you log on to the Business Availability Center Dashboard, a cookie is created. If you then attempt to access the Business Process Dashboard from within Business Availability Center, the cookie is passed by the browser to the BPI servlet engine, which validates it.



In order that the Servlet Engine authorization can be enabled:

- Both HPBPI and Business Availability Center must be in the same domain.
- the user name specified for the Dashboard in the file `tomcat-users.xml` needs to be the same as the name specified for the Business Availability Center user within Business Availability Center administration.

This is also the User name specified in the Business Availability Center settings in the HPBPI Administration Console. The passwords do not need to match, only the user names.

Refer to the *Business Process Insight Administration Guide* for details of how to configure Servlet Engine authorization for the Dashboard and modify the file `tomcat-users.xml`.

Using SSL for Connections with Business Availability Center

You can specify an SSL (secure socket layer) connection between HPBPI and Business Availability Center.

In order to configure Business Availability Center to support HTTPS, you need to obtain an SSL certificate. SSL uses public key cryptography, which is based on key pairs. These key pairs contain one public key and one private key. These keys are used by Business Availability Center to enable HTTPS. When installed the certificate encrypts the data that is sent between HPBPI Web interfaces (clients) and the Business Availability Center Web Server.

In order that HPBPI can use SSL when integrating with Business Availability Center, you need to import the required SSL certificate from Business Availability Center. You import the certificate into the keystore for your Java distribution on the HPBPI Server machine.

The instructions for importing the certificate are provided in the Business Availability Center documentation in a document called: *Hardening Guide* within the `System Deployment` category of documents.

Business Availability Center System Configuration

There is no specific configuration required for the Business Availability Center system. HPBPI passes the necessary security information to Business Availability Center when it sends data to be stored in the UCMDB.

You can create My BAC applications for HPBPI, from within Business Availability Center, as described in section [Creating a My BAC Application for HPBPI](#) on page 88.

Creating a My BAC Application 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 Process information.

There is a Category, within the Personalize option, for My BAC called HP Business Process Insight. The following are the portlets that you can add for the BPI Integration category to display HPBPI data:

- a BPI Process Diagram portlet to display an HPBPI Process diagram that corresponds to a selected CI.
- a BPI Monitors Over Time portlet to display:
 - the average, minimum, maximum, process value or duration of a CI over a specific time period.
 - the violated instances of a CI over a specific time period.
- a Business Process Over Time portlet to display information about the volume, backlog and health of a selected Business Process CI over a specific time interval.
- a Business Process Summary portlet to display the summary data for a selected business process.

For more details of the My BAC portlets available for HPBPI, refer to the Business Availability Center documentation.

You cannot link directly from My BAC to the HPBPI Business Process Dashboard. This is because My BAC needs to be able to call the Business Process Dashboard as a portlet and as the Business Process Dashboard is not a portlet, My BAC cannot access it.

SLM and Reporting

As mentioned earlier in the chapter, the data samples sent to Business Availability Center are written to an offline data repository.

The Business Availability Center Service Level Management (SLM) component uses the data stored in the offline data repository as a source for building Service Level Agreements and setting Service Level Objectives, and for creating reports.

In terms of Service Level Management, you can:

- enable service level objectives (SLOs) to be set for the HPBPI configuration items (CIs), which are managed within Business Availability Center.
- configure alerts, within the Business Availability Center SLM component, based on the SLOs that have been set on the HPBPI CIs.

Refer to the Business Availability Center Service Level Management documentation for information about setting Service Level Objectives.

In terms of reporting, you can configure the Business Availability Center Report Administration component to generate reports based on the HPBPI CIs.

Refer to the Business Availability Center documentation related to custom reporting and alerting for more information.

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 Monitor statistics.

- throughput

The data sample for this KPI is taken from the throughput value as calculated as part of the Business Monitor statistics.

- duration

The data sample for this KPI is taken from the duration value.

This KPI is displayed only if the Business Process includes a Business Monitor where the Monitor 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 number of *blocked*, *at risk* and *healthy* instances reported in the Business Process Insight data samples. This KPI applies to processes only.

These data samples include as much information as is possible to give the KPI context when displayed within the Business Availability Center Dashboard.

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



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

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

If any of the backlog, throughput and duration thresholds in the Business Process Monitor 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.

5 HPBPI and TransactionVision

This chapter describes how you can integrate HPBPI and TransactionVision.

TransactionVision enables you to view the life cycle and associated data for specific business transactions within your organization. TransactionVision discovers transaction flows within your business, analyses the steps within these flows and then displays reports based on the results of the analysis. Using TransactionVision, you can better monitor and manage these business transactions.

This chapter covers the following topics:

- [Overview of TransactionVision Integration](#) on page 94
- [Configuring HPBPI to Integrate with TransactionVision](#) on page 100
- [Example Configuration for the TransactionVision System](#) on page 117

Overview of TransactionVision Integration

TransactionVision comprises a set of sensors. The sensors collect transaction events by monitoring the behavior of your organization's business applications. The sensors report these events to the TransactionVision analyzer, which correlates the events and further processes them. Specifically, the analyzer is responsible for assembling the individual transaction events into business transactions. It can then process and action these business transactions according to rules that you define. You can configure more than one analyzer for TransactionVision. Multiple analyzers are used for load balancing and all communicate with the same database. It therefore does not matter which analyzer you use to manage your data, as the data source is common to them all.

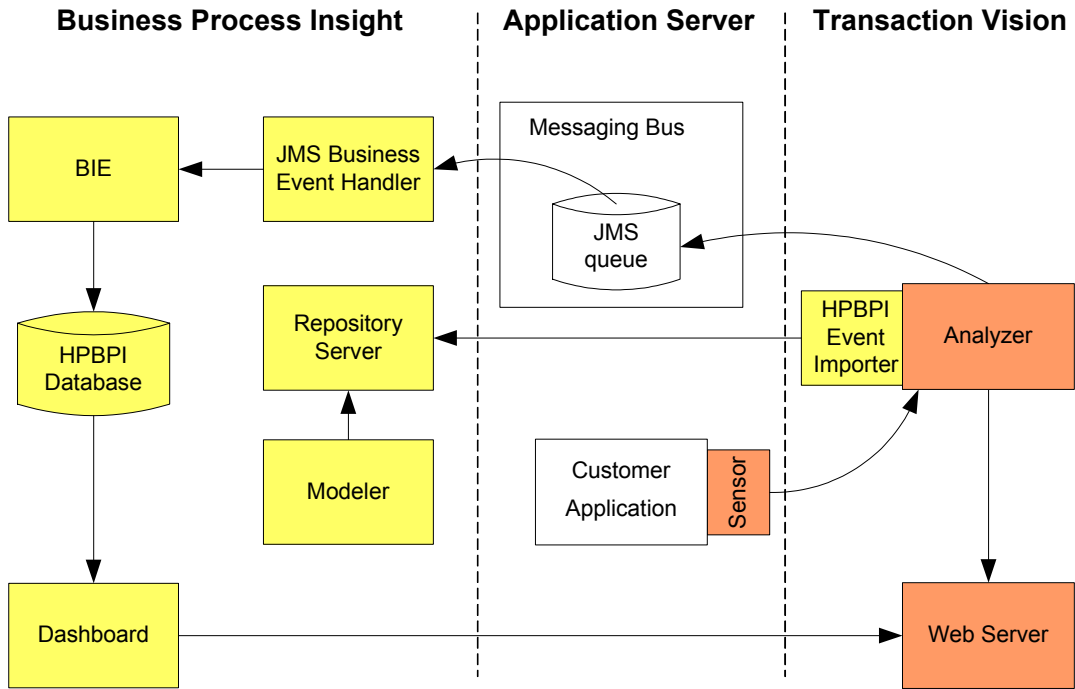
You can define rules within TransactionVision that are specific to HPBPI. These rules define the transaction events that you want to propagate to HPBPI. TransactionVision then collates these events and aggregates the data. Business, including the aggregated data are then placed on a JMS queue to be read by the JMS Business Event Handler.

In addition, you can use the TransactionVision Web User Interface to provide you with reports, based on the business transactions that TransactionVision has recorded.

