

HP

Business Process Insight

For the Windows® Operating System

Software Version: 2.20

Accelerator for SAP Applications Guide

Document Release Date: October 2007

Software Release Date: October 2007



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

© Copyright 2007 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://ovweb.external.hp.com/lpe/doc_serv/

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

Support

You can visit the HP Software Support web site at:

www.hp.com/go/hpsoftwaresupport

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

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

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

To find more information about access levels, go to:

www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

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

Contents

1	HPBPI and SAP Overview	9
	Business Process Insight Accelerator for SAP	10
	A Brief History of SAP Applications	12
	System/R	12
	R2	12
	R3	12
	SAP Web Application Server	13
	Exchange Infrastructure	13
	NetWeaver	14
	SAP Terminology	15
	Intermediate Document	15
	Application Link Enabling	15
	Advanced Business Application Programming	16
	NetWeaver	16
	Exchange Infrastructure	16
	Business Application Programming Interface	16
	Remote Function Call	17
	Work Flow	17
	User Exit	17
	Basis Administrator	18
2	Installing the HPBPI Accelerator for SAP	19
	Before Starting the Installation	20
	Windows	20
	HP-UX	21
	Windows and HP-UX Software Requirements	22
	Installing XPL Viewer	23
	SAP Transport Files	24

Installing the Accelerator	26
Installation on Windows.....	26
Installation on Unix	27
Obtaining and Installing the SAP Libraries	28
Setting up SAP User for the Accelerator	31
Checking the Accelerator Installation	33
Licensing and the HPBPI Accelerator for SAP	35
Trial License Extension	35
Permanent License Key Password.....	36
HPBPI Files Duplicated Under the Accelerator Installation Directory	37
Removing the Accelerator for SAP	38
Removing the Accelerator from a Windows Machine	38
Removing the Accelerator from an HP-UX Machine	39
Re-Installing the Accelerator	40
3 Configuring the HPBPI Accelerator for SAP	41
The Adaptor and Business Events	42
Adaptor as Data Source	43
Configuring the Adaptor	45
Identifying IDocs to Monitor	45
Adaptor Properties File	45
Password Encoding Utility.....	55
Starting the Adaptor on Windows	56
Java Service Wrapper Configuration.....	57
Starting the Adaptor on HP-UX	59
Remote Control	60
Stopping the Adaptor on HP-UX.....	61
Logging.....	62
Configuring HPBPI to Monitor SAP Events	63
Example - Configuring Idoc Adaptors.....	64
Example SAP System.....	64
Adaptor Property Files.....	65
Event and Data Definitions for HPBPI.....	71
Example - Configuring Name Mapping	75
The Configuration Segments	77

An Example Adaptor Properties File.....	78
A SAP EDIDC Table Fields.....	81
B IDoc Status Fields.....	85
C Additional HPBPI and SAP Integration Options.....	89
Integrating with SAP.....	90
Business Monitoring and BPM.....	90
Integration Options.....	91
SAP Product Knowledge.....	92
Generating Business Data for HPBPI.....	93
Pull Integrations.....	94
BAPI/RFC.....	94
Programmatic SQL Selects.....	95
Push Integrations.....	96
Custom DB Triggers.....	96
IDocs.....	97
ALE.....	98
User Exits.....	100
XI.....	102

1 HPBPI and SAP Overview

The chapter introduces the HP Business Process Insight Accelerator for SAP, which is an HP product that you can use to monitor the status of your IDocs within the SAP environment.

This chapter also gives a brief overview of the history of SAP applications and SAP terminology.

The chapter is structured as follows:

- [Business Process Insight Accelerator for SAP](#) on page 10
- [A Brief History of SAP Applications](#) on page 12
- [SAP Terminology](#) on page 15

Business Process Insight Accelerator for SAP

The HPBPI Accelerator for SAP Applications comprises an adaptor that enables you to monitor the status fields within an IDoc. IDocs are used to transfer business data between SAP applications, and between SAP and external systems (SAP client systems); for example, the status of Materials documents.

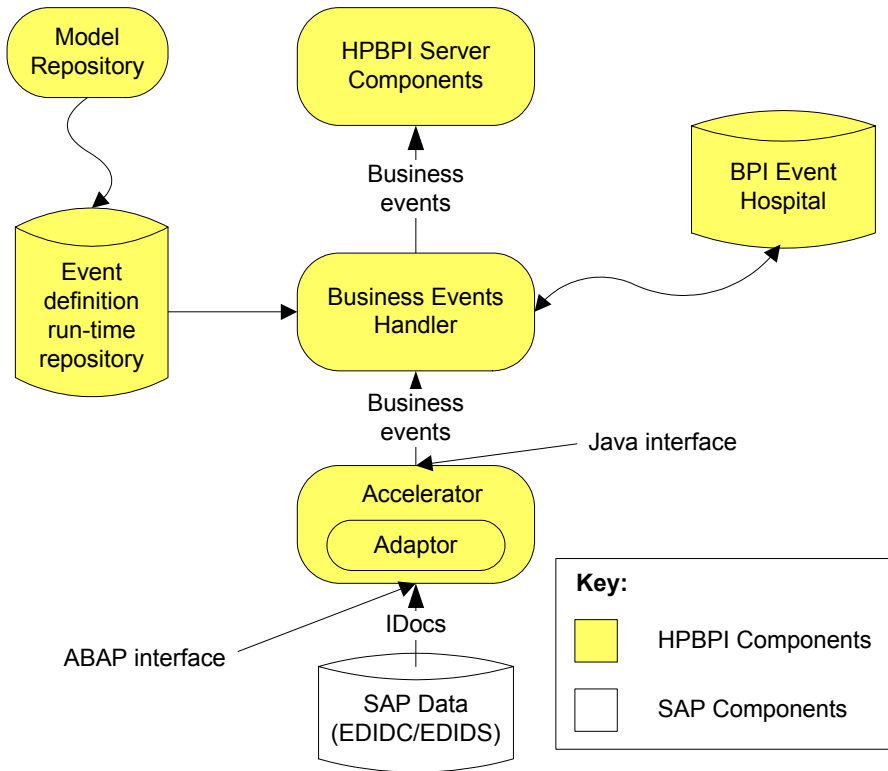
IDoc status changes, or transitions, are considered critical points within an SAP business process, and the HPBPI Accelerator adaptor enables you to use HPBPI to monitor these transitions and report on the health of the associated business processes.

By using the HPBPI Accelerator for SAP with HPBPI, you can use HPBPI to create business flows within HPBPI. These business flows can then have Event definitions that are based on the status changes of the IDocs that the Accelerator is accessing within SAP business system.

The Accelerator uses the SAP ABAP libraries to access the status information within an IDoc header. The Accelerator then sends this information to the Business Events Handler, which in turn passes the information on to the Business Impact Engine.

Figure 1 shows the high-level architecture of this integration using the HPBPI Accelerator.

Figure 1 HPBPI Accelerator



The remainder of this chapter provides some background information on the history of SAP applications and also defines some of the SAP terms used in this document. If you are familiar with SAP you can skip these sections and continue at [Chapter 2, Installing the HPBPI Accelerator for SAP](#), which provides the instructions for installing the Business Process Insight Accelerator for SAP.

A Brief History of SAP Applications

In 1972, five former IBM employees founded the German company SAP (Systems, Applications and Products in Data Processing). Their vision was to develop and market enterprise software which would integrate all business processes.

System/R

The first product that SAP developed was called System/R and consisted of the following application packages:

- a financial package
- a materials management package.

R2

R/2 was the next version of System/R and introduced more application packages; it was still primarily for IBM mainframes.

With the R/2 release, SAP developed their own report writing language. This language was called “Allgemeiner Berichtsaufbereitungsprozessor” (ABAP) - which is German for “generic report preparation processor”. ABAP grew into a complete programming language and was later renamed Advanced Business Application Programming. ABAP, with its object oriented extensions, is a very powerful language with a syntax similar to Cobol. ABAP soon became the language of choice for the SAP developers and nowadays most SAP applications are written in ABAP.

R3

R/3 was the third release of System/R. This was a significant release for SAP as the R/3 release implemented a client/server architecture. SAP applications could now extend into the emerging client/server market and were no longer restricted to mainframes. With this release, SAP introduced a number of new applications and R/3 became the Enterprise Resource Planning (ERP) solution for SAP. It addressed areas such as manufacturing, supply-chain, distribution, finance, customer management, human resources and more.

However, with an expanding customer base, the need for application integration became more important for SAP. In most cases, integrations were carried out on a point-to-point basis; that is, each integration required custom development to enable the applications to share and communicate data.

One of the earliest mechanisms introduced by SAP to help integrate SAP applications was the Intermediate Document (IDoc). An IDoc is a data container with a format defined and documented by SAP and it can be held in a database and can also be sent as a text file. SAP introduced the idea that one SAP application could be configured to write out information to an IDoc, and another application could be configured to read this IDoc. This enabled information to be transferred from one application to another. IDocs are still very much a part of SAP and used extensively in current implementations.

In release 3.0 of R/3, SAP introduced Application Link Enabling (ALE). This made it much easier to configure SAP applications to pass data between themselves, and to third party systems. It was an early form of middleware technology. Specifically, ALE enables you to configure SAP to monitor a business object and to send notifications when the business object changes. ALE sends this notification data within an IDoc, and you can configure transformations to this data. ALE is a powerful integration tool.

SAP then went on and developed products that enabled applications to communicate and easily share data.

SAP Web Application Server

SAP then looked at developing a platform to enable their applications to run within an application server. This was the SAP Web Application Server. Originally this application server ran only ABAP programs; however, it was later made J2EE compliant, thus adding support for Java.

Exchange Infrastructure

Exchange Infrastructure (XI) is a middleware bus that SAP developed in order to more easily integrate SAP applications with third party applications. SAP then provided the Business Process Management (BPM) product to orchestrate the messages exchanged across the XI bus.

NetWeaver

NetWeaver is a suite of SAP products, which are provided under a single license to an Enterprise. It is a new integration platform and, with the exception of R/3 ERP, NetWeaver includes all of its existing J2EE products, and its J2EE application server. The NetWeaver application server supports both ABAP and Java application code.

NetWeaver was the application that SAP used to encourage their customers to to run their business in a single platform; that is, the SAP application server.

SAP Terminology

The following sections describe some of the key SAP terms, which are used in this guide and within the SAP community.

Intermediate Document

An Intermediate Document (IDoc) is a data container with a format defined and documented by SAP. An IDoc can be held in a database or sent as a text file, or both held in a database and sent as a text file. IDocs are one of the original mechanisms used for exchanging data asynchronously between two or more SAP applications. Typically an IDoc contains the multi-line data for a single transaction.

If one application can write out information to an IDoc, a second application can then read in this IDoc.

An IDoc can be generated at any point within an SAP transaction (or work flow; see [Work Flow](#) on page 17). As an example, in a shipping transaction, an IDoc might be generated to include the data fields required to print a shipping manifest.

After a user completes an SAP transaction, one or more IDocs can be generated and potentially passes to other applications.

Application Link Enabling

Application Link Enabling (ALE) provides SAP customers with a program distribution model and technology, which enables them to interconnect programs across various platforms and systems. ALE is available in R/3 Version 3.0 or later.

You can configure ALE to monitor events within SAP. As these events occur, ALE can send the event details (typically as an IDoc) to any configured output, such as a file, a database, socket, another SAP system or to a third-party system outside the SAP environment. You can also use ALE to configure transformations for the data content.

Advanced Business Application Programming

Advanced Business Application Programming (ABAP) is a programming language created by SAP. The syntax of ABAP is somewhat similar to COBOL.

Most of the application code of SAP is written in ABAP, and so any user extensions are typically also written using ABAP.

NetWeaver

NetWeaver is a suite of SAP products which are provided under a single license to an Enterprise. NetWeaver does not include R/3. The NetWeaver application server supports both ABAP and Java application code.

Exchange Infrastructure

Exchange Infrastructure (XI) is a SAP middleware offering. It is the SAP mechanism for enabling integration between SAP applications and third-party applications.

Business Application Programming Interface

ABAP is an object-oriented programming interface developed and used by SAP for its applications. For each application, SAP has defined a set of central Business Object Types (similar to Java class definitions) and for each of these types, a set of methods has been defined. This set of methods is called the Business Application Programming Interface (BAPI).

External access to Business Object data is possible only through the defined BAPI; for example, the functionality that is implemented with the SAP Business Object Type `Material` includes a check for the material's availability. Thus, the Business Object Type `Material` offers a BAPI called `Material.CheckAvailability`.

SAP provided the first BAPIs for customers and external providers in R/3 Release 3.1, thus enabling them to integrate their software components with SAP applications. The number of BAPIs increases with each SAP Release. BAPIs are implemented and stored in an R/3 system as remote function call (RFC) modules.

Remote Function Call

A Remote Function Call (RFC) is an application program interface to R/3 applications from SAP.

BAPIs are often discussed in terms of Remote Function Calls (RFCs); this is because BAPIs are implemented as RFC modules.

Work Flow

When an R/3 system is implemented for a customer, a SAP consultant tailors the functionality of the application to meet the customer's need; for example, the generic process of creating a new sales order can be tailored for your organization. The SAP consultant is able to link the required steps, in the correct order, and provide the user with the flow of work for creating a new sales order in their organization. This series of steps is called a work flow within SAP.

So when an SAP consultant sets up a new system for a customer, they set up all the work flows, tailored for that customer.

User Exit

Work flows have the ability to link to user-provided code. These links are referred to as user exits within SAP. This enables a consultant to add custom code into virtually any work flow of the SAP system. As an example, a consultant might provide a user exit that exports an order number to a file whenever a user creates a new sales order.

To embed a user exit in a code module, a consultant needs to know:

- the SAP code module that is to call the code for the user exit.
- the software access key to unlock the module, which is provided by the SAP administrator.

Custom user exit code is usually written in ABAP.

Basis Administrator

SAP Administrators are typically referred to as Basis Administrators. The term `Basis` is used to refer to the central core of SAP R/3; for example, a key to unlock SAP code modules, is part of the SAP Basis.

2 Installing the HPBPI Accelerator for SAP

This chapter describes the installation options for installing the HPBPI Accelerator for SAP.

The Accelerator installation comprises two parts:

- the adaptor Java code (`sapadaptor.jar`), which is integrated into the openadaptor framework
- the SAP transport, which is applied to the SAP System

Instructions for installing the Accelerator are described in the following sections:

- [Before Starting the Installation](#) on page 20
- [Installing XPL Viewer](#) on page 23
- [SAP Transport Files](#) on page 24
- [Installing the Accelerator](#) on page 26
- [Obtaining and Installing the SAP Libraries](#) on page 28
- [Setting up SAP User for the Accelerator](#) on page 31
- [Checking the Accelerator Installation](#) on page 33
- [Licensing and the HPBPI Accelerator for SAP](#) on page 35

Before Starting the Installation

This section describes the tasks that you must complete in order to install the HPBPI Accelerator adaptor component. These tasks must be completed before running the installation process.

You can install the adaptor on any supported Windows or HP-UX machine.

Windows

The following sections described the requirements that are specific to a Windows machine.

System Requirements

Table 1 lists the minimum system requirements for the HPBPI Accelerator installation on Windows. If you install the Accelerator on the same machine as the HPBPI Server, there are no additional system requirements as the HPBPI Server requirements are also adequate for the Accelerator.

Table 1 Minimum System Requirements for an HPBPI Accelerator Installation on Windows

Requirement	Value	How to Check
System Processor	Intel (at least 2Ghz)	Use: Programs > Accessories > System Tools > System Information
Windows version	<ul style="list-style-type: none">• Windows 2000 Server, plus SP4• Windows 2003 Server, plus SP1• Windows XP Professional, plus SP2	
Physical memory	1024MB	
Virtual memory	1536MB	

Disk Space Requirements

Table 2 describes the disk space requirements.

Table 2 HPBPI Accelerator for SAP Installation Disk Space Requirements on Windows

Installation Type	Free Disk Space	How to Check
HPBPI Accelerator for SAP	30MB	Use My Computer to check disk space. Click on the drive name where you intend to install HPBPI and check the available capacity.

HP-UX

The following sections described the requirements that are specific to an HP-UX machine.

System Requirements

Table 3 lists the minimum system requirements for the HPBPI Accelerator installation HP-UX.

Table 3 Minimum System Requirements for an HPBPI Accelerator Installation on HP-UX

Requirement	Value	How to Check
Hardware	Integrity Server	/usr/bin/uname -rs
HP-UX version	11.23	
Physical memory	64MB	
Virtual memory	256MB	

Disk Space Requirements

Table 4 describes the disk space requirements.

Table 4 HPBPI Accelerator for SAP Installation Disk Space Requirements on HP-UX

Installation Type	Free Disk Space	How to Check
HPBPI Accelerator for SAP	30MB	Verify the available disk space for your system using the following command: <code>/bin/df -k</code>

Windows and HP-UX Software Requirements

Table 5 describes the software and versions that are required to be installed before the installation is started.

Table 5 HPBPI Accelerator Software Requirements

Software Component	Version
J2SE	J2SE/JDK 5.0. This version of the J2SE can also be described as JDK version 1.5 and jdk1.5.0. You can download the correct version of the JDK from the following location: http://java.sun.com/products/archive/
SAP	SAP application components that are based on SAP version 4.6c and later, up to and including SAP Web AS ABAP 7.0.
SAP Java Connector (SAP JCo)	Release 2.1.8

Installing XPL Viewer

The HPBPI Accelerator for SAP generates trace files. These are binary files that are generated by XPL and that can be read using the HP BTO Software XPL trace viewer. XPL is installed automatically with the Accelerator, however, the XPL trace viewer is not installed. If you want to view the XPL trace files, the viewer is available for installation on your distribution media.

These trace files are intended to aid your support representative to isolate problems with the HPBPI system, including the Accelerator. The description of the use of the XPL viewer is beyond the scope of this document. However, when running the XPL Viewer to view and save trace files, the Accelerator is listed as the following XPL Application:

```
org.openadaptor.adaptor.RunAdaptor
```

SAP Transport Files

Before starting the Accelerator, you need to import, into SAP, the SAP transport files; these files contains the transport objects provided by the Accelerator.

Table 6 lists the naming convention for these transport objects.

Table 6 Naming Convention for Transport Objects

Transport Class	Transport Object	Description
Development class	/HPOV/BSM	The container of HPBPI objects, including the HPBPI Function group.
Function group	/HPOV/BPIADAPTOR	A list, or container, of logically related function modules and their components. In this case, the modules are HPBPI related.
Function modules	/HPOV/*	Procedures that enable you to encapsulate and reuse global functions within the SAP system.

The transport file contains Accelerator-specific information, which is required by the SAP system (in particular ABAP). If you do not import the content of the transport file into SAP, your SAP system and the Accelerator are not able to communicate.

The required Accelerator transport file is located in the following directory on the system where the Accelerator is installed:

```
acc-install-dir\SAPTransport\AcceleratorSAP.car
```

To install the transport files, you need to complete the following steps:

1. Copy the transport file from the Accelerator system to the SAP system that you intend to use.
2. Log in as the SAP administrator on the SAP system where the transport files are now located. You need to make sure that the SAP administrator is logged on with the correct SID. (The SAP administrator is identified as *SID*>adm, where *SID* is the instance identifier of the SAP database that holds the data for the IDocs that you intend to monitor.)

3. Move to the following directory on the SAP system:

```
/usr/sap/trans
```

4. Enter the following SAPCAR command to unpack the transport file:

```
SAPCAR -xvf AcceleratorSAP.car
```

You now need to import the extracted transport files into SAP.

5. Using the SAP GUI, locate the SAP Transport Management System (STMS) login page using the following transaction identifier:

```
STMS
```

6. From the SAP Transport Management System login page select the SAP system where you want to import the Accelerator transport file settings. The SAP system is identified by its SID.

7. Add the following in the `Transport-Request-to-import` queue:

- a. `K900776.SP1`

This is the main SAP transport, which includes the functions for the Accelerator.

- b. `K900777.SP1`

This is the user role transport for the Accelerator, which is used within the SAP system.

- c. the target client number

000 is the default client number

These transports can also be found in the following file

```
acc-install-dir\SAPTransport\readme.txt
```

8. Select the option to import the transport and select the options that are appropriate for your system in the import dialog.

Installing the Accelerator

The following sections describes how to install the Accelerator on a Windows or Unix system:

- [Installation on Windows](#) on page 26
- [Installation on Unix](#) on page 27

Installation on Windows

This section describes the steps for installing the HPBPI Accelerator for SAP on a Windows machine. Before starting the installation, make sure that the system environment variable for `JAVA_HOME` is set.

Complete the steps described in the following sections to install the Accelerator:

1. From an account with Administration privileges, run the installation procedure, which is located on the distribution media at the following location:

```
cd-drive\i386\HPBpiSAP-02.20.000-WinNT4.0-release.msi
```

You are presented with an introduction to the installation. Click `Next` to start the installation.

Alternatively, you can follow the link from the `readme` file presented when you insert the CD into the CD drive.

2. Complete the `License Agreement` dialog, and click `Next`, or `Cancel` as appropriate.
3. From the `Destination Folder` dialog, accept the default location offered for the product files, or navigate to a directory of your choice.

You can ignore the question for `Data Files` as the Accelerator does not have any `Data Files`. Click `Next`.

4. From the `Ready to Install the Program` dialog, click `Install` to install the Accelerator, or click `Back` if you want to change any of your settings.

The installation procedure continues and installs the Accelerator directories and files. When the installation is complete, click `Finish` to exit the installation procedure.

Installation on Unix

This section describes the steps for installing the HPBPI Accelerator for SAP on HP-UX. Before starting the installation, make sure that system environment variable for `JAVA_HOME` is set.

Complete the following steps to install the Accelerator on HP-UX:

1. Log on with system administration (`root`) capability.
2. Make sure that you have access to the depot server where the Accelerator is located.
3. Locate the depot on your distribution media, for example, on a CD-ROM:

```
cd-image/hp-ux/  
HPBpiSAP-02.20.000-HPUX11.22_IPF32-release.depot.gz
```

where *cd-image* is the mount point for the CD-ROM.

4. unzip the depot image to a temporary directory, for example:

```
gunzip HPBpiSAP-02.20.000-HPUX11.22_IPF32-release.depot.gz
```
5. Type `swinstall`
6. Click HP Business Process Insight Accelerator for SAP Applications
7. Select Actions | Mark forInstall.
8. Select Actions | Install (analysis) to start the installation.

This may take a few minutes.

The installation script checks that your system meets the installation requirements. When it has finished the check, select `OK` to begin the installation.

9. When the installation is complete, click `Done`, and exit from the installer.

Obtaining and Installing the SAP Libraries

When you have completed the Accelerator installation, and before you start the Accelerator, you need to obtain a copy of the required SAP libraries from the SAP Service Marketplace. These file are then copied to specific locations on the machine where you have installed the Accelerator.

The SAP Service Marketplace contains a variety of internet portals one of which is a connectors portal.

Before you can connect to the SAP Service Marketplace, you need to make sure that you have the appropriate account details. These are supplied when you purchase a SAP product or service.

The required libraries can be obtained from the SAP Service Marketplace, which is accessible from the following URL:

<http://service.sap.com/connectors>

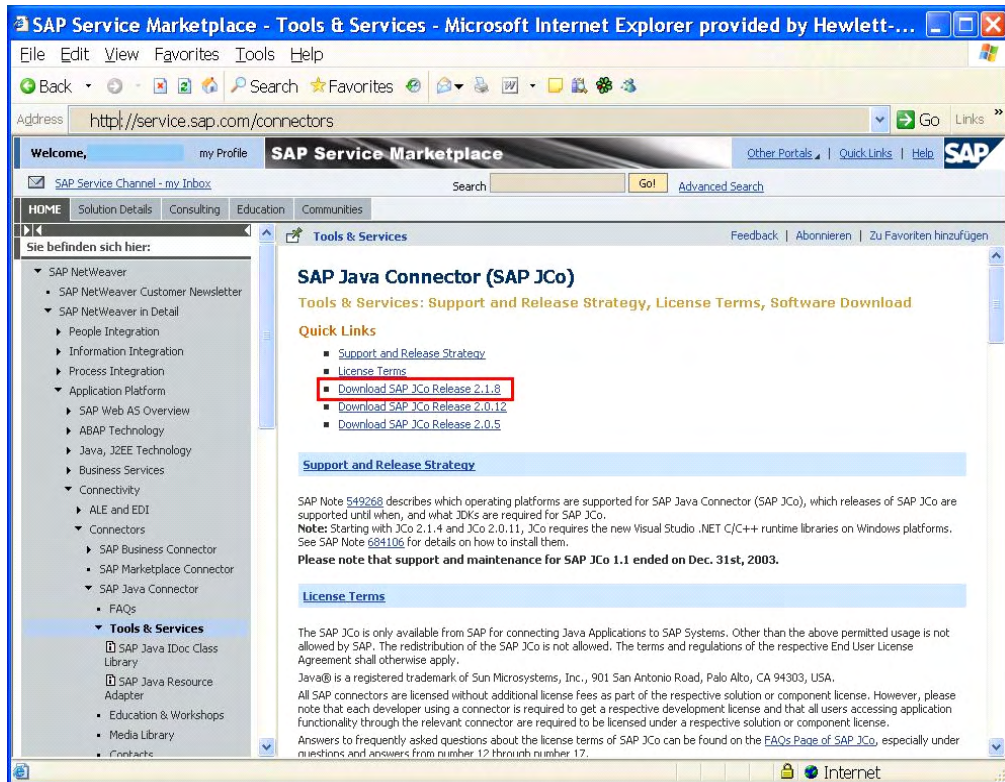
The required libraries are part of the SAP Java Connector (SAP JCo) Release 2.1.8. To the obtain the libraries, you need to locate the download area for the correct version of SAP JCo for your platform.

You can navigate to the correct location using the left-hand navigation pane. The navigation pane operates as a hierarchical view and you need to use the following menu options to open the page that you need:

Connectors > SAP Java Connector > Tools and Services

Figure 2 shows an example of the menu options and the download page.