The components of TransactionVision communicate using an application server, specifically, the message-oriented middleware component of the application server; this is referred to as a messaging bus in this chapter. You need a component of the same messaging bus on the machine where HPBPI is installed in order that HPBPI can communicate with TransactionVision. Typically, the messaging bus components are referred to as client and server components.

Figure 24 shows the high level architecture of the HPBPI and TransactionVision integration.

Figure 24 HPBPI and TransactionVision Integration Architecture



The integration between HPBPI and TransactionVision enables you to:

1. Import, into the HPBPI Model Repository, event definitions that have been defined within TransactionVision.
You can then use the Modeler to create processes, which have Data definitions that subscribe to these imported events.
2. Receive events from TransactionVision and use the data within them to progress HPBPI business processes.
3. Link from the process instance details page within the HPBPI Business Process Dashboard to the details of the instances within TransactionVision.

This provides HPBPI users with more information relating to problems that might be occurring in a process at the transaction level.

Design-Time Integration

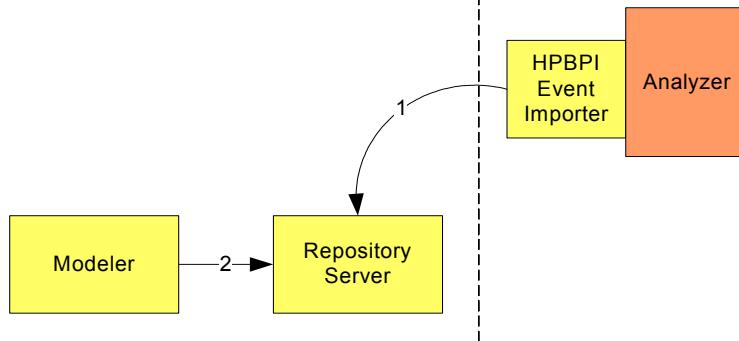
This section describes how data is moved between HPBPI and TransactionVision when you are designing your processes and configuring the HPBPI and TransactionVision systems.

[Figure 25](#) shows the flow of information from TransactionVision to HPBPI. It assumes that:

- TransactionVision Sensors are already configured to collect the required transaction information.
- You have created a project within TransactionVision for the transactions that you want to collect.
- You have specified how you want to store the data that is extracted from the transactions that TransactionVision is collecting in its database; this is the structure of a database table.
- You have classified rules for how the transaction data should be saved in the database; that is, which data is stored in which database column.

Figure 25 TransactionVision Design-Time Integration

Business Process Insight **TransactionVision**



The flow of data, as shown in [Figure 25](#), is described in the following steps:

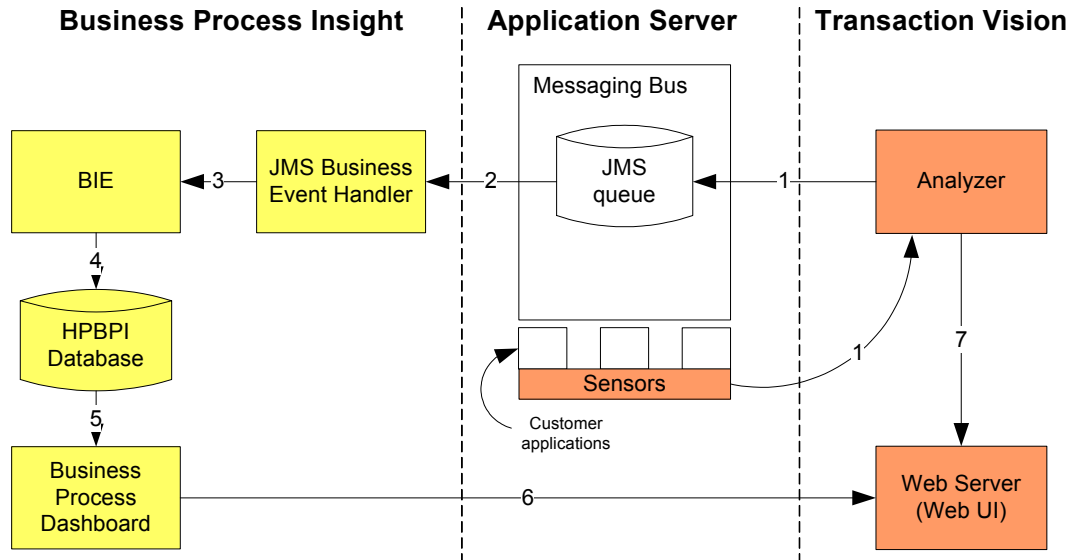
Step 1. Event and transaction rules have been defined to filter the incoming transactions, within TransactionVision and according to your requirements. A BPI Action has been defined that creates a Business Event from the transaction event data, The HPBPI Event Importer imports the required Business Event definitions into the Model Repository, where they are available to the Modeler.

Step 2: You define your processes as usual and create Data definitions, which can then subscribe to the Business Event definitions imported from TransactionVision.

Run-Time Integration

This section describes how data is processed between HPBPI and TransactionVision when your system is operational.

Figure 26 TransactionVision Run-Time Integration



The flow of data, as shown in [Figure 26](#), is described in the following steps:

Step 1: Using the rules that you have defined for the transaction events, TransactionVision filters the appropriate events and completes the BPI Action, which in turn places Business Events on the JMS queue that you have defined.

Step 2: The JMS Business Event Handler is listening for new events that are arriving on the queue.

Step 3: The JMS Business Event Handler accepts the business events and sends them to the Business Impact Engine for processing.

Step 4: Details of the process and its progression are held in the HPBPI database as usual.

Step 5: The Business Process Dashboard presents the results of the process progression.

Step 6: The Business Process Dashboard provides links to the appropriate transaction events within the TransactionVision Web user interface.

Step 7: Details of the TransactionVision events are displayed within the TransactionVision Web user interface.

Configuring HPBPI to Integrate with TransactionVision

This section describes the tasks that you must complete in order that the HPBPI and TransactionVision integration can be enabled.

The configuration tasks described in these sections assume that the necessary HPBPI components have been installed on the TransactionVision and the HPBPI systems.

The required TransactionVision components are installed automatically, on the HPBPI system, so no additional installation is required; however, you need to install the HPBPI-required components (Event Importer) on the appropriate TransactionVision machine. The installer and instructions for installation can be found on your distribution media in the following directory:

```
cd-drive\common\bpiTvIntegration
```

TransactionVision Configuration Tasks

The following tasks need to be completed on the TransactionVision server where a TransactionVision analyzer is installed. All analyzers can communicate with each other and all communicate with the same database. It therefore does not matter which analyzer you use to manage your data, as the data source is common to them all:

- Make sure that you have added the required TransactionVision Sensors to the applications you want to monitor; see the *TransactionVision Administration Guide*.
- Create a TransactionVision Project; see the *TransactionVision Administration Guide*.
- Define how to store the data extracted from the application within TransactionVision database; see the *TransactionVision Advanced Customization Guide*.
- Classify the rules that determine how and where to save the data extracted from the transaction events for HPBPI data in the TransactionVision database table, or tables; see section [Classify Events](#) on page 102.

- Create a JMS queue for HPBPI that is accessible from both TransactionVision and HPBPI; see the documentation for your messaging bus.
- Configure the JMS queue and JNDI repository for TransactionVision. The settings for the JMS queue need to be consistent across HPBPI and TransactionVision; see section [Configure a JMS Queue and JNDI Repository On the TransactionVision System](#) on page 101.

This document does not cover the full details of completing these tasks, it provides an overview of the key tasks; you need to refer to the TransactionVision documentation for details of using the TransactionVision GUI to complete the required tasks. The TransactionVision integration with HPBPI is described in the *TransactionVision Administration Guide*.

There are also some examples, at the end of this chapter, which provide information related to the configuration information required on the TransactionVision system. The examples are based on a WebSphere MQ series messaging bus.

Configure a JMS Queue and JNDI Repository On the TransactionVision System

When you have created a JMS queue, you use the TransactionVision GUI to configure the details of this queue for the TransactionVision and HPBPI integration.

You need to configure a JNDI repository, on the TransactionVision system, and add the JNDI entries for accessing the newly configured JMS queue, which has been created on the messaging bus for the integration. Refer to the documentation for your JMS messaging bus for the details that you need for the JNDI repository configuration.