Figure 2 SAP Marketplace Example



When you have selected Tools and Services, you are taken to a download page where SAP JCo Release 2.1.8 is listed. Follow the link to this release and you are then offered a list of platforms, and you need to download the files appropriate to your platform.

When you have downloaded the correct SAP JCo package for Release 2.1.8, you need to extract the following files into a temporary directory on the machine where you intend to install the Accelerator:

- For Windows:
 - sapjco.jar
 - librfc32.dll
 - sapjcorfc.dll
- For HP-UX:
 - sapjco.jar
 - librfccm.so
 - libsapjcorfc.so

The SAP JCo release package contains many files, including documentation and training modules. You need only download those files listed in order to use the Adaptor. In addition, you need to make sure that you use a set of files from the same version of the SAP JCo release, do not mix files from different versions.

Copy the extracted files to their target location as follows:

- Copy `sapjco.jar` to `acc-install-dir\nonOV\java`
- For a Windows installation, copy `librfc32.dll` and `sapjcorfc.dll` to the following Windows directory:
`Windows\system32`
- For an HP-UX installation, copy `librfccm.so` and `libsapjcorfc.so` to the following HP-UX directory:
`acc-install-dir/nonOV/lib`

Setting up SAP User for the Accelerator

The Accelerator needs a User profile to be configured within SAP in order that SAP and the Accelerator can exchange information. Some of the details of this user are defined in the transport file that you imported in section [SAP Transport Files](#) on page 24. You cannot create a SAP User for the Accelerator until the appropriate transport file has been imported into the SAP system that you intend to integrate with.

You need to create a user account for each instance identifier (SID) for the SAP database that holds the data for the IDocs that you intend to monitor. If there is more than one client for the SAP system, then you need a user account for each client. Following the naming guidelines in place for your SAP system.

For each SAP SID, complete the following steps:

1. Log on to the SAP System from an account that has sufficient privileges to execute SU01.
2. Enter the transaction identifier SU01.

Alternatively, you can navigate to the SU01 user in the SAP menu tree.

3. From the SU01 account, you need to create new user with a name that identifies the Accelerator; for example: accsap.

When creating this new user you need to:

- a. add a password for the user. This is a temporary password and you are asked to change this password the first time you log into the Accelerator user account.
- b. define a role for the Accelerator. Use the following function group, which you imported as part of the transport file, and which defines the user role for the Accelerator:

```
/HPOV/BIAADAPTOR
```

This function group contains the details of the transactions that the Accelerator is able to execute within the SAP environment, plus information such as the data that the Accelerator can access.

The user role is client-specific must be imported into the same client as you imported the transport file for the Accelerator. This is also the same client that you specify when you create the properties file for the Accelerator in [Chapter 3, Configuring the HPBPI Accelerator for SAP](#).

4. Activate the user profile and assign it to the SAP user that you created in step 3.

Make sure that all the user authorization objects are activated when you activate the user profile.

5. Log onto the SAP system using the new user account that you have created and using the password you set in step 3a.

You are prompted to change the password.

6. Change the password as requested. You must use this new password when you create the adaptor properties file as described in [Chapter 3, Configuring the HPBPI Accelerator for SAP](#).

You have completed the setup for the SAP user.

Checking the Accelerator Installation

The Accelerator installation provides a Checker script that checks the integrity of your system and that all the required files are in place.

The Checker script is located as follows:

`acc-install-dir\bin\Checker.bat` (Windows)

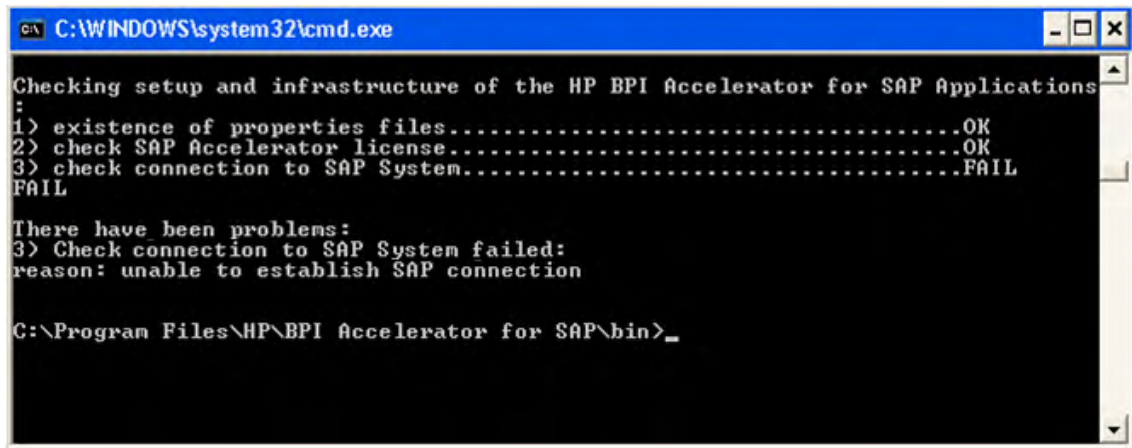
`acc-install-dir/bin/Checker.sh` (HP-UX)

You can run the script from a Windows Command prompt or as an HP-UX shell script.

The Checker script checks the following:

- the adaptor property file exists and is in the expected location.
- that the Accelerator has a suitable license.
- the adaptor is configured to be able to contact an SAP system. To do this, it checks that the required SAP libraries are in place.

The following is an example of the result of running the script.



```
C:\WINDOWS\system32\cmd.exe
Checking setup and infrastructure of the HP BPI Accelerator for SAP Applications
:
1) existence of properties files.....OK
2) check SAP Accelerator license.....OK
3) check connection to SAP System.....FAIL
FAIL
There have been problems:
3) Check connection to SAP System failed:
reason: unable to establish SAP connection

C:\Program Files\HP\BPI Accelerator for SAP\bin>_
```

You can also use the Checker script with your customized properties file. The following is an example of the using the Checker script on Windows:

```
acc-install-dir\bin\Checker adaptor-name  
adaptor-properties-file-name
```

where:

- *adaptor-name* is the name of your adaptor
- *adaptor-properties-file-name* is the name of your properties file

Licensing and the HPBPI Accelerator for SAP

You must have a license key password to use the HPBPI Accelerator for SAP. Following your initial installation, you are automatically assigned a 60-day trial license. This license covers both the HPBPI Server and the HPBPI Accelerator. Within this 60 day period, you must obtain either a trial license extension, or a permanent license key password for the HPBPI Accelerator.

The Accelerator uses a Node locked license based on the short hostname (not the fully-qualified hostname) of the machine where the HPBPI Server that the Accelerator connects to is installed.

You must supply a short hostname for the machine where you intend to run HPBPI when completing your license request form. You can obtain the short hostname using the following command from a Windows Command Prompt:

```
hostname
```



Do not supply an IP Address when applying for your license key as you will not be able to use the generated password to unlock your HPBPI Accelerator installation.

The following is an example of a short host name:

```
server1
```

The following is an example of a fully-qualified hostname:

```
server1.hp.com
```

The following is an example of an IP Address:

```
11.22.33.44
```

Trial License Extension

If you need to request an trial (or evaluation) license extension, access the HP BTO Software license and password delivery web site and navigate to the page for evaluation extension licenses.

<http://www.webware.hp.com>

Be aware that evaluation license extensions are granted only once, after which time a permanent license must be purchased. Full details of the process can be found on the Web site.

Permanent License Key Password

If you have purchased the HPBPI Accelerator software and received your entitlement certificate, you need to enter your permanent license key password into the licensing configuration software. The licensing configuration utility is installed as part of the HPBPI server as an option on the HPBPI Administration Console. Refer to the *HPBPI Installation Guide* for details of how to access the licensing configuration utility.

HPBPI Files Duplicated Under the Accelerator Installation Directory

There are some HPBPI files, which are installed as part of the Accelerator installation and are located as follows:

- `acc-install-dir\java\bia-common.jar`
- `acc-install-dir\java\bia-event.jar`

You need to make sure that you are aware of any updates and patches applied to these HPBPI files, which might also impact the Accelerator.

There is a readme file supplied with each HPBPI patch and which contains details of the components impacted by the patch. You need to read this file to determine if the patch impacts the Accelerator.

Removing the Accelerator for SAP

The following sections describe how to remove your Accelerator software from a Windows or an HP-UX machine:

- [Removing the Accelerator from a Windows Machine](#) on page 38
- [Removing the Accelerator from an HP-UX Machine](#) on page 39

Removing the Accelerator from a Windows Machine

To remove the Accelerator from a Windows machine, complete the following steps:

1. Make sure the Accelerator is not being used and is not accepting connections from HPBPI or SAP.
2. Select `Start | Settings | Control Panel | Add/Remove Programs`
3. Select `HP Business Process Insight Accelerator for SAP` from the currently installed programs.
4. Click `Change/Remove` to start the uninstall program.

You are presented with an introduction screen labeled `Uninstall HP Business Process Insight Accelerator for SAP`.

5. Click `Next` to continue.
6. The uninstaller continues and removes the Accelerator.

The uninstaller reports any files that it cannot delete as part of the uninstallation process.

Click `Done` to complete the uninstallation procedure.

You can also remove the Accelerator by running the installation program again. Re-running the installation program provides options for you to modify, repair or remove the HP Business Process Insight Accelerator for SAP. Select the `remove` option to remove the Accelerator.

Removing the Accelerator from an HP-UX Machine

Follow the steps below to remove the Accelerator from an HP-UX system.

1. Make sure that the Accelerator and any other applications the Accelerator are shut down before you remove the Accelerator software.
2. Use `/usr/sbin/swremove` to remove the Accelerator and its configuration information.

The remove program reports any errors that it encounters and informs you when the Accelerator has been removed from the system.

Re-Installing the Accelerator

You can reinstall the same version of the Accelerator to repair or modify a system. To do this, run the installation program again and you are presented with the options to Modify, Repair or Remove. Select the required option and follow the instructions presented by the installation program.

You can also remove the Accelerator installation as described in section [Removing the Accelerator for SAP](#) on page 38 and section [Installing the Accelerator](#) on page 26.

3 Configuring the HPBPI Accelerator for SAP

This chapter describes how to configure HPBPI and the adaptor component of the Accelerator in order that IDoc status changes can be converted into HPBPI Business events and passed to HPBPI. These business events can then be monitored by the HPBPI system.

The chapter is structured as follows:

- [The Adaptor and Business Events](#) on page 42
- [Configuring the Adaptor](#) on page 45
- [Starting the Adaptor on Windows](#) on page 56
- [Starting the Adaptor on HP-UX](#) on page 59
- [Logging](#) on page 62
- [Configuring HPBPI to Monitor SAP Events](#) on page 63
- [Example - Configuring Idoc Adaptors](#) on page 64
- [Example - Configuring Name Mapping](#) on page 75

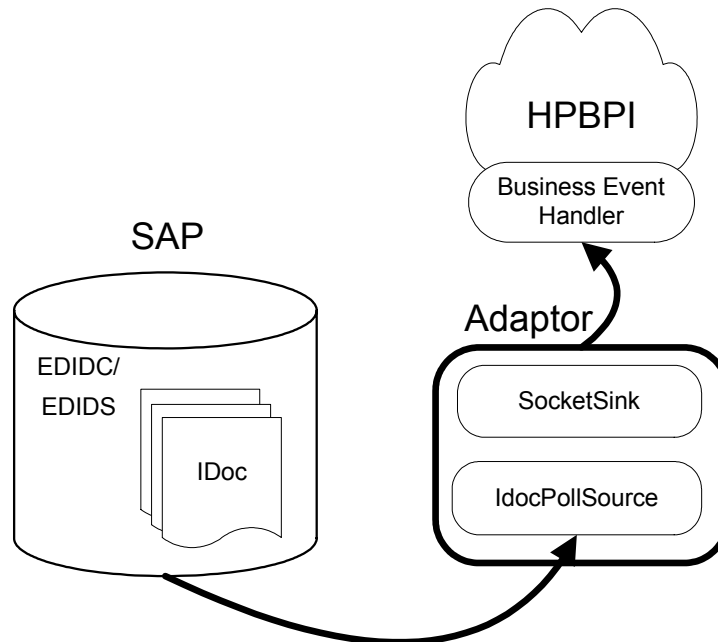
The Adaptor and Business Events

The adaptor comprises two parts. One part is an ABAP interface that communicates with the SAP system to obtain the IDoc header information. ABAP (Advanced Business Application Programming) is the programming language that SAP designed and that is used by most of the SAP applications. The second part is the Java interface that sends the IDoc header information to HPBPI as Business events

The adaptor obtains information about the status of the IDocs by combining data that it receives from the EDIDS and EDIDC database tables. The adaptor then polls the database tables at intervals defined by a polling period that you can configure in the adaptor properties file. The structure of this properties file is described in section [Adaptor Properties File](#) on page 45.

This properties file contains the configuration details for each instance of the adaptor, including the details of the adaptor's source and sink.

Figure 3 Adaptor Source and Sink Architecture



Adaptor as Data Source

The adaptor is a Java application and is a source component in openadaptor terminology. The adaptor enables you to specify the IDoc status changes that you want to monitor and how you want these status changes to be raised as Business events within HPBPI.

The adaptor calls the following function module, within the SAP environment, to collect the IDoc information from the EDIDS and EDIDC database tables:

```
/HPOV/IDOC_GET_STATUS
```

This function module collects the appropriate IDoc information from the SAP database tables and returns this to the adaptor. You configure property files to specify the data that the adaptor collects. The adaptor then sends the IDoc data to the Business Event Handler for processing.

The EDIDS and EDIDC are used as follows.

- EDIDC

This database table contains the IDoc control record, for example, details of the IDoc message type and Receiver Port. The information in the IDoc control record can be used to uniquely identify the IDoc as it is passed between SAP applications. You use the data from the control record when creating a unique Data definition within your flow for the IDoc status document that you want to monitor.

- EDIDS

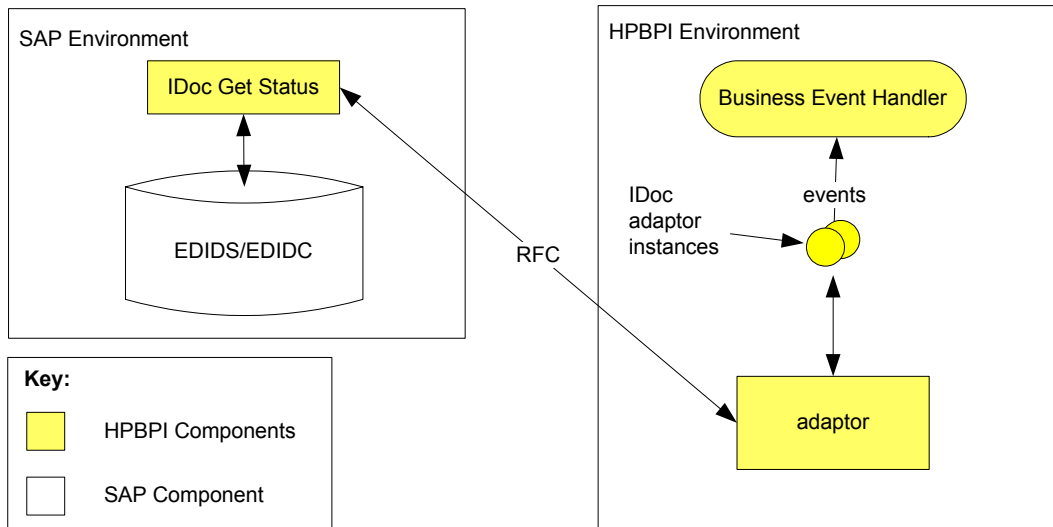
This database table contains the history of the status of the IDoc and is used as the progression data for your HPBPI business flow.

The adaptor searches the EDIDS table for a list of the status changes for each IDoc that has changed.

The function module `IDOC_GET_STATUS` first checks the EDIDC table to determine what has changed. The function module then reads each IDoc, based on the document number, and uses the data in the database table EDIDS to obtain the history status information since the function was last executed. The function module records a timestamp for the last time it accessed the database and next time the function module runs it records the changes to the IDoc since the last timestamp.

Figure 4 shows the architecture of the adaptor.

Figure 4 Adaptor Architecture



The adaptor uses the SAP Java Connector (SAP JCo.) to communicate with the SAP system; SAP JCo. is available from the SAP Service Marketplace (you need to hold credentials to identify yourself in order to access the SAP Service Marketplace). In addition, SAP JCo has a dependency on platform-specific libraries, which you also need to obtain. The installation instructions for the HPBPI Accelerator describe these libraries, how to obtain them and where to copy them. Refer to [Chapter 2, Installing the HPBPI Accelerator for SAP](#) for full details of installing the Accelerator.

Configuring the Adaptor

This section explains how you use the adaptor to obtain the information that you need from the IDoc in order to monitor your business processes. This section assumes that the HPBPI Accelerator for SAP is already installed on the system or systems where you want to run the adaptor instances.

The outline of the steps you need to complete to configure the adaptor is:

1. [Identifying IDocs to Monitor](#) on page 45
2. [Adaptor Properties File](#) on page 45
3. [Password Encoding Utility](#) on page 55

Identifying IDocs to Monitor

You need to identify the specific IDocs within SAP that you want to monitor. When you have done this, you configure the adaptor to collect these IDocs.

The adaptor collects the IDocs based on the content of the status query property that you specify when you configure the adaptor. This property is described in section [Status Query Option](#) on page 52. If you do not specify a query string in the query field, the adaptor instance collects all IDocs.

Adaptor Properties File

You configure the Accelerator adaptor as you do any other openadaptor adaptor, by creating a properties file. You need to create one properties file for each instance of the adaptor that you want to run.

You might have a number of adaptor instances collecting information from IDocs, according to the value of the query property. You also have different instances for different business events within HPBPI. Refer to the *Business Process Insight Integration Training Guide - Business Events* for details of the structure of the openadaptor properties file. This section assumes that you understand the purpose of the properties file.

The adaptor properties file contains the details of the configuration options for the adaptor as follows:

- it identifies the Java classes for the source and the sink components of the adaptor.
- it contains details of how to connect to the appropriate SAP system and the information that you want to filter from the IDoc within the defined SAP system.

When configuring the data that you want to collect from the IDoc header, be aware of the following: the data item names in the property files either have to exactly match the Data definitions you configured in the HPBPI Modeler when you designed the flow, or you need to provide a mapping, or transformation, pipe to map the data obtained from SAP to the Data definitions within the Modeler; see [Example - Configuring Name Mapping](#) on page 75 for an example of using a transformation pipe.

The following sections describe the options available for you to configure in the properties file for the adaptor. Note that in the description of the property names *accelerator-name* is the name that you have given the adaptor within the properties file.

Adaptor Source Class Name

You need to specify the Java classname for the SAP Idoc Source by setting the following property:

- `ClassName`

The full Java class name for the adaptor source component. This property value must be set to:

```
com.hp.ov.sap.bia.idocadaptor.IdocPollSource
```

Polling Properties

You need to configure the adaptor to operate in a polling mode by setting the following properties:

- `PollSource`

When set to `true`, the adaptor operates in a polling mode, where it checks for updates to the IDocs at intervals defined by a polling period set in `PollPeriod`.

In polling mode, the adaptor opens a connection to the SAP database and processes all the available IDocs. When all these IDocs are processed, the adaptor closes the database connection and waits for the period of time specified by the property `PollPeriod` and then opens a connection to the SAP database to process the next batch of IDocs.

If you set `PollSource` to `false`, the adaptor opens a connection to the SAP database, processes all the available IDocs, closes the connection and then terminates.

If this property is not included in the properties file, a value of `true` is assumed.

- `PollPeriod`

The length of time that the adaptor waits before checking for the next set of IDocs. This period is specified in milliseconds.

If this property is not included in the properties file, a value of 60000 is assumed.

Test Mode

You can run the adaptor in test mode by setting the following property:

- `InhibitHistory`

Setting `InhibitHistory` to `true` enables you to use the adaptor in a testing mode. When running in this testing mode, the adaptor retrieves the current set of IDocs, process them and then terminates without marking these IDocs as having been processed. This enables you to test your adaptor using any current IDocs without processing these IDocs.

You can also test your settings for attributes and mappings and when you are satisfied that your adaptor is correctly configured, you can set `InhibitHistory` to `false` and rerun your adaptor.

If this property is not included in the properties file, a value of `false` is assumed.

Licensing

These properties specify the location of the machine where the HPBPI Server system is installed that has the appropriate HPBPI Accelerator for SAP license registered. Refer to section [Licensing and the HPBPI Accelerator for SAP](#) on page 35 for details of licensing and the Accelerator.

- `BPI.serverhostname`

The hostname (full domain name) of the machine where the HPBPI Server managing the HPBPI Accelerator for SAP licensing is installed.

This property defaults to `localhost`.

- `BPI.rmiregistryport`

The RMI port number used by the HPBPI Server installed on the machine identified by `BPI.serverhostname`. As installed, this port number is 44000. The actual value being used by the HPBPI Server is shown in the Administration Console, under Port Numbers (RMI Registry).

If this property is not included in the properties file, a value of 44000 is assumed.

SAP Connection Options

The following properties configure the SAP connection. Values for these properties must not be omitted from the properties file:

- `SAP.hostname`
This is the fully qualified hostname for the SAP system where the IDoc is updated with the data that you need.
- `SAP.sid`
The instance identifier of the SAP database that holds the required data.
- `SAP.client`
The SAP client identifier, which is a self-contained unit with separate master records and its own set of tables. The user could represent a business or a function within a business, according to the structure of the SAP data.
- `SAP.systemnumber`
The system number of the SAP system and is a two-digit numeric value.
- `SAP.user`
The name of the SAP user that the adaptor uses to access the database.
- `SAP.password`
The password for the SAP user that the adaptor uses to access the database. There is a utility that you can use to encode the password described in section [Password Encoding Utility](#) on page 55.
- `SAP.language`
The two-character country code for the language that SAP system is configured for.

The following properties are optional:

- `SAP.timezone`
The timezone of your SAP database. This property is required if your SAP database is running in a timezone that is different from the HPBPI server that you are sending the event to.

If you do not specify a `timezone` property then the adaptor passes the SAP database timestamps through to HPBPI without adding any timezone information.

As an example, if you want to configure the adaptor to process all dates within the SAP database as London local time, set the `timezone` property as follows:

```
IdocAdaptor.C1.SAP.timezone = @Europe/London
```

- `SAP.gatewayhost`

This might be required when the SAP configuration uses a gateway, on a central host, to allow access to a number of SAP server systems. This property is the hostname of the system where the SAP gateway is running.

- `SAP.gatewayservice`

Specifies the name of the SAP gateway service that is used by the adaptor to monitor the IDocs.

SAP Connection Tracing

The following tracing options relate to the levels of SAPRFC tracing that you want to set. These tracing options enable you to trace connections and queries within the SAP system:

- `Tracing.SAPRFC.tracelevel`

Level of tracing require, where:

- 0 = no trace files are generated
- 1 = logging for the SAP JCo Java API is generated
- 3 = logging for the SAP JCo JNI API is generated
- 5 = error diagnostic data is generated

You are recommended to set the trace level to 5 when trying to identify problems with the adaptor.

If this property is not included in the properties file, a value of 0 is assumed.

- `Tracing.SAPRFC.tracepath`

The directory where you want the trace file(s) to be written.

- `Tracing.SAP.queryconditions.enable`

Set this to `True` if you want to create a query table, set this to `False` if you do not want to create a query table.

If this property is not included in the properties file, a value of `false` is assumed

- `Tracing.SAP.querytable`

This is the name of the html file that contains the details of the query that is being processed within the SAP system for this adaptor instance. The file is written in the directory specified by the `Tracing.SAPRfc.tracepath` property.

Update Time Interval

This option relates to the time interval used by the adaptor to check for updates to the IDoc in SAP database.

- `IdocAdaptor.deltatime`

Each time the adaptor checks for updates, it uses the time that was recorded for the start of the `PollPeriod` as its start time for identifying new IDocs to process. In addition, the adaptor adjusts this time by the value given for `deltatime`, to create an overlap period. This then removes the possibility of IDocs being missed.

If this property is not included in the properties file, a value of 30 seconds is assumed.

Processing Type

Processing type is used to map the IDoc number between SAP systems. This is because when an IDoc is transferred from one SAP system to another SAP system its IDoc number changes.

Each IDoc has a unique number (`DOCNUM`); however, the SAP system sending the IDoc to another SAP system does not know the IDoc number used by the SAP system receiving the IDoc. As a result, there needs to be a way of maintaining a unique number for the same IDoc across two SAP systems in order that HPBPI has a unique identifier that it can use to monitor the progress of the IDoc across systems.

If, in your implementation, IDocs are transferred between SAP applications, you need to set `Idoc.processing.type` to the value of `inbound`. In this case, the adaptor creates a new field called `CURRENTDOCNUM` in which it maintains the unique value for `DOCNUM` and maps `CURRENTDOCNUM` onto `DOCNUM` in order to maintain a unique identifier that can be used for the HPBPI business flow monitoring the IDoc.

If, in your implementation, IDocs are transferred between SAP and third-party applications, set `Idoc.processing.type` to the value of `outbound`.

The following property can be set for processing type:

- `Idoc.processing.type`

This can be set to either `outbound` or `inbound`

If this property is omitted from the properties file, a value of `outbound` is assumed.

Status Query Option

If you do not enter a query statement, all IDocs are collected; however, collecting all IDocs is not recommended for performance reasons. You are advised to have multiple adaptors and split the queries between the adaptors. You are able to specify the `where` clause of the select statement used to retrieve the IDocs from the SAP system.

The following are examples of a `where` clause:

```
RCVPRN EQ 'WA9CLT111'
```

```
DOCTYPE = 'MATMAS' AND RCVPRN like 'PROD%'
```

SAP uses Open SQL (a set of ABAP statements) to complete operations on its database. Open SQL thus provides a common syntax for all of the database systems supported by SAP. Open SQL grammar works only with database tables that have been created in the ABAP Dictionary.