Using the TransactionVision interface, navigate to the page for the Business Process Insight integration option. You can access this page from the Integration option under the Administration menu. From the Business Process Insight page, enter the JNDI details that reference the JMS queue for the integration.

Classify Events

You need to classify the TransactionVision events that you want to have delivered to HPBPI. You classify TransactionVision events to label the classes and define the rules for extracting the required data from the events.



You need to make sure that you specify appropriate filters to make sure that you are not monitoring events that are generated by TransactionVision.

TransactionVision provides a structured editor to enable you to manage transaction rules and associated attributes. You need to define rules which identify the transaction events that are sent to HPBPI. You use TransactionVision's Transaction Definition Editor to classify the events within the TransactionVision Web-based GUI.

Using the Transaction Definition Editor, you see an overview of all the transaction rules that are defined on your TransactionVision Server system. Using these rules, you can define:

- which events belong to which transaction class, or classes.
- how data is mapped between transaction events and business transactions.
- actions, based on criteria that you define.

You access the Transaction Definition Editor from the Administration menu within the TransactionVision GUI. Refer to the *TransactionVision User Guide* in the section covering Classifying Transactions for full details of using the editor.

The following is an overview of the tasks that you need to complete:

1. From within the Transaction Definition Editor, select the Common Section for the event that you want to send to HPBPI.
2. Define the data that you want included in the event; you do this by defining Attribute Rules for each item of data .

3. Configure TransactionVision to generate a Business Event for HPBPI in a format and that can be imported into HPBPI. To do this, you need to specify:

- the Action Java Bean provided by TransactionVision.

This is a TransactionVision action, which is specific to HPBPI, and defines a standard action for communicating events from TransactionVision to HPBPI.

To specify the Java Bean, you need to create an HPBPI Action. Use the option `New BPI Action` to create the Action for HPBPI.

The Java Bean specification is as follows:

```
com.bristol.tvision.services.analysis.actions.OVBPIEventPublisher
```

- the HPBPI business event name and event group that you have defined within the HPBPI Modeler for the TransactionVision process. This is the event name that you want to reference from within TransactionVision.

The configured events are then placed on the JMS event queue ready to be accessed by the HPBPI JMS Business Event Handler.



If you want to link different transaction events from TransactionVision into an HPBPI business process instance, there needs to be a common identifier in all TransactionVision events for HPBPI.

You must determine this unique identifier from a data item in the event payload, which is mapped to a transaction attribute. It is important to note that the business transaction identifier generated by TransactionVision cannot be used as a unique identifier. This is because TransactionVision business transactions identifiers can change as the transaction is processed.

HPBPI Configuration Tasks

The following tasks need to be completed on the HPBPI system; most of these tasks are described in the sections that follow:

- Make sure that you have installed the remote component of the HPBPI TransactionVision Event Importer on the system where the TransactionVision analyzer is installed; these components are located on your distribution media as follows:

```
cd-drive\common\bpiTvIntegration
```
- If required, install the remote components of the application server's messaging bus, as used by TransactionVision, on the HPBPI server. You need to install the components that enable HPBPI to access the configured JMS queue. Refer to your messaging bus manuals for details of what is required.
- Run the HPBPI Event Importer on the TransactionVision Server to create the required event definitions in the Model Repository; see section [Import TransactionVision Events to Model Repository](#) on page 105.
- Using the HPBPI Modeler, create Data definitions that subscribe to the imported TransactionVision events; refer to the *Integration Training Guide - Modeling Processes* and the Modeler Online Help for more information.
- Using the HPBPI Modeler define progression rules, based on the data being imported from TransactionVision events; refer to the *Integration Training Guide - Modeling Processes* and the Modeler Online Help for more information.
- Deploy the HPBPI process that you have created for your TransactionVision integration from within the HPBPI Modeler.
- Configure HPBPI to receive Business Events from the JMS queue that you have created for HPBPI; see section [Configure a JMS Business Event Handler](#) on page 113.
- As the process progresses, monitor it using the HPBPI Business Process Dashboard. You can also link from the Business Process Dashboard to the TransactionVision Web user interface for more details of the transactions related to the process instances. You do this from the individual instance page, where there is a link in the Associated Data table under the heading View Transaction Tracking Details.

Import TransactionVision Events to Model Repository

You need to import the Business Events, defined in TransactionVision as BPI Actions, into HPBPI. This is in order that they are available within your process definitions. When you have imported them, you can then subscribe to the events from the Data definitions that you have created for the Business Process.

The TransactionVision Event Importer for HPBPI must be run on the machine where the TransactionVision Analyzer component is installed and running.

The Event Importer can be set up to run periodically as a background task to ensure that HPBPI has the most up to date information relating to the transaction events. You can also run the Event Importer manually; this might be required if you make significant changes to the TransactionVision configuration, which then need to be synchronized with HPBPI immediately.

You can run the TransactionVision Event Importer in the following ways:

- to connect directly to both HPBPI and TransactionVision, where the importer synchronizes the event definitions as they are imported.

This is the most likely method of using the Event Importer.

- to first extract the transaction event information from TransactionVision and write the results to an intermediate XML file. You then need to copy the intermediate file to the machine where HPBPI is installed and run the Event Importer again to take the information from the intermediate XML file and synchronize the event definitions with HPBPI.

You are most likely to use this method when testing and setting up your integration, or if there are problems with the network link between HPBPI and TransactionVision.

Running the Event Importer

You cannot run the Event Importer on the TransactionVision machine unless you have already installed the required HPBPI components on the TransactionVision machine.



The version of the Event Importer must be the same as the version of the HPBPI Server (Model Repository) that you are using.

Complete the following steps to run the Event Importer:

1. Open a Command Prompt window on the TransactionVision machine.
2. Make sure the `TVISION_HOME` environment variable is set for this command prompt window session.

`TVISION_HOME` should be set to the value of the directory where TransactionVision is installed, for example:

```
C:\Program Files\HP\TransactionVision
```

3. Use the following command to run the Event Importer from the location where you unpacked it on the TransactionVision machine:

```
EvImporter-install-dir\bin\bpitveventimporter tv-info  
bpi-info options
```

where:

- *tv-info* identifies the TransactionVision server where the transaction event details are sourced, and where the Business Process Dashboard obtains the details of the events for display and monitoring purposes.

If you do not specify any *tv-info* settings, the event details are obtained from the machine where the command is run. In addition the Business Process Dashboard uses the local machine and port number 9080 to link to the details of the events on the TransactionVision system.

Refer to [Table 2](#) on page 107 for details of the possible settings for *tv-info*.

- *bpi-info* identifies the HPBPI server where the Process definition that requires the TransactionVision event details is defined. This is the machine where you want to synchronize the event definitions.

If you do not specify any *bpi-info* settings, the Event Importer connects to the HPBPI Server on the local machine.

Refer to [Table 3](#) on page 109 for details of the possible settings for *bpi-info*.

- *options* identifies optional parameters that you can include with the command.

Refer to [Table 4](#) on page 110 for details of the possible settings for *options*.

Table 2 describes the settings that you can use for *tv-info* within the `bpitveventimporter` command.

Table 2 TransactionVision Settings for Event Importer

tv-info Settings	Description
<code>-tvHost <i>hostname</i></code>	<p><i>hostname</i> is the full domain name of the TransactionVision Server (where both the Analyzer and Web UI are located). Use this setting to define a source machine for event definitions and for the Business Process Dashboard to use for monitoring purposes.</p> <p>If omitted, the local machine name is assumed.</p>
<code>-tvWebHost <i>hostname</i></code>	<p><i>hostname</i> is the full domain name of the TransactionVision Server where the Web UI is located.</p> <p>Required only if the Web UI is located on a different machine to the TransactionVision Analyzer.</p> <p>If omitted, <i>hostname</i> for <code>tvHost</code> is used. If there is no <code>tvHost</code> setting specified, the local machine name is assumed.</p>

Table 2 TransactionVision Settings for Event Importer

tv-info Settings	Description
<code>-tvWebPort</code> <i>number</i>	<p><i>number</i> is the port number of the TransactionVision Server where the Web UI is located.</p> <p>Required only if the Web UI is using a port number other than 9080.</p> <p>If omitted, a value of 9080 is assumed for the port number.</p>
<code>-tvInfoFromXML</code> <i>path</i>	<p>Optional setting for the importer to read the transaction event details from an intermediate XML file. This is an XML file that has previously been created using the <code>-tvInfoToXML</code> setting.</p> <p><i>path</i> is the location of the XML file.</p> <p>If specified, the values of the settings for <code>-tvHost</code>, <code>-tvWebHost</code> and <code>-tvWebPort</code> override the equivalent values specified in this XML file.</p>
<code>-tvInfoToXML</code> <i>path</i>	<p>Optional setting for the importer to save the details of the transaction events to an XML file instead of automatically synchronizing with HPBPI.</p> <p><i>path</i> is the name and location of the file where the details are to be written.</p>

The Event Importer does not allow the event definitions from one TransactionVision Analyzer to overwrite those from another. This means you need to choose event group names carefully when defining events in the TransactionVision classification files. If a naming clash is found, the importer reports the problem and does not proceed with the import. If you decide that you do want to overwrite the event definitions, use the `-rehost` option described in [Table 4](#). You might need to use this option if you move from using one TransactionVision Analyzer to another, perhaps for testing purposes.