The Open SQL grammar is defined in the SAP online help and is also available from the SAP knowledge warehouse at the following Web site:

http://help.sap.com/saphelp_nw04/helpdata/en/e0/4cc71fd0bb40d78c2a7cd6aa6a8b59/frameset.htm

You can also refer to the SAP documentation for details of the Open SQL grammar.

Use the following property to set the where clause of the Open SQL query for this IDoc:

- `Idoc.query`

The where clause of the Open SQL statement that you want to use to select the appropriate IDocs.

Within this `Idoc.query` property you can use all the attributes from the EDIDC table except for the `STATUS` attribute.

Business Event Fields

These are the fields from the IDoc header (SAP table EDIDC) that you want to monitor and therefore define as properties of the Event definition in the HPBPI Modeler.

The names of these properties must match the property names in the Modeler, or you must use a transformation pipe to transform the property names within the adaptor. Refer to the [Example - Configuring Name Mapping](#) on page 75 for an example of using a transformation pipe.

In addition to these fields, the adaptor also has access to the IDoc number of the current SAP System using the field `CURRENTDOCNUM`.

You can specify as many fields as you need and each field name must be numbered sequentially, starting with the number one (1) as follows:

- `Idoc.event.field1`
- `Idoc.event.field2`
- `Idoc.event.field3`
- `Idoc.event.field4`
- `Idoc.event.field5`
- ...
- `Idoc.event.fieldn`

The possible field values that can be specified are listed in [Appendix A, SAP EDIDC Table Fields](#).

Business Event Name

The following property sets the event name that is sent into HPBPI:

- `IdocAdaptor.eventname`

The full name of the Event definition for the HPBPI business flow that is monitoring this IDoc, as defined in the HPBPI Modeler.

If this property is not included in the properties file, a value of `SAP/IdocStatus` is assumed.

This property value must match the value defined in the HPBPI Modeler.

Adaptor Sink Class Name

The sink component is a component taken from the openadaptor framework:

```
org.openadaptor.adaptor.standard.SocketSink
```

The following are the property names for the sink component:

- `ClassName`

The Java class name for the sink component. This property value can be set to one of:

- `org.openadaptor.adaptor.standard.SocketSink`
- `BIAEventStoreSink`

For information about the `BIAEventStoreSink`, see the *Business Process Insight Integration Training Guide - Business Events*.

For the `org.openadaptor.adaptor.standard.SocketSink`, the following are additional properties required to identify the location of the sink component:

- `HostName`

The host name of the machine where the HPBPI Server is installed.

- `Port`

The port number for the Business Event Handler Source as shown in through the Administration Console on the machine where the HPBPI Server is installed. Unless modified this port number is 44005.

Password Encoding Utility

This section explains how to use the password encoding utility supplied with the adaptor to encode the value of the password that you specify in the properties file.

The encoding utility is based on the encoder supplied with openadaptor with a user-friendly interface for generating the password.

Complete the following steps to use the interface for the utility. The steps describe the utility on a Windows system, and are identical for an HP-UX system, with the exception of the file name extension (`encodePassword.sh`) and directory:

1. Move to the following HPBPI Accelerator for SAP installation directory where the encoder utility is installed:

```
acc-install-dir\bin
```

2. Run the utility as follows:

```
encodePassword path_name\IdocPollSource.props
```

where *path_name* is the directory specification for the location of your adaptor properties file.

3. Enter 1 to change a password property that you specify, enter 2 if you want encode password to search for and prompt you for all the passwords that it finds, enter 3 to exit without making any changes.
4. If you enter 1, `encodePassword` requests the name of the password property that you want to change.
 - a. Enter the full name of the property, for example, `IdocAdaptor.C1.SAP.password`, and press Enter.
 - b. Enter the password that you want to set.
 - c. Confirm the password and the utility exits
5. If you enter 2, `encodePassword` searches for all properties that end with the token `.password`. For each password property that the utility finds:
 - a. Enter the password that you want to set.
 - b. Confirm the password and the utility exits
6. If you enter 3, the utility exits.

Starting the Adaptor on Windows

You can start the adaptor in one of the following ways:

- If you are running the adaptor on a machine where HPBPI is installed, you can use the `biaeventadaptor.bat` script as described in the *HPBPI Integration Training Guide - Business Events*. On startup, this script writes log information to the log directory under the HPBPI log directory.
- If you are running the adaptor on a machine that does not have HPBPI installed, you need to use the `openadaptor runAdaptor` class and provide the name of your adaptor property file and the name of the adaptor; refer to the *HPBPI Integration Training Guide - Business Events* for information about starting the adaptor using `runAdaptor`. On startup, the `runAdaptor` class writes log information to `stdout` or `stderr`.
- As a Windows Service

Using a Java Wrapper Service, you can create a Windows Service that you can use to start and stop instances of the adaptor on the system where it is installed. The Java Wrapper Service and how to configure it is described in the *HPBPI Integration Training Guide - Business Events*.

There is a Java wrapper configuration file for the adaptor installed in the following location:

```
acc-install-dir\data\conf\IdocAdaptorWrapper.conf
```

This file is a template and you need to modify the properties in this file for your specific instance, or instances, of the adaptor. Section [Java Service Wrapper Configuration](#) on page 57 describes the properties in this file that are specific to the adaptor.

Java Service Wrapper Configuration

Before starting the adaptor as a Windows Service, you need to edit the Java Service Wrapper properties file and set the environment variables for your specific instance of the adaptor. The following is an extract from the file listing the environment variables following a new installation:

```
set.IDOC_NON_JAVA=C:/Program Files/HP/BPI Accelerator for SAP/  
nonOV/java  
set.IDOC_JAVA_DIR=C:/Program Files/HP/BPI Accelerator for SAP/  
java  
set.WRAPPER_JAVA_DIR=C:/Program Files/HP/BPI Accelerator for  
SAP/nonOV/java  
set.WRAPPER_LIB_DIR=C:/Program Files/HP/BPI Accelerator for SAP/  
nonOV/lib  
set.IDOC_PROPS_FILE="C:/Program Files/HP/BPI Accelerator for  
SAP/data/conf/IdocPollSource.props"  
set.IDOC_DATA_DIR="C:/Program Files/HP/BPI Accelerator for SAP/  
data"  
set.WRAPPER_LOG_FILE=C:/Program Files/HP/BPI Accelerator for  
SAP/data/log/OVIdocAdaptorWrapperSs.log  
set.WRAPPER_SERVICE_TITLE="BPIIdocAdaptor"  
set.NT_SRV_NAME=BPIIdocAdaptor  
set.NT_SRV_DISPLAY_NAME=BPIIdocAdaptor
```

where:

- IDOC_NON_JAVA
The location of the third-party Java application files used by the adaptor.
- IDOC_JAVA_DIR
The location of the adaptor and HPBPI Java application files.
- WRAPPER_JAVA_DIR
The location of the Java Service Wrapper .jar file.
- WRAPPER_LIB_DIR
The location of the dll for the Java Service Wrapper.
- IDOC_PROPS_FILE
The location of the adaptor properties file.
- IDOC_DATA_DIR
The location where the adaptor history information is maintained.

- WRAPPER_LOG_FILE

The filename and location for the generated log files.

- WRAPPER_SERVICE_TITLE

The name used in the title for the Wrapper console.

- NT_SRV_NAME

Internal name for the Windows Service

- NT_SRV_DISPLAY_NAME

Display name for the Windows Service as shown in the Services dialog.

Modify these settings for your instance of the adaptor and then start the adaptor and install a service for the adaptor, using a command similar to the following:

```
acc-install-dir\nonOV\bin\wrapper.exe -i  
                  acc-install-dir\data\conf\IdocAdaptorWrapper.conf
```

Starting the Adaptor on HP-UX

If you have installed the adaptor on Unix machine, use the following shell script to start it:

```
/opt/HP/BPI_Accelerator_for_SAP/bin/BPIIdocAdaptor.sh
```

Make sure you have changed to the directory where the shell script is located and enter the following command:

```
./BPIIdocAdaptor.sh start
```

This starts the adaptor as a background daemon process.

If you are having problems starting the adaptor, you can modify the script and configure it to show the configuration parameters that it is using when starting. The following is an extract from the script `BPIIdocAdaptor.sh`:

```
start')
# Start the adaptor in the background
echo "Starting ${ADAPTOR_DESC}..."
{
    while [ true ]
    do

        nohup $JAVA_EXE -classpath "$CP" $ARGS $START_CLASS $PROPS_FILE
            $ADAPTOR_NAME >/dev/null 2>&1

#    $JAVA_EXE -classpath "$CP" $ARGS $START_CLASS $PROPS_FILE $ADAPTOR_NAME
```

You need to add a comment character to the line in the shell script that starts:

```
nohup....
```

You then need to remove the comment character from the line that starts:

```
#    $JAVA_EXE
```

When you have solved the problem, you need to return the script to its original configuration.

Remote Control

When starting and stopping your adaptor on HP-UX, you also need to make sure that you configure the adaptor and enable remote control within its property file. If you do not, you cannot stop the adaptor when you have started it.

The remote control property settings are as follows in the properties file:

```
#IdocAdaptor.Controller.RemoteControl.ClassName =  
org.openadaptor.adaptor.standard.HTTPRemoteControl
```

```
#IdocAdaptor.Controller.RemoteControl.HTTPPort = 29292
```

You need to remove the comment characters on these lines to enable remote control for your adaptor.

Stopping the Adaptor on HP-UX

If you have installed the adaptor on Unix machine, use the following shell script to stop it:

```
/opt/HP/BPI_Accelerator_for_SAP/bin/BPIIdocAdaptor.sh
```

Make sure you have changed to the directory where the shell script is located and enter the following command:

```
./BPIIdocAdaptor.sh stop
```

Logging

The adaptor provides the following logging options:

- Standard openadaptor logging that you set in the adaptor properties file. This logging information is written to `stderr`. You can also set the logging levels from within the properties file by setting the property:

```
IdocAdaptor.Logging.LogSetting1
```

You can set this property to the values: INFO, DEBUG, WARN or FATAL

- Service wrapper logging, which is useful if you are starting the adaptor using the Windows Services. This logging information is written to the file location that you configure within the Wrapper properties file, for example:

```
acc-install-dir\data\log\OVIIdocAdaptorWrapper.log
```

- IDocPollSource logs using XPL.

You can view the real-time trace output using the XPL Viewer; see section [Installing XPL Viewer](#) on page 23.

You can view the log output by going to the file(s):

```
XPL-install-dir\data\log\AccSAP_IdocAdaptor.log.n.lang
```

where:

- *lang* indicates the language of the log messages.
- *n* indicates the version number of the log file. The latest log file is numbered zero (0)

For example on Microsoft Windows:

```
XPL-install-dir\data\log\AccSAP_adaptor-name.log.0.en_US
```

where *adaptor-name* is the name that you have given your adaptor.

On an HP-UX system, the log file is located and named as follows:

```
/var/opt/ov/log/AccSAP_adaptor-name.log.0.en_US
```

Configuring HPBPI to Monitor SAP Events

When you have configured the adaptor to send the required Business events to HPBPI, you then need to link these Business events into the Flow definition for the SAP business flow that represents the states of the IDoc instances that you want to monitor.

When you create the Data definition within the Business Flow, you must either use the values as defined in the adaptor properties file as the Business Events Fields (`Idoc.event.fieldn`), or you must provide a transformation pipe to map the Business Event Fields within the IDoc to the Data definition names defined within HPBPI.

The example in section [Example SAP System](#) on page 64 uses the exact matching option. Refer to [Example - Configuring Name Mapping](#) on page 75 for an example of using a transformation pipe.

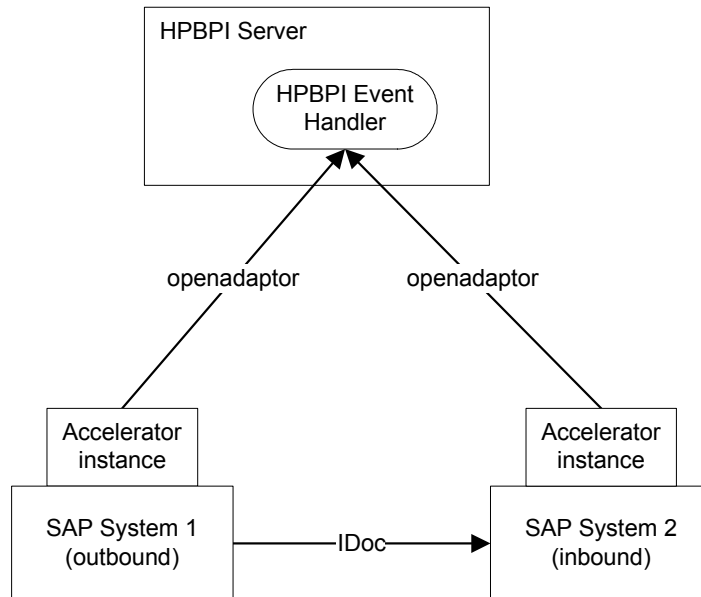
Example - Configuring Idoc Adaptors

This sections provides an example of an HPBPI business flow that is monitoring the status of a Materials IDoc.