Table 3 describes the settings that you can use for `bpi-info` within the `bpitveventimporter` command.

Table 3 HPBPI Settings for Event Importer

bpi-info Settings	Description
<code>bpiHost hostname</code>	<p><i>hostname</i> is the full domain name of the machine where the HPBPI server, to which you want to import event definitions, is installed.</p> <p>Use this setting in conjunction with <code>-bpiUser</code> and <code>-bpiPassword</code>.</p>
<code>-bpiPort number</code>	<p><i>number</i> is the port number for the RMI Registry on the HPBPI server. Unless you have modified it, the RMI Registry port number is 44000.</p> <p>If this setting is omitted, a value of 44000 is assumed.</p>
<code>-bpiUser username</code>	<p><i>username</i> is the user name that is used when you login to the HPBPI Modeler (and also used for the Model Repository Server).</p> <p>Use this setting in conjunction with <code>-bpiHost</code> and <code>-bpiPassword</code>.</p> <p>If this setting is omitted, a value of <code>admin</code> is assumed.</p>
<code>-bpiPassword password</code>	<p><i>password</i> is the password that is used when you login to the HPBPI Modeler.</p> <p>Use this setting in conjunction with <code>-bpiHost</code> and <code>-bpiUser</code>.</p> <p>If you omit this option you are prompted for the password. Be aware that the password is echoed to the screen.</p>

If none of the settings in Table 3 is specified, the Event Importer assumes that HPBPI is installed on the local machine and determines the settings that it requires from the HPBPI installed files on the local machine.

If the Event Importer cannot determine the information that it needs, the command fails.

Table 4 describes the options that are available for the `options` parameter within the `bpitveventimporter` command.

Table 4 Options for Event Importer

options Settings	Description
<code>-proceed value</code>	<p>value can be yes or no.</p> <p>If set to <code>yes</code>, proceed with the import and synchronization automatically.</p> <p>If set to <code>no</code>, report the changes that are required, but do not synchronize the changes. This enables you to preview the changes before committing to them.</p> <p>If omitted, the required changes are listed and you are asked whether or not to proceed. This setting is optional.</p>
<code>-rehost</code>	<p>If you do want to overwrite the event definitions, use the <code>-rehost</code> option.</p> <p>Normally, the importer does not allow the event definitions from one TransactionVision Analyzer server to overwrite those from another. If a naming clash is found, the Event Importer reports the problem and does not proceed with the import.</p> <p>You might need to use this option if you move from using one TransactionVision server to another, or to enable events that have not been imported from TransactionVision to be overwritten.</p> <p>If you use this option, events from another host are overwritten.</p>

Table 4 Options for Event Importer

options Settings	Description
-delete	<p>Use this option to request that the Event Importer deletes all the event definitions from the Model Repository that are associated with the named TransactionVision server. The TransactionVision server must be explicitly identified using the <code>-tvHost</code> setting.</p> <p>The Event Importer prompts for confirmation when all the events targeted to be deleted are listed.</p> <p>Use this option when you no longer want to synchronize with a particular TransactionVision server. You can also deleted the individual events from the HPBPI Modeler, but using this command line option more easily identifies the correct events to delete.</p> <p>This setting is optional.</p>

Examples Using the `bpitveventimporter` Command

This section provides some examples of the `bpitveventimporter` command and how it can be used.

1. In the following example, the Event Importer:
 - a. imports the transaction events from the local TransactionVision Analyzer.
 - b. connects to the Model Repository using the user and password credentials supplied with the `-bpiUser` and `-bpiPassword` options.

The `-proceed` option is not specified in this example, so the Event Importer lists the changes that need to be made for the synchronization and prompts for confirmation before proceeding.

```
bpitveventimporter -bpiHost bpiserver -bpiUser admin  
-bpiPassword pwd
```

2. In the following example, the Event Importer:
 - a. imports the transaction events from the local TransactionVision Analyzer.
 - b. connects to the Model Repository using the user and password credentials supplied with the `-bpiUser` and `-bpiPassword` options.
 - c. connects to the Model Repository and completes the synchronization.

As the `-proceed` option is specified, the Event Importer does not prompt for confirmation of the changes, but lists them prior to proceeding.

```
bpitveventimporter -bpiHost bpiserver -bpiUser admin  
-bpiPassword pwd -proceed yes
```

3. In the following example, the Event Importer imports the transaction events from the local TransactionVision analyzer and synchronizes the changes immediately with the local HPBPI Server system. As the `-proceed` option is specified, the Event Importer does not prompt for confirmation of the changes, but lists them prior to proceeding.

```
bpitveventimporter -proceed yes
```

4. In the following example, the Event Importer locates all the events defined in the file called `intermediate.xml` and verifies whether or not the event exists in the Model Repository on the HPBPI Server. Missing events are created, existing events are updated and redundant events are deleted. As the `-proceed` option is not specified, the Event Importer prompts with details of the changes, which you need to confirm.

```
bpitveventimporter -tvInfoFromXML intermediate.xml
```

5. In the following example, the Event Importer imports the transaction events from the local TransactionVision Analyzer and writes the results to an intermediate file called `intermediate.xml`.

```
bpitveventimporter -tvInfoToXML intermediate.xml
```


Configure a JMS Business Event Handler

You configure the integration with TransactionVision through the HPBPI Administration Console. HPBPI uses an instance of a JMS Business Event Handler to integrate with TransactionVision.

If HPBPI is on a different machine to TransactionVision, you need to configure a JNDI repository and add the JNDI entries for accessing the shared JMS queue on the messaging bus for the HPBPI and TransactionVision integration. Refer to the documentation for your JMS messaging bus for the details that you need for the JNDI repository configuration.

You then need to make sure that the correct JMS Queue and JNDI entries are set up for each JMS Business Event Handler

You also need to supply details of a Service Provider and Event Source as described in section [Service Provider](#) on page 113 and section [Event Source](#) on page 115.

Service Provider

You can configure the JMS Business Event Handler to receive events from one, or more, JMS sources. You configure a Service Provider for each JMS service on the TransactionVision server from which you want to receive event data. You need to configure a Service Provider before you can configure a JMS Event Source.

You configure the Service Providers settings 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 JMS Business Event Handler option in the left-hand navigation pane.
3. Click the Service Provider option, which is a sub option under JMS Business Event Handler.

A configuration page is opened in the right-hand pane, where you can add a new Service Provider.

4. Select the Add button to add a new Service Provider.

A JMS Service Provider Properties dialog is opened where you can enter values for the properties of the Service Provider. The properties are fully described in the *Business Process Insight Administration Guide*:

— Service Provider Name

Unique name used within HPBPI to identify the service provider. The name can be up to 40 characters.

— Description

Optional description of the service provider.

— Service Provider Classpath

Location, or set of locations, relating to the service provider Classpath.

— JNDI Provider Location

Location of the JNDI service for the named service provider, typically this is a URL.

— JNDI Factory Class

The Java class name for the initial context factory for the JNDI service.

— JMS Factory Class

The Java class name for the initial context factory for the specific JMS service provider.

5. Click the OK button to close the dialog.

6. Click the Apply button to apply your configuration changes.

The values for these parameters need to match the JNDI values that reference the JMS queue, configured for the HPBPI integration, on the messaging bus software. You need to refer to your messaging bus documentation for details of the values for some of the parameters; for example, Service Provider Classpath.