Example SAP System

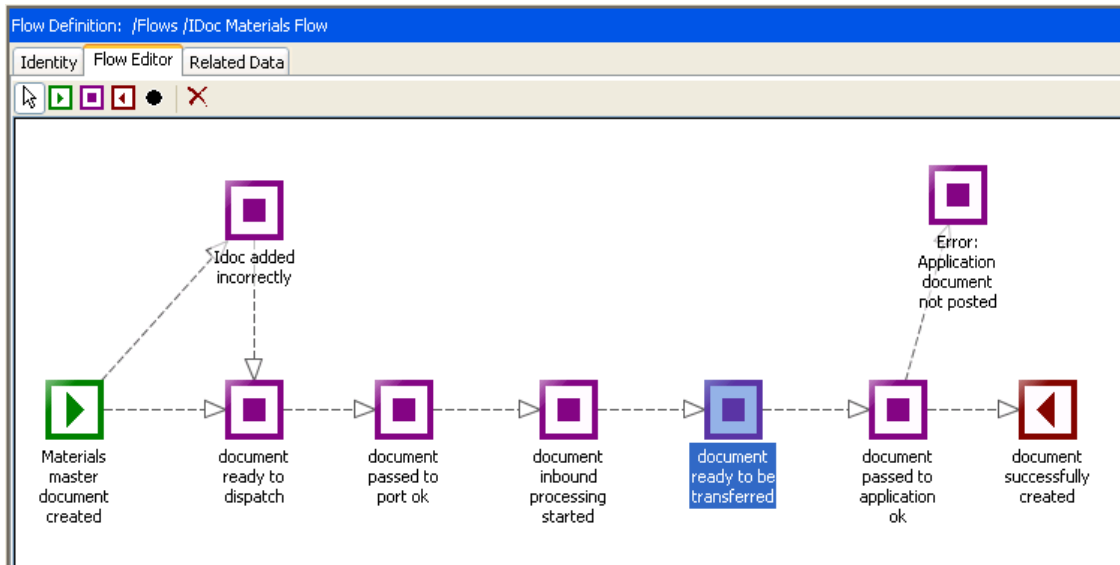
In this example, a Materials IDoc is being transferred between two SAP systems SAP System 1 and SAP System 2. There is an adaptor for SAP configured for each system; one monitoring the IDocs leaving SAP System 1 (outbound) and one monitoring the IDocs being received by SAP System 2 (inbound).

Figure 5 SAP System Configuration



The HPBPI flow definition and flow diagram for the Materials flow is shown in [Figure 6](#).

Figure 6 Business Flow for IDoc Materials



The Data and Event definitions for this flow are discussed later in this chapter. The next sections show examples of the adaptor properties files for the two adaptors for the two SAP systems.

Adaptor Property Files

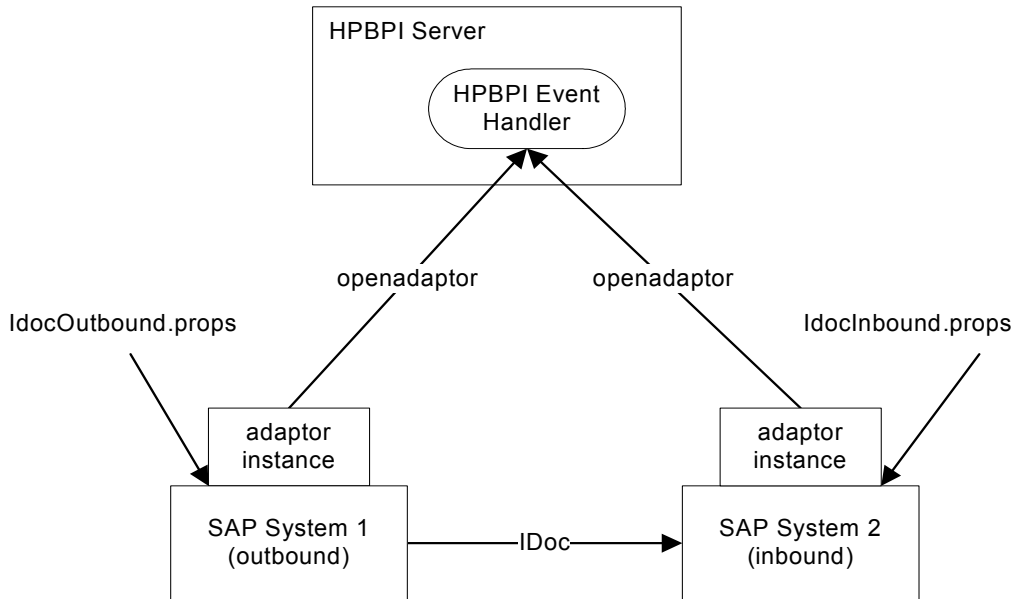
There are two adaptors that need to be configured; one for monitoring the IDoc as they are received by SAP system 1 and one monitoring the IDoc as they are received by SAP system 2. The adaptors are monitoring different values within the Partner Number of Receiver (RCVPRN) field in the IDocs.

There are two property files. One for each adaptor:

- IdocInbound.props
- IdocOutbound.props

Figure 7 shows these property files relative to the SAP system they relate to.

Figure 7 Property Files



The following sections describe these property files in more detail.

The property files for the adaptors are described in the following sections:

- [IdocInbound.props](#) on page 67
- [IdocOutbound.props](#) on page 69

IdocInbound.props

```
IdocInbound.Component1.Name = C1
IdocInbound.Component2.Name = C2

IdocInbound.C1.LinkTo1 = C2

# C1 - IdocPollSource
# -----
IdocInbound.C1.ClassName = com.hp.ov.sap.bia.idocadaptor.IdocPollSource

# Polling
# -----
# Polling time in milliseconds
IdocInbound.C1.PollSource = true
IdocInbound.C1.PollPeriod = 180000
IdocInbound.C1.MaxCallsPerPoll = 1

# BPI license check
# -----
IdocInbound.C1.BPI.serverhostname = bpiserver.domain
IdocInbound.C1.BPI.rmiregistryport = 44000

# SAP Connection
# -----
IdocInbound.C1.SAP.hostname = sapservers.domain
IdocInbound.C1.SAP.sid = sap_sid
IdocInbound.C1.SAP.client = sap_client
IdocInbound.C1.SAP.systemnumber = sap_number
IdocInbound.C1.SAP.user = myUser
IdocInbound.C1.SAP.password = 1;5y1p62e5k512n0w6l3u031r
IdocInbound.C1.SAP.language = EN
IdocInbound.C1.SAP.timezone = @Europe/London

# SAP Interface Tracing
# -----
IdocInbound.C1.Tracing.SAPRfc.tracelevel = 0
IdocInbound.C1.Tracing.SAPRfc.tracepath = c:/HP/BPI/data/log/
IdocInbound.C1.Tracing.SAP.queryconditions.enable = true
IdocInbound.C1.Tracing.SAP.querytable = options.html

# Delta time lag
# -----
# This property configures the delta time lag in seconds.
IdocInbound.C1.deltatime = 30

# Processing mode
# -----
IdocInbound.C1.Idoc.processing.type = inbound
```

```

# Idoc filter
# -----
IdocInbound.query = RCVPRN EQ 'ABC123'

# Idoc event fields to send
# -----
IdocInbound.event.field1 = STATUS
IdocInbound.event.field2 = MESTYP
IdocInbound.event.field3 = DOCTYP
IdocInbound.event.field4 = RCVPOR
IdocInbound.event.field5 = RCVPRT
IdocInbound.event.field6 = DOCNUM
IdocInbound.event.field7 = CURRENTDOCNUM

# BPI event
# -----
IdocInbound.C1.eventname = SAP/IdocStatus

# C2 - SocketSink
# =====
IdocInbound.C2.ClassName = org.openadaptor.adaptor.standard.SocketSink
IdocInbound.C2.HostName = bpiserver.domain
IdocInbound.C2.Port = 44005

```

IdocOutbound.props

```
IdocOutbound.Component1.Name = C1
IdocOutbound.Component2.Name = C2

IdocOutbound.C1.LinkTo1 = C2

# C1 - IdocPollSource
# -----
IdocOutbound.C1.ClassName = com.hp.ov.sap.bia.idocadaptor.IdocPollSource

# Polling
# -----
# Polling time in milliseconds
IdocOutbound.C1.PollSource = true
IdocOutbound.C1.PollPeriod = 180000
IdocOutbound.C1.MaxCallsPerPoll = 1

# BPI license check
# -----
IdocOutbound.C1.BPI.serverhostname = bpiserver.domain
IdocOutbound.C1.BPI.rmiregistryport = 44000

# SAP Connection
# -----
IdocOutbound.C1.SAP.hostname = sapsrver2.domain
IdocOutbound.C1.SAP.sid = sap_sid2
IdocOutbound.C1.SAP.client = sap_client2
IdocOutbound.C1.SAP.systemnumber = sap_number2
IdocOutbound.C1.SAP.user = myUser2
IdocOutbound.C1.SAP.password = 1;5y1p612n0w613u031r2
IdocOutbound.C1.SAP.language = EN
IdocOutbound.C1.SAP.timezone = @Europe/Germany

# SAP Interface Tracing
# -----
IdocOutbound.C1.Tracing.SAPRfc.tracelevel = 0
IdocOutbound.C1.Tracing.SAPRfc.tracepath = c:/HP/BPI/data/log/
IdocOutbound.C1.Tracing.SAP.queryconditions.enable = true
IdocOutbound.C1.Tracing.SAP.querytable = options.html

# Delta time lag
# -----
# This property configures the delta time lag in seconds.
IdocOutbound.C1.deltatime = 30

# Processing mode
# -----
IdocOutbound.C1.Idoc.processing.type = outbound
```

```

# Idoc filter
# -----
IdocOutbound.query = RCVPRN EQ 'ABC123'

# Idoc event fields to send
# -----
IdocOutbound.event.field1 = STATUS
IdocOutbound.event.field2 = MESTYP
IdocOutbound.event.field3 = DOCTYP
IdocOutbound.event.field4 = RCVPOR
IdocOutbound.event.field5 = RCVPRT
IdocOutbound.event.field6 = DOCNUM
IdocOutbound.event.field7 = CURRENTDOCNUM

# BPI event
# -----
IdocOutbound.C1.eventname = SAP/IdocStatus

# C2 - SocketSink
# =====
IdocOutbound.C2.ClassName = org.openadaptor.adaptor.standard.SocketSink
IdocOutbound.C2.HostName = bpiserver.domain
IdocOutbound.C2.Port = 44005

```

Refer to [Appendix A, SAP EDIDC Table Fields](#) for details of the IDoc business event fields that are available to you for your HPBPI business flows.

Event and Data Definitions for HPBPI

This section describes the Data and Event definitions used for this example flow.

Event Definition

The example Event definition is named:

`SAP/IdocStatus`

The content of the event comprises:

- CURRENTDOCNUM
- DOCNUM
- DOCTYP
- MESTYP
- RCVPOR
- RCVPRT
- STATUS

These property names match the names of the IDoc header fields as shown in [Appendix A, SAP EDIDC Table Fields](#). These are the fields that you want to extract from the IDoc header and use to progress your business flow and for reporting on the IDoc. You need to consult your SAP documentation for the latest and complete set of header fields.

It is the Status field, from the EDIDC table that is used to progress the flow. As an example when the IDoc Status field is set to 1 it means an IDoc has been generated; this is the condition that causes the HPBPI business flow to start.

How the IDoc STATUS field is used when progressing each node in the example flow are shown in [Table 7](#).

Table 7 Example Flow - Node and IDoc Status Values

Node Name	Status Value Used for Progression
Idoc generated	1 - start a new instance of the flow as a new IDoc has been created.
ready to dispatch	30 -IDoc is ready for dispatch, therefore progress the flow to next node.
Idoc passed to port ok	3 - Data passed to port OK - progress the flow to the next node.
Idoc added	50 - IDoc added - progress the flow to the next node.
Idoc ready to be transferred	64 - IDoc ready to be transferred to application - progress the flow to the next node.
Idoc passed to application	62 - IDoc passed to application - progress the flow to the next node.

Table 7 Example Flow - Node and IDoc Status Values

Node Name	Status Value Used for Progression
Application document posted	53 - Application document has been posted; therefore the flow is considered to have Completed.
Idoc added incorrectly	37: Error condition where the IDoc has been added incorrectly. When the problem is fixed, the status of the IDoc changes and the business flow progresses. If the error is not corrected, the flow cannot progress for the IDoc.
Error: Application document not posted	51 - Error condition where the IDoc has not been posted as expected. When the problem is fixed, the status of the IDoc changes and the business flow progresses. If the error is not corrected, the flow cannot progress for the IDoc.

An example of the set of values available for STATUS is listed in [Appendix B, IDoc Status Fields](#). You need to find the latest and complete set of values from your SAP documentation.

Figure 8 shows the Data definition for the example flow from within the HPBPI Modeler.