Refer to section [JMS Service Provider Settings](#) on page 124 for an example using WebSphere MQ.

Event Source

When you have configured a service provider, you can then configure an event source.

Having the event source configuration separate from the service provider configuration enables you to have multiple event sources for one service provider. The event source parameters enable you to configure JMS queue and login details.

You configure the Event Sources settings using the Administration Console as follows:

1. Start the Administration Console:

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

2. Click the JMS Business Event Handler option in the left-hand navigation pane.

3. Click the Event Source option, which is a sub option under JMS Business Event Handler.

A configuration page is opened in the right-hand pane, where you can add a new Event Source.

4. Select the Add button to add a new Event Source.

A JMS Event Sources Properties dialog is opened where you can enter values for the properties of the Event Source. The properties are fully described in the *Business Process Insight Administration Guide*:

- Service Provider Name

Select the name of a service provider that you have previously configured for this event source.

- JMS Queue Subject

The JNDI value that references the JMS queue at the selected service provider. JMS Queue Subject needs to reference the name of the JMS queue that has been configured for HPBPI to receive transaction events.

- User Name

The name of a user with authority to access the named JMS Queue Subject. Not all service providers require a user name to be specified.

- Password

Password for User Name where required.

5. Click the OK button to close the dialog.
6. Click the Apply button to apply your configuration changes.

When you have configured an event source, HPBPI generates the appropriate files to enable the JMS Business Event Handler to access the transaction events from the JMS queue. To complete the configuration, you need to stop and restart the JMS Business Event Handler from the Status screen on the Administration Console.

You can now receive events from the TransactionVision system. Any errors are reported to the JMS Business Event Handler log file, which can be accessed from the Status screen of the HPBPI Administration Console.

Refer to section [Event Source Settings](#) on page 125 for an example using WebSphere.

Example Configuration for the TransactionVision System

The following sections provide examples of the configuration required on the TransactionVision system. Unless otherwise indicated, the examples use WebSphere MQ 6.0 as the messaging bus.

You should refer to the TransactionVision documentation for full details of these tasks and the tools and interfaces available to complete them. This section describes how to:

- [Create a JMS Queue for the Integration](#) on page 117
- [Set up the JNDI Entries on the TransactionVision System](#) on page 118
- [Configure TransactionVision to Send Events to the JMS Queue](#) on page 120

Create a JMS Queue for the Integration

Complete the following to set up a JMS queue for the integration.

When you create the queue, make a note of the name that you assign, as when you configure HPBPI, you need to name this queue in the HPBPI configuration.

Specify the name of the queue manager in the `runmqsc` command as follows:

```
$ cd c:\Program files\IBM\WebSphere MQ\bin
$ runmqsc.exe QueueManager
  define QLOCAL(BPIQUEUE) MAXDEPTH(1000000)
  end
```

This command creates a JMS queue called `BPIQUEUE`, with a maximum queue depth of 1000000. This means that 1000000 is the maximum number of events that can be held on the JMS queue.

Set up the JNDI Entries on the TransactionVision System

When you have created the JMS queue, you then need to configure the JNDI reference to the JMS queue, on the TransactionVision system. This enables TransactionVision to access the configured JMS queue on the messaging bus.

Set up your JNDI entries for WebSphere MQ as follows:

1. Make sure that you have installed the required WebSphere MQ client software on the TransactionVision system.
2. Edit the file `JMSAdmin.config`, located in the WebSphere MQ installation directory:

```
C:\Program Files\IBM\WebSphere MQ\Java\bin\JMSAdmin.config
```

Ensure that it is using a file context by removing the comment character from the appropriate lines. (Read the comments in the `JMSAdmin.config` for further explanation.). As an example, remove the comment character from the following two lines:

```
INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory  
PROVIDER_URL=file:/C:/temp/JNDI-Directory
```

3. Create the following directory, if it does not already exist:

```
C:\temp\JNDI-Directory
```

4. Run the following script:

```
C:\Program Files\IBM\WebSphere MQ\Java\bin\JMSAdmin.bat
```

5. Create the JNDI entry for the queue connection factory, using the `JMSAdmin def` command:

```
def qcf(QueueConnectionFactory) hostname(Mom-HostName)  
qmanager(Queue-Manager) transport(CLIENT)  
channel(qmanager.chl) port(Port-Number)
```

where:

- *QueueConnectionFactory* is set to the name of the queue connection factory.
- *Mom-HostName* is set to the name of the server where WebSphere MQ is running.
- *Queue-Manager* is set to the name of the queue manager.

- `qmanager.chl` is set to the channel you need to connect to the queue.
- `Port-Number` is set to the port number of the queue manager listener.

For example:

```
def qcf(myQCF) hostname(MB-Server.domain.com)
qmanager(QManager) transport(CLIENT) channel(QManager.CHL)
port(1421)
```

6. Create the JNDI entry for the queue reference, using the `JMSAdmin` `def` command:

```
def q(LocalQueueName) queue(ActualQueueNameOnTheMBServer)
```

For example:

```
def q(theBPIQueue) queue(BPIQUEUE)
```

This creates a queue reference called `theBPIQUEUE`, which references the the queue called `BPIQUEUE` on the WebSphere MQ system.

When you have completed the configuration, you can check the JMS configuration by running the following script on the machine where the WebSphere MQ is running:

```
C:\Program Files\IBM\WebSphere MQ\Java\bin\JMSAdmin.bat
```

Use the `display` command, for example:

```
display ctx
display qcf(myQCF)
display q(theBPIQueue)
end
```

You specify the queue details when you configure the integration with HPBPI from the [TransactionVision Integration Configuration](#) page. The [Business Process Insight](#) page enables you to enter the values for the JMS settings, including the JNDI reference (within [TransactionVision](#)) on a Web-based form.

Configure TransactionVision to Send Events to the JMS Queue

To configure TransactionVision to send events to the JMS queue, complete the steps described in sections:

- [Configure a JMS Queue and JNDI Repository On the TransactionVision System](#) on page 101.
Use the information that you configured in the sections above.
- [Classify Events](#) on page 102.

Example Configuration for the HPBPI System

The following sections provide examples of the configuration required on the HPBPI system. Unless otherwise indicated, the examples use WebSphere MQ 6.0 as the messaging bus. The section describes how to:

- [Set up JNDI Entries on the HPBPI System](#) on page 121.
- [Configure a JMS Business Event Handler for HPBPI](#) on page 123.

Set up JNDI Entries on the HPBPI System

If HPBPI is running on a different system to that of TransactionVision, you need to configure the JNDI entries to reference the JMS queue on the messaging bus that was set up for the integration.

If HPBPI is running on the same system as TransactionVision, you do not need to complete the tasks described in this section as the JNDI entries were created as part of the TransactionVision configuration in section [Set up the JNDI Entries on the TransactionVision System](#) on page 118.

1. Make sure that you have installed the required WebSphere MQ client software on the HPBPI system.
2. Create a new directory and file:
`C:\temp\JNDI-Directory`
3. Edit the file `JMSAdmin.config`, located in the WebSphere MQ installation directory on the HPBPI system:
`C:\Program Files\IBM\WebSphere MQ\Java\bin\JMSAdmin.config`
4. Ensure that the file is using a file context by removing the comment character from the appropriate lines. (Read the comments in the `JMSAdmin.config` for further explanation.). As an example, remove the comment character from the following two lines:

```
INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory  
PROVIDER_URL=file:/C:/temp/JNDI-Directory
```

5. Run the script:

```
C:\Program Files\IBM\WebSphere MQ\Java\bin\JMSAdmin.bat
```

- Create the JNDI entry for the queue connection factory, using the `JMSAdmin def` command:

```
def qcf(QueueConnectionFactory) hostname(MB-HostName)
qmanager(Queue-Manager) transport(CLIENT) channel(QM.chl)
port(Port-Number)
```

where:

- *QueueConnectionFactory* is set to the name of the queue connection factory.
- *MB-HostName* is set to the name of the messaging bus.
- *Queue-Manager* is set to the name of the queue manager that manages the BPI queue you're trying to point to.
- *QM.chl* is set to the channel you need to connect to the queue.
- *Port-Number* is set to the port number of the queue manager listener.