Figure 8 Data Definition for Materials Flow

Data Definition: /Data /IdocStatusInfo

Identity Properties Subscriptions

Name ▲	Type	Constraint	Unique	Description
CURRENTDOCNUM	String	20	<input type="checkbox"/>	
DOCNUM	String	20	<input checked="" type="checkbox"/>	
DOCTYP	String	40	<input type="checkbox"/>	
MESTYP	String	40	<input type="checkbox"/>	
RCVPOR	String	20	<input type="checkbox"/>	
RCVPRT	String	20	<input type="checkbox"/>	
STATUS	String	20	<input type="checkbox"/>	

[Click here to create a new property...](#)

Todo List: Errors: 0 Warnings: 0

Severity ▲	Location	Description
------------	----------	-------------

Example - Configuring Name Mapping

When configuring an IDoc adaptor to collect data fields from the IDoc header, the field names from the EDIDC table become the names of these fields within your adaptor. If these names match the names defined in the HPBPI event definition then your adaptor can send these directly to HPBPI. If your HPBPI event definition has defined these fields using different names then you need to configure a transformation pipe within your IDoc adaptor to map these names.

As an example, your `IdocPollSource` might be configured to extract the following fields from the EDIDC table:

```
IdocAdaptor.C1.Idoc.event.field1 = STATUS
IdocAdaptor.C1.Idoc.event.field2 = CURRENTDOCNUM
IdocAdaptor.C1.Idoc.event.field3 = DOCNUM
IdocAdaptor.C1.Idoc.event.field4 = DOCTYP
IdocAdaptor.C1.Idoc.event.field5 = MESTYP
IdocAdaptor.C1.Idoc.event.field6 = SERIAL
```

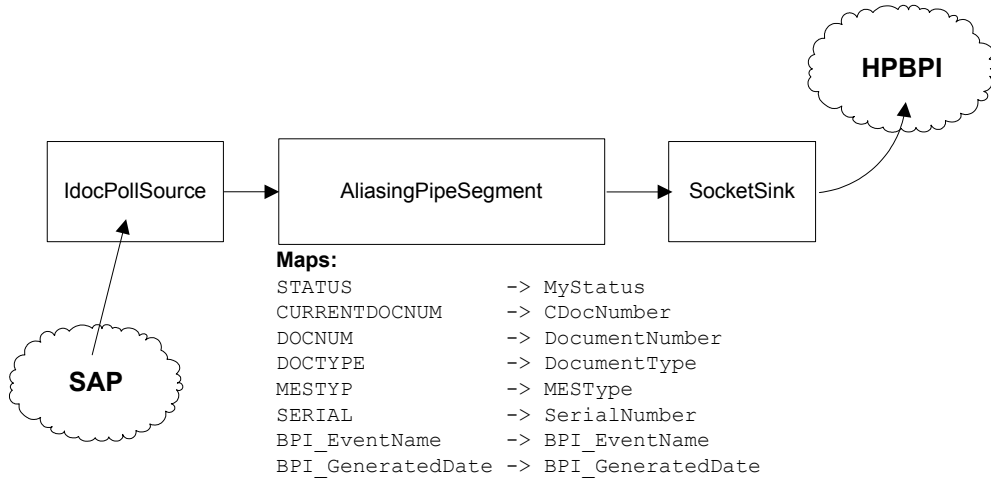
Within HPBPI, the Event definition fields are named as follows:

```
MyStatus
CDocNumber
DocumentNumber
DocumentType
MESType
SerialNumber
```

The adaptor needs to link in a pipe such that the field names are mapped to the field names defined within the HPBPI Event definition.

The architecture for the mapping is shown in [Figure 9](#) on page 76.

Figure 9 IDoc Adaptor with Field Mapping



The `AliasingPipeSegment` pipe is a standard component provided by `openadaptor`. This pipe enables you to map the field names that come from the `IdocPollSource` component and rename them to match the names as defined within `HPBPI`.

As well as mapping the names for the fields selected from the `EDIDC` table, the pipe also needs to map the two `HPBPI`-specific fields (`BPI_EventName` and `BPI_GeneratedDate`) that are set by the `IdocPollSource`.

The Configuration Segments

To add in an `AliasingPipeSegment` as shown in [Figure 9](#) on page 76, make the following changes in the IDoc adaptor properties file:

- Define an additional component, as follows:

```
IdocAdaptor.Component3.Name = MAP
```

- Change the component linkage to include your new pipe, as follows:

```
IdocAdaptor.C1.LinkTo1 = MAP  
IdocAdaptor.MAP.LinkTo1 = C2
```

- Add the configuration for your pipe, as follows:

```
IdocAdaptor.MAP.ClassName      =  
                                org.openadaptor.adaptor.standard.AliasingPipeSegment  
IdocAdaptor.MAP.Type1         = Any Any  
IdocAdaptor.MAP.Type1.Alias1  = BPI_EventName BPI_Eventname  
IdocAdaptor.MAP.Type1.Alias2  = BPI_GeneratedDate BPI_GeneratedDate  
IdocAdaptor.MAP.Type1.Alias3  = STATUS MyStatus  
IdocAdaptor.MAP.Type1.Alias4  = CURRENTDOCNUM CDocNumber  
IdocAdaptor.MAP.Type1.Alias5  = DOCNUM DocumentNumber  
IdocAdaptor.MAP.Type1.Alias6  = DOCTYP DocumentType  
IdocAdaptor.MAP.Type1.Alias7  = MESTYP MESType  
IdocAdaptor.MAP.Type1.Alias8  = SERIAL SerialNumber
```

Notice the line:

```
IdocAdaptor.MAP.Type1          = Any Any
```

The value of the `Type1` property states that the mapping applies only to records of type `Any`. This is the default record type assigned by the `IdocPollSource`; therefore, by default, all records output from the `IdocPollSource` component are of type `Any`.

The other lines in the configuration define the incoming field name and the corresponding new name for the field.

You need to supply a mapping for every field that you want to pass through to the sink.

An Example Adaptor Properties File

The following is an example of a complete adaptor properties file where you have configured an `AliasingPipeSegment` to map the field names as shown in [Figure 9](#) on page 76.

```
IdocAdaptor.Component1.Name = C1
IdocAdaptor.Component2.Name = C2
IdocAdaptor.Component3.Name = MAP

IdocAdaptor.C1.LinkTo1 = MAP
IdocAdaptor.MAP.LinkTo1 = C2

# C1 - IdocPollSource
# -----
IdocAdaptor.C1.ClassName = com.hp.ov.sap.bia.idocadaptor.IdocPollSource

# Polling
# -----
IdocAdaptor.C1.PollSource = true
IdocAdaptor.C1.PollPeriod = 180000
IdocAdaptor.C1.MaxCallsPerPoll = 1

# BPI license check
# -----
IdocAdaptor.C1.BPI.serverhostname = BPIserver.domain
IdocAdaptor.C1.BPI.rmiregistryport = 44000

# SAP Connection
# -----
IdocOutbound.C1.SAP.hostname = sapservers.domain
IdocOutbound.C1.SAP.sid = sap_sid
IdocOutbound.C1.SAP.client = sap_client
IdocOutbound.C1.SAP.systemnumber = sap_number
IdocOutbound.C1.SAP.user = myUser
IdocOutbound.C1.SAP.password = 1;5y1p612n0w6l3u031r
IdocOutbound.C1.SAP.language = EN

# SAP Interface Tracing
# -----
IdocOutbound.C1.Tracing.SAPRFC.tracelevel = 0
IdocOutbound.C1.Tracing.SAPRFC.tracepath = c:/HP/BPI/data/log/
IdocOutbound.C1.Tracing.SAP.queryconditions.enable = true
IdocOutbound.C1.Tracing.SAP.querytable = options.html

# Delta time lag
# -----
IdocAdaptor.C1.deltatime = 30
```

```

# openadaptor DO record type
# -----
IdocAdaptor.C1.Type = Any

# Idoc event fields to send
# -----
IdocAdaptor.C1.Idoc.event.field1 = STATUS
IdocAdaptor.C1.Idoc.event.field2 = CURRENTDOCNUM
IdocAdaptor.C1.Idoc.event.field3 = DOCNUM
IdocAdaptor.C1.Idoc.event.field4 = DOCTYP
IdocAdaptor.C1.Idoc.event.field5 = MESTYP
IdocAdaptor.C1.Idoc.event.field6 = SERIAL

# BPI event
# -----
IdocAdaptor.C1.eventname = SAP/IdocStatus

# C2 - SocketSink
# -----
IdocAdaptor.C2.ClassName = org.openadaptor.adaptor.standard.SocketSink
IdocAdaptor.C2.HostName = localhost
IdocAdaptor.C2.Port = 44005

#-----
# Map the names from SAP names to the names within the HPBPI event definition
#
# As well as the field names (as chosen in the Idoc.event.field<n> properties)
# you need to map BPI_EventName and BPI_GeneratedDate.
# You also need to set the DOType (which is normally 'Any').
# You need to map every field that you want to pass through to HPBPI.
#-----

IdocAdaptor.MAP.ClassName =
    org.openadaptor.adaptor.standard.AliasingPipeSegment

IdocAdaptor.MAP.Type1 = Any Any
IdocAdaptor.MAP.Type1.Alias1 = BPI_EventName BPI_Eventname
IdocAdaptor.MAP.Type1.Alias2 = BPI_GeneratedDate BPI_GeneratedDate
IdocAdaptor.MAP.Type1.Alias3 = STATUS MyStatus
IdocAdaptor.MAP.Type1.Alias4 = CURRENTDOCNUM CDocNumber
IdocAdaptor.MAP.Type1.Alias5 = DOCNUM DocumentNumber
IdocAdaptor.MAP.Type1.Alias6 = DOCTYP DocumentType
IdocAdaptor.MAP.Type1.Alias7 = MESTYP MESType
IdocAdaptor.MAP.Type1.Alias8 = SERIAL SerialNumber

```


A SAP EDIDC Table Fields

This appendix lists the fields that are available as Data definition properties for your HPBPI business flow. You are advised to refer to the SAP documentation for the definitive list of fields.

Use the EDIDC fields to obtain information from the IDoc headers.

Table 8 SAP Table EDIDC

Field	Description	Data Type	Length
MANDT	Client	CLNT	3
DOCNUM	Doc number	NUMC	16
DOCREL	SAP Release for IDoc	CHAR	4
STATUS	Status of IDoc	CHAR	2
DOCTYP	Doc type	CHAR	8
DIRECT	Direction for IDoc transmission	CHAR	1
RCVPOR	Receiver port (SAP System, EDI subsystem)	CHAR	10
RCVPRT	Partner type of receiver	CHAR	2
RCVPRN	Partner number of receiver	CHAR	10
RCVSAD	Recipient address (SADR)	CHAR	10
RCVSMN	SADR client (recipient)	CLNT	3
RCVSNA	SADR flag for international recipient address	CHAR	1
RCVSCA	Communication type (SADR) of recipient	CHAR	3

Table 8 SAP Table EDIDC

Field	Description	Data Type	Length
RCVSDF	SADR default flag for recipient address	CHAR	1
RCVSLF	Sequential Number of Recipient Address (SADR)	CHAR	3
RCVLAD	Logical address of recipient	CHAR	70
STD	EDI standard	CHAR	1
STDVRS	Version of EDI standard	CHAR	6
STDMES	EDI message type	CHAR	6
MESCOD	Logical message code	CHAR	3
MESFCT	Logical message function	CHAR	3
OUTMOD	Output mode	CHAR	1
TEST	Test flag	CHAR	1
SNDPOR	Sender port (SAP System, EDI subsystem)	CHAR	10
SNDPRT	Partner type of sender	CHAR	2
SNDPRN	Partner number of sender	CHAR	10
SNDSAD	Sender address (SADR)	CHAR	10
SNDSMN	SADR client (sender)	CLNT	3
SNDSNA	SADR flag for international sender address	CHAR	1
SNDSCA	Communication type (SADR) of sender	CHAR	3
SNDSDF	SADR default flag for sender address	CHAR	1
SNDSLFL	Sequential Number of the Sender Address (SADR)	CHAR	3

Table 8 SAP Table EDIDC

Field	Description	Data Type	Length
SNDLAD	Logical address of sender	CHAR	70
REFINT	Reference to interchange file	CHAR	14
REFGRP	Reference to message group	CHAR	14
REFMES	Reference to message	CHAR	14
ARCKEY	EDI archive key	CHAR	70
CREDAT	Doc creation date	DATS	8
CRETIM	Doc creation time	TIMS	6
MESTYP	Message type	CHAR	30
IDOCTP	Basic type	CHAR	30
CIMTYP	Extension	CHAR	30
RCVPFC	Partner function of receiver	CHAR	2
SNDPFC	Partner Function of Send	CHAR	2
SERIAL	Serialization field	CHAR	20
EXPRSS	Overriding in inbound processing	CHAR	1
UPDDAT	Date on which control record was last changed	DATS	8
UPDTIM	Time at which control record was last changed	TIMS	6
MAXSEGNUM	Number of data records	NUMC	6

B IDoc Status Fields

This appendix lists the IDoc status codes that you can use within your HPBPI progression rules when defining your business flow.