For example:

```
def qcf(myBPIQCF) hostname(MB-Server.domain.com)
qmanager(QManager) transport(CLIENT) channel(QManager.CHL)
port(1421)
```

6. Create the JNDI entry for the queue reference, using the `JMSAdmin def` command:

```
def q(LocalQueueName) queue(ActualQueueNameOnTheMBServer)
```

For example:

```
def q(myBPIQueuePointer) queue(BPIQUEUE)
```

This creates a local JNDI queue reference called `myBPIQueuePointer` which references the JMS queue created for the integration on the messaging bus called `BPIQUEUE`.

When you have completed the configuration, you can check the JMS configuration by running the following script on the machine where the WebSphere MQ is running:

```
C:\Program Files\IBM\WebSphere MQ\Java\bin\JMSAdmin.bat
```

Use the display command, for example:

```
display ctx
display qcf(myBPIQCF)
display q(myBPIQueuePointer)
end
```

As a result of this configuration, when you connect to the JMS queue called myBPIQueuePointer, you are connecting to the JMS queue (BPIQUEUE) on the messaging bus.

Configure a JMS Business Event Handler for HPBPI

The following sections provide examples of the configuration parameters that you can set for HPBPI when using WebSphere MQ:

- [JMS Service Provider Settings](#) on page 124
- [Event Source Settings](#) on page 125

JMS Service Provider Settings

If you are configuring the service provider details for a WebSphere MQ JMS implementation, the following is an example of the configuration settings you might set for HPBPI.

Service provider name	WebSphereMQ
Description	TV connection using WebSphere MQ
Service Provider Classpath	C:\Program Files\IBM\WebSphere MQ\ Java\lib\jms.jar; C:\Program Files\IBM\WebSphere MQ\ Java\lib\com.ibm.mqjms.jar; C:\Program Files\IBM\WebSphere MQ\ Java\lib\com.ibm.mq.jar; C:\Program Files\IBM\WebSphere MQ\ Java\lib\fscontext.jar; C:\Program Files\IBM\WebSphere MQ\ Java\lib\jndi.jar; C:\Program Files\IBM\WebSphere MQ\ Java\lib\providerutil.jar; C:\Program Files\IBM\WebSphere MQ\ Java\lib\connector.jar
JNDI Provider Location	file:///C:/temp/JNDI-Directory
JNDI Factory Class	com.sun.jndi.fscontext.RefFSContextFactory
JMS Factory Class	myBPIQCF

Event Source Settings

If you are configuring the service provider details for a WebSphere MQ JMS implementation, the following is an example of the configuration settings you might set for HPBPI:

JMS Queue Subject	myBPIQueuePointer
User Name	not required for WebSphere MQ when running on the local system.
Password	not required for WebSphere MQ when running on the local system

The example shows an event source for a JMS Queue Subject called myBPIQueuePointer. There are no user name and password details required for this queue.

6 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 128.
- Obtaining status information from SOA Manager; see section [Business Events and SOA Manager](#) on page 131.
- Configuring the HPBPI SOA Manager Adapter; see section [Configuring Access to the SOA Manager Adapter](#) on page 132.

SOA Manager and HPBPI Integration

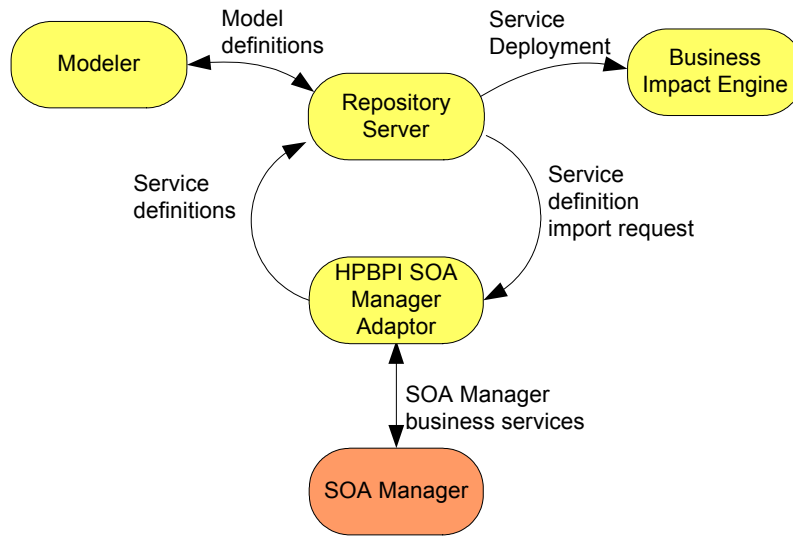
HP SOA Manager enables you to manage your service oriented architecture (SOA) resources to ensure their reliability and optimize their performance. It is a model-based management solution for managing a SOA in the context of understanding the health of the business services that the SOA architecture 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 27](#) on page 129. Configuring the SOA Manager Adapter is described in section [Configuring Access to the SOA Manager Adapter](#) on page 132.
2. The second is where HPBPI can receive business events from SOA Manager for the HPBPI business processes that it is monitoring. This integration is described in section [Business Events and SOA Manager](#) on page 131. 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 27 shows how the SOA Manager Adapter enables you to link to SOA Manager business services and make them available to HPBPI business processes through the Repository Server.

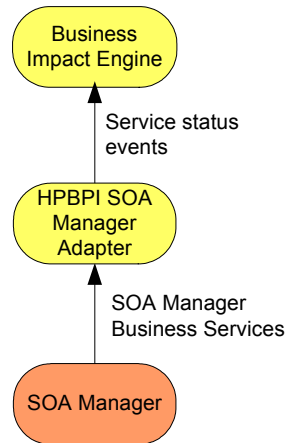
Figure 27 HPBPI and SOA Manager Business Services



When you have configured the SOA Manager Adapter, you can import the required SOA Manager business services and link them to your business processes. 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 process and the services are deployed to the Business Impact Engine to be monitored; see [Figure 28](#).

Figure 28 HPBPI SOA Manager Adapter - Status Propagation



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 processes are monitored and managed by the Business Impact Engine.

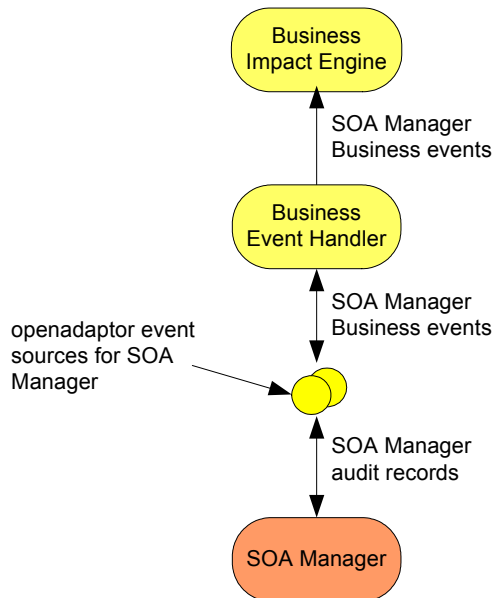
The HPBPI SOA Manager adapter polls SOA Manager for updates to the business services that it is reporting on. These are the services that you have linked to through the HPBPI Modeler.

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 29](#). 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 29 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.

A Expression Grammar in Process, Data and Filter Definitions

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

The progression rules and expressions are within the HPBPI Modeler for the start and complete conditions for steps, and for filter expressions for Data definitions when subscribing to events.

Business process monitor filter expressions are used within the Monitor Definer to optionally specify the conditions for collecting statistical data for the business process monitors.

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 Processes*.

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 138 for possible values for `function_return_value`.


```
constant_value =>
    string_constant
    | 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 Processes*:

- `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.

Process Progression Rules

The grammar for process 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 142.

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 step. Ultimately, each style uses a subset, or selection of the methods described in section [Methods for Progression Rules](#) on page 142. 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 processes is provided in the *Business Process Insight Integration Training Guide - Modeling Processes*.

The methods used for the progression rule styles are listed in [Table 5](#) on page 140.

Table 5 Methods Used for Progression Rule Styles

Style	Methods Used	Comments
Complete on first assignment	<code>before() == null</code>	This style of progression rule uses the method listed to determine when the property value changes from Null to any other value.

Table 5 Methods Used for Progression Rule Styles

Style	Methods Used	Comments
Complete on transition	Complete Condition: <code>before().in()</code> and <code>after().in()</code>	This style of progression rule uses the methods listed to determine when the step 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 <code>in()</code> 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: <code>before().in()</code> and <code>after().in()</code> Complete Condition: <code>before().in()</code> and <code>after().in()</code>	This style of progression rule uses the methods listed to determine when the step start and step 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 <code>in()</code> 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 process progression rules.

Table 6 Methods for Progression Rules

Method	Description
<code>this.data.property.changed()</code>	This function is used to test when the property value changes. When the property value changes it returns a result of <code>True</code> .
<code>this.data.property.before() == "prop-value"</code>	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 <i>prop-value</i> and is now no longer that value. The type returned is the same as the type of the property.
<code>this.data.property.after() == "prop-value"</code>	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 <i>prop-value</i> and is now <i>prop-value</i> . The type returned is the same as the type of the property.

Table 6 Methods for Progression Rules

Method	Description
<code>this.data.created()</code>	<p>This evaluates to true when the data definition is created. This method is executed only when a new Data definition is created.</p> <p>The method returns a Boolean result.</p>
<code>this.data.property.in("prop-value1", "prop-value2", ...)</code>	<p>This evaluates to true when the value of the property is set to any of the listed values.</p> <p>The method returns a Boolean result.</p>
<code>this.data.terminated()</code>	<p>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.</p> <p>The method returns a Boolean result.</p>

Example of Complete Condition for Start Step When Data Definition Created

The following shows an example of the expression that you use when you want a process instance to be started when a Data definition for the process is created. In this case, a new process instance is started only when the start and complete conditions for the start step in the process are met, for example:

```
this.order.created()
```

where:

- `this` is the reference to the process, in this case the Order process.
- `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 Step When Data Definition Modified

The following shows an example of the expression that you use when you want a process instance to be started when the Data definition for the process is modified. In this case a process 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 process, in this case the Order process.
- `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 process instance to be started when the Data definition for the process is modified and the Data definitions contains a specific string.

```
this.order.customer_id.after().contains("X")
```

where:

- `this` is the reference to the process, in this case the `Order` process.
- `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"`

B 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 7 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 <i>prop-name</i> to property <i>prop-name</i> 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 <i>prop-name</i> to property <i>prop-name</i> 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 8

Operator	Behavior
Arithmetic: <ul style="list-style-type: none">• +• -• *• /• %	Supported between Numeric properties or values that can be coerced to numeric using the assignment rules described in Table 7 .
Comparison: <ul style="list-style-type: none">• <• <=• =• !=• >• >=	<p>The comparison operators can be used to provide a:</p> <ul style="list-style-type: none">• numeric comparison between Numeric properties or values that can be coerced to numeric using the assignment rules described in Table 7.• 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 7 (including Boolean). <p>The result of a comparison is always a boolean value.</p>
String tests: <ul style="list-style-type: none">• starts()• ends()• contains()• in()	<p>Performs a test on a String property.</p> <p>Note that these tests cannot be applied to non-string properties.</p>

Table 8

Operator	Behavior
Logical: <ul style="list-style-type: none">• ! (Not)• && (And)• (Or)	Supported between Boolean properties or values that can be coerced to boolean using the assignment rules described in Table 7 .
Conditional value <code>if</code> , <code>then</code> , <code>else</code> : <ul style="list-style-type: none">• ?:	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 9

Operator	Comparison
<ul style="list-style-type: none">• ==• !=	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 <i>index</i> is null: <ul style="list-style-type: none">• <i>index</i> > 10• <i>index</i> <= 10
Arithmetic operators	Arithmetic operators for expressions containing a null value always return null as the result, for example: <i>index</i> + 1 returns null if <i>index</i> 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

Adapter

- Business Event Handler, 39
- for HPBPI Server, 27
- for iWay integration, 60

Administration Console

- architecture, 46
- HPBPI Notification Server, 46
- HPBPI Server Administration Console, 46

Advanced conditions

- grammar rules, 140
- process progression rules, 140

Architecture

- Administration Console, 46
- and HPBPI database, 18
- Business Event Handler, 38
- Business Impact Engine, 25
- Business Process Dashboard, 13
- diagram for HPBPI, 12, 50
- for Business Availability Center, 68
- HPBPI, 11, 49
- HPBPI Operations Manager Adapter, 56
- HPBPI Server, 23
- Intervention Client, 16
- JMS Business Event Handler, 42
- Monitor Definer, 29
- Monitor Engine, 31
- Notification Server, 35
- Repository Explorer, 34
- Security, 47
- Self-Healing Services, 63

Arithmetic operator

- coercion, 149

Assignment

- coercion and data types, 148
- expressions and coercion rules, 147

B

backlog

- Data Sample for Business Availability Center, 90

BIAEngineAdaptor

- Engine Adapter, 40

- Binding expressions
 - coercion rules, 147
- Boolean
 - data type coercion rule, 148
- BPEL
 - importing into Model Repository, 44
- BPI
 - Administration Console, 46
 - and iWay, 60
 - and SAP, 59
 - Modeler, 44
 - Model Repository, 44
 - process model deployer, 45
 - propagating events, 40
- BPI. See also HPBPI
- Business Availability Center
 - adapter colleague for HPBPI Server, 28
 - and BPI data sampling, 72
 - and MY BAC, 68
 - and Reporting, 89
 - and Select Access, 84
 - and Servlet Engine authorization, 84
 - and SLM, 89
 - and SLOs, 68
 - architecture diagram, 68
 - configuring for HPBPI, 87
 - configuring the integration, 77
 - creating a portlet Server for HPBPI, 88
 - data sample mapping, 90
 - design-time integration, 69
 - exporting HPBPI Process definitions, 66
 - exporting monitor definitions, 67
 - integrating with HPBPI, 65
 - integration
 - Data Sample settings, 80
 - General settings, 78
 - Service Source settings, 82
 - load balancing, 76
 - modeling IT processes, 66
 - Monitor Definer, 29
 - monitoring business impact data from HPBPI, 68
 - monitoring by HPBPI, 54
 - operational service status, 68
 - run-time integration, 71
 - security, 84
 - service status, 72
 - single sign on, 84
 - using SSL, 86
 - Viewing processes and monitors, 68
- Business Event Handler
 - architecture, 38
 - event adapter, 39
 - HPBPI Server component, 23
 - openadapter, 38
 - receiving events, 40

- business health
 - Data Sample for Business Availability Center, 90

- Business Impact Engine
 - architecture, 25
 - BIAEngineAdaptor, 40

- Business Monitors
 - architecture diagram for Business Process Monitor Definer, 30

- Business object manager
 - HPBPI Server, 27

- Business Process Dashboard
 - and business process monitors, 15
 - architecture, 13

- Business process monitor
 - data in the HPBPI database, 20

- Business Process Monitor Definer
 - and Business Availability Center, 29

- Business process monitors
 - Business Process Monitor Definer an HPBPI Server component, 29

C

- Case sensitivity
 - for expressions within progression rules, 146

- Classify events
 - TransactionVision, 102

- Classifying transactions
 - for TransactionVision integration, 100

- Coercion
 - arithmetic expressions, 149
 - comparison expressions, 149
 - conditional expressions, 150
 - for expressions, 149
 - logical expressions, 150
 - rules for binding, filter and assignment
 - expressions, 147
 - string expressions, 149

- Comparison operator
 - coercion, 149

- Complete condition
 - after(), 142
 - before(), 142
 - changed(), 142
 - created(), 143
 - in(), 143
 - methods for, 142
 - terminated(), 143

- Complete on first assignment
 - grammar rules, 140

- Complete on transition
 - grammar rules, 140

- Component
 - Model Repository, 44
 - Process model Deployer, 45

- Conditional operator
 - coercion, 150

- Configuration items
 - stored in UCMDB for HPBPI, 74

- Configuring
 - Business Availability Center integration, 77
 - Event Source for TransactionVision, 115
 - HP Operations Dashboard integration, 61
 - JMS Business Event Handler for TransactionVision, 113
 - Service Provider for TransactionVision, 113

- Console
 - HPBPI Administration Console, 45
 - Notification Server Administration Console, 45

- Create a JMS queue
 - TransactionVision, 101

D

- Dashboard
 - and business process monitors, 15
 - and HP Operations Dashboard, 61
 - architecture, 13

- Data Sample
 - backlog, 90
 - business health, 90
 - duration, 90
 - throughput, 90

- Data sample mapping
 - and Business Availability Center, 90

- data synchronization, 75

- Data type
 - assignment coercion, 148

- Date
 - data type coercion rule, 148
 - functions, 138

- Defining
 - expressions for processes, 135

- duration
 - Data Sample for Business Availability Center, 90

E

- Email
 - notification data in HPBPI database, 22

- Engine
 - HPBPI Server component, 23

- Event
 - hospital data and HPBPI database, 21
 - propagation, 40
 - receivers and transmitters for the HPBPI Server, 28

- Event adapter
 - Business Event Handler, 39

- Event Handler
 - See Business Event Handler
 - See JMS Business Event Handler

- Event Importer
 - for importing TransactionVision events, 105
 - settings for, 107

- Event Source
 - configuring for TransactionVision, 115

- Example
 - start condition for Start Step
 - Data definition created, 144
 - Data definition modified, 144
 - start condition using a method on a method, 145

- Expression
 - coercion, 149
 - grammar, 135
 - properties
 - case sensitivity for Process definitions, 146
 - using NULLs, 151

F

- Filter expression
 - coercion rules, 147
- Functional return values
 - process expressions, 138

G

- Grammar rules, 135
 - Data definitions, 135
 - Monitor definitions, 135
 - Process definitions, 135

H

HPBPI

- Administration Console, 45
 - and Business Availability Center data sampling, 72
 - and SOA Manager integration, 127
 - architecture, 11, 49
 - architecture diagram, 12, 50
 - Business Availability Center portlet server, 88
 - Business Event Handler, 38
 - importing SOA Manager business events, 128, 131
 - importing SOA Manager status events, 128
 - JMS Business Event Handler, 42
 - monitoring Business Availability Center, 54
 - monitoring SOA Manager, 56
 - Notification Server Administration Console, 45
 - openadapter, 38, 42
 - Server components, 23
- HPBPI Accelerator
 - for SAP integration, 59
 - HPBPI Adapters, 27

HPBPI database

- business process monitor data, 20
 - diagram of relationship with HPBPI components, 18
 - email notification data, 22
 - event hospital data, 21
 - model repository data, 21
 - monitor threshold data, 21
 - part of architecture, 18
 - process data, 19
- HPBPI Intervention Client, 46
 - HPBPI Modeler
 - and BPEL, 44
 - grammar description, 136
 - grammar rules, 135
 - HPBPI Notification Server
 - Administration Console, 46
 - HPBPI Operations Manager Adapter
 - architecture, 56
 - HPBPI Server
 - architecture, 23
 - Business Availability Center Adapter Colleague, 28
 - business object manager, 27
 - Business Process Monitor Definer component, 29
 - event receivers and transmitters, 28
 - Monitor Engine, 31
 - Notification Server, 35
 - Repository Explorer, 34
 - SOA Manager Client Colleague, 28
 - HPBPI SOA Manager
 - configuring access to adapter, 132
 - HP Business Availability Center
 - configuring the integration, 77
 - integration
 - Data Sample settings, 80
 - General settings, 78
 - Service Source settings, 82

- HP Operations Dashboard
 - and HPBPI Dashboard, 61
 - single signon, 62
- HP Operations Dashboard integration, 61
- HTTPS
 - and Business Availability Center, 86

I

- Importing TransactionVision events
 - into Model Repository, 105
- Integrating with iWay, 60
- Integrating with SAP, 59
- Integrating with the Business Availability Center, 65
- Integrating with TransactionVision, 93, 94
- Integration
 - with TransactionVision, 96, 98
- Intervention Client
 - architecture, 16
- iWay integration, 60

J

- JMS Business Event Handler
 - architecture, 42
 - configuring for TransactionVision, 113
 - openadaptor, 42
- JMS queue
 - configuring for TransactionVision integration, 101
 - creating for TransactionVision integration, 100
- JNDI repository
 - configuring for TransactionVision integration, 101

L

- Load balancing
 - Business Availability Center, 76
- Logical operator
 - coercion, 150

M

- Modeler
 - Advanced conditions grammar, 140
 - Complete on first assignment grammar, 140
 - Complete on transition grammar, 140
 - functional return values for process definitions, 138
 - grammar description, 136
 - grammar rules, 135
 - Process definition language, 135
 - progression rules
 - defining for processes, 140
 - Start and complete on transition grammar, 140
 - String values for Process definitions, 138
- Modeling IT processes
 - using the Business Availability Center, 66
- Model Repository
 - data and the HPBPI database, 21
 - description, 44
- Monitor data
 - HPBPI database, 20
- Monitor Definer
 - and Business Availability Center, 29
 - architecture, 29
- Monitor Engine
 - architecture, 31
 - HPBPI Server, 31
- Monitoring Business Availability Center, 54
- Monitoring SOAM Manager, 56

Monitor threshold data
 HPBPI database, 21

Monitor thresholds
 grammar rules, 135

MY BAC
 for HPBPI, 88

N

Notification Server
 Administration Console, 45
 architecture, 35
 component of BPI Server, 35
 HPBPI Server component, 23
 Velocity email templates, 37
 XSLT email templates, 37

NULL
 when used within an expression, 151

Numeric
 data type coercion rule, 148

O

openadaptor
 handling HPBPI business events, 38, 42

P

Process
 data and the HPBPI database, 19
 expressions
 functional return values, 138
 progression rules
 process definitions, 140

Process definition, 135
 expression grammar, 135
 expression properties
 case sensitivity, 146
 functions for dates and times, 138

Modeler
 date and time functions for Process
 definitions, 138
 Progression rules, 140
 String values, 138
 Time functions, 138

Progression rule
 after(), 142
 before(), 142
 case sensitivity for expressions, 146
 changed(), 142
 created(), 143
 in(), 143
 methods for, 142
 terminated(), 143

Propagating events, 40

R

Reporting
 using Business Availability Center, 89

Repository Explorer
 architecture, 34
 component of HPBPI Server, 34

Rules
 for Modeler grammar, 135
 for thresholds, 135

S

SAP integration, 59

Security
 and Business Availability Center, 84
 architecture, 47
 Select Access, 47
 Tomcat Servlet Engine, 47

- Select Access
 - and Business Availability Center, 84
 - security, 47
- Self-Healing Services
 - architecture, 63
- Service Level Management
 - using Business Availability Center, 89
- Service Provider
 - configuring for TransactionVision, 113
- Service status
 - Business Availability Center, 72
- Servlet Engine authorization
 - and Business Availability Center, 84
- Settings for Event Importer
 - TransactionVision, 107
- Single sign on
 - and Business Availability Center, 84
- Single signon
 - for HP Operations Dashboard links to HPBPI, 62
- SLM
 - using Business Availability Center, 89
- SOAM
 - monitoring by HPBPI, 56
- SOA Manager
 - business events, 128, 131
 - client colleague for HPBPI Server, 28
 - configuring access to adapter, 132
 - integration with HPBPI, 127, 128
 - status of Business Services, 128
- SSL
 - using with Business Availability Center, 86
- Start and complete on transition
 - grammar rules, 140

- Start condition
 - after(), 142
 - before(), 142
 - changed(), 142
 - created(), 143
 - in(), 143
 - methods for, 142
 - terminated(), 143
- String
 - data type coercion rule, 148
 - operator for coercion, 149
 - values for process definitions, 138
- Synchronization
 - HPBPIdata in UCMDB, 75

T

- Threshold
 - grammar rules, 135
- Threshold definitions, 135
- throughput
 - Data Sample for Business Availability Center, 90
- Tomcat Servlet Engine
 - security, 47

TransactionVision

- classifying events, 102
- configuration example (WebSphere MQ), 117
- configuration tasks, 100
- configuring HPBPI, 104
- configuring integration, 100
- create a JMS queue, 101
- design-time integration, 96
- HPBPI integration, 94
- importing events into Model Repository, 105
- integrating with HPBPI, 93
- run-time integration, 98
- using for importing transaction events, 95

U

- UCMDB, 75
 - and HPBPI configuration items, 74