You are advised to refer to the SAP documentation for the definitive list of status codes.

Table 9 IDoc Status Codes

Status	Description
00	Not used, only R/2
01	Doc generated
02	Error passing data to port
03	Data passed to port OK
04	Error within control information of EDI subsystem
05	Error During Translation
06	Translation OK
07	Error during syntax check
08	Syntax check OK
09	Error during interchange handling
10	Interchange handling OK
11	Error during dispatch
12	Dispatch OK
13	Retransmission OK
14	Interchange Acknowledgement positive

Table 9 IDoc Status Codes

Status	Description
15	Interchange Acknowledgement negative
16	Functional Acknowledgement positive
17	Functional Acknowledgement negative
18	Triggering EDI subsystem OK
19	Data transfer for test OK
20	Error triggering EDI subsystem
21	Error passing data for test
22	Dispatch OK, acknowledgement still due
23	Error during retransmission
24	Control information of EDI subsystem OK
25	Processing despite syntax error (outbound)
26	Error during syntax check of IDoc (outbound)
27	Error in dispatch level (ALE service)
28	IDoc sent to ALE distribution unit retroactively
29	Error in ALE service
30	IDoc ready for dispatch (ALE service)
31	Error - no further processing
32	IDoc was edited
33	Original of an IDoc which was edited
34	Error in control record of IDoc
35	IDoc reloaded from archive
36	Electronic signature not performed (timeout)

Table 9 IDoc Status Codes

Status	Description
37	IDoc added incorrectly. The IDoc has arrived at a destination system, but is not recognized on that system as an IDoc that it can process.
38	IDoc archived
39	IDoc is in the target system (ALE service)
40	Application document not created in target system
41	Application document created in target system
42	IDoc was created by test transaction
50	IDoc added
51	Application document not posted
52	Application document not fully posted
53	Application document posted
54	Error during formal application check
55	Formal application check OK
56	IDoc with errors added
57	Test IDoc: Error during application check
58	IDoc copy from R/2 connection
59	Not used
60	Error during syntax check of IDoc (inbound)
61	Processing despite syntax error (inbound)
62	IDoc passed to application
63	Error passing IDoc to application
64	IDoc ready to be transferred to application
65	Error in ALE service

Table 9 IDoc Status Codes

Status	Description
66	IDoc is waiting for predecessor IDoc (serialization)
67	Not used
68	Error - no further processing
69	IDoc was edited
70	Original of an IDoc which was edited
71	IDoc reloaded from archive
72	Not used, only R/2
73	IDoc archived
74	IDoc was created by test transaction
75	IDoc is in inbound queue

C Additional HPBPI and SAP Integration Options

The chapter describes additional methods of integrating HPBPI with SAP. You can use these methods in addition to using the HPBPI Accelerator for SAP as they enable you to access different types of SAP information.

Be aware that some of the SAP integrations described in this chapter require some support and cooperation from a SAP consultant or partner. The SAP expert needs to understand and implement the integration from the SAP perspective.

This chapter is structured as follows:

- [Integrating with SAP](#) on page 90
- [Pull Integrations](#) on page 94
- [Push Integrations](#) on page 96

Integrating with SAP

When you consider how to integrate SAP with HPBPI, there are three main questions that need to be considered. These being:

1. How to get the data out of SAP?
2. Do you need to transform this data in any way?
3. How best to send this data into HPBPI?

But question 1 in this list can take some time to sort out. Any decision about how to get data out of the SAP system requires discussion and agreement between you, an SAP consultant and the customer.

Business Monitoring and BPM

An SAP consultant is used to addressing integration needs within a BPM (Business Process Management) type of scenario; that is, they are quite used to an integration model where a step within a business process, or application, needs to send data into SAP, or request data from SAP. In these cases, the application can typically make the appropriate BAPI/RFC call to complete process steps such as: Raise a Purchase Order or Get Order Details.

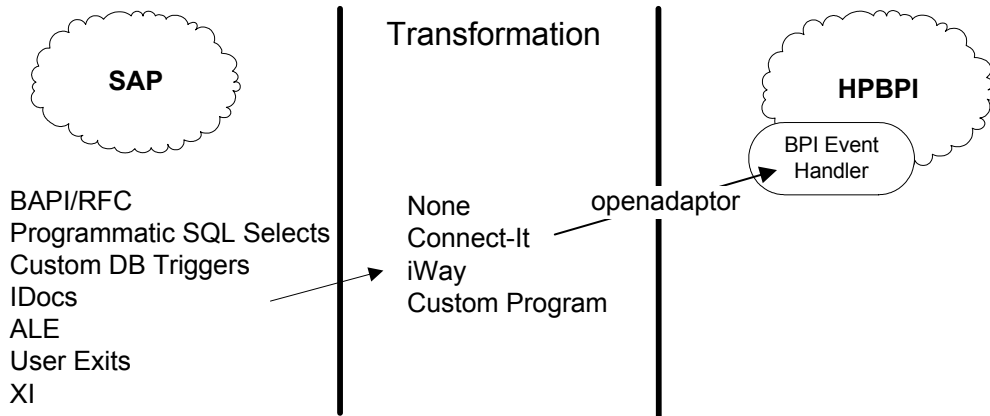
This familiarity means that an SAP consultant can often start by seeing the HPBPI integration as simply a call to a BAPI to pull the required data out of SAP; however, an HPBPI integration has different requirements. This is because HPBPI is monitoring SAP business processes, not orchestrating or taking part in them. Typically, HPBPI does not need to make synchronous calls to run specific function modules within SAP, it needs to be informed about certain events occurring within SAP. As an example, if HPBPI is monitoring how purchase orders are raised, SAP needs to send the appropriate purchase order details out to HPBPI when each new purchase order is raised; that is, SAP needs to make the details available to HPBPI as they happen.

If you use the BAPI/RFC approach to monitor how purchase orders are raised, you need to identify a BAPI call to provide something like: the set of Purchase Orders that have been raised since the last time you asked for them. You then need to periodically call this BAPI and get the latest updates to this information.

Integration Options

Returning to the three main questions that need to be considered when integrating SAP with HPBPI, [Figure 10](#) shows these three main areas of integration - SAP, Data Transformation and HPBPI. It also lists some of the possible options available to you.

Figure 10 Possible Options for the Areas of Integration

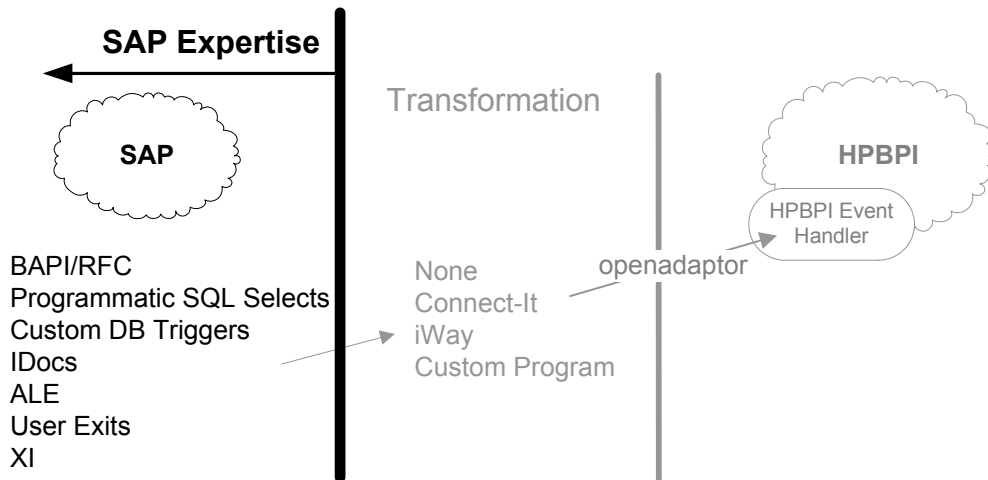


SAP Product Knowledge

Your SAP installation can be complex, and you need to work with a qualified SAP consultant or partner to implement a solution based on SAP. The SAP consultant is the one who can make the decisions about the appropriate mechanism for retrieving the business data, and it is the SAP consultant that is able to then implement that part of the integration.

So [Figure 10](#) on page 91 can be redrawn as shown in [Figure 11](#) on page 92:

Figure 11 SAP Expertise Required



You need to partner with an SAP consultant, and have them decide and implement the necessary mechanism for generating the business data from the SAP system.

When you have the necessary business data, you can then decide how to monitor that data within HPBPI. If the data needs to be transformed you can either use Connect-It, in which case, involve an iWay consultant, or you might decide to write a custom transformation program. When the data has been transformed, you are then able to make this data available to HPBPI using openadaptor and the Business Event Handler.

Generating Business Data for HPBPI

It is not immediately obvious from [Figure 10](#) on page 91 which mechanism, or combination of mechanisms, provides the best solution for your organization. All the options available for getting data out of SAP (IDocs, ALE, User Exits and so on) can provide the information you require; however, the SAP Basis administrator is very likely to have a preference for the kind of mechanism they want to use. As an example, they may prefer to use User Exits, or they may already use ALE and see that as their integration mechanism.

The mechanism you select also depends on the exact make up of the customer's SAP installation; for example, the applications they have installed, the version of the applications and so on.

There is no single answer to how best to integrate SAP and HPBPI, which is why you need to partner with an SAP consultant and work with them, and with the customer, to decide which strategy to use.

The following sections discuss the various integration mechanisms in more detail, grouping them as either push or pull integration mechanisms.

Pull Integrations

When implementing a pull mechanism there is a possibility that you can miss any events, or unnecessarily send events more than once. You need to take this into account when designing your integration.

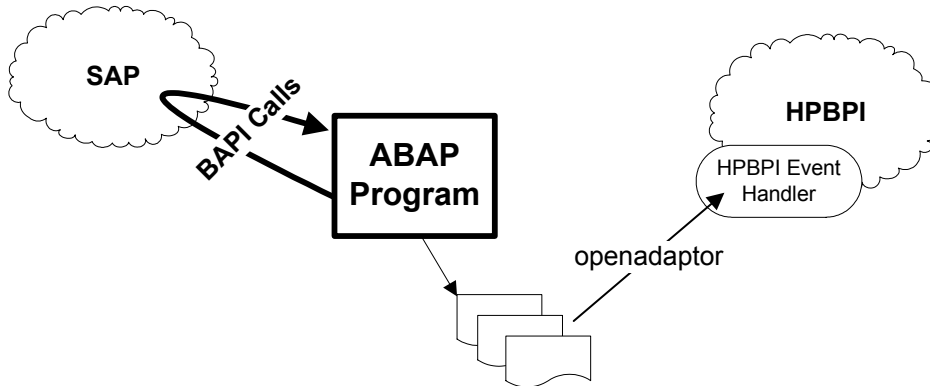
The following sections describe the main pull mechanisms available:

- [BAPI/RFC](#) on page 94
- [Programmatic SQL Selects](#) on page 95

BAPI/RFC

An SAP consultant can use BAPIs (RFCs) to write an ABAP program to periodically call a BAPI to get business data; for example, to get all new Purchase Order records. [Figure 12](#) shows the integration model when using BAPI.

Figure 12 Using BAPI/RFC



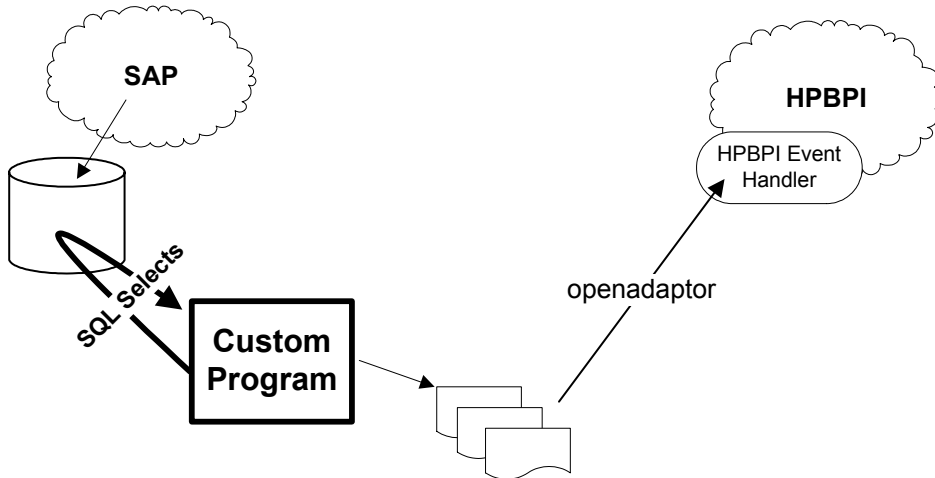
When using this mechanism, you need to make sure that your ABAP program does not miss any data, and you typically need to add program logic to ensure that you do not keep getting back the same data with each BAPI call.

Using BAPIs you can develop a solution where an integration program pulls the data from SAP at regular intervals.

Programmatic SQL Selects

An SAP consultant can write a custom program (probably in ABAP) to poll one or more tables in the SAP database looking for new and updated data records. As new data is written or updated in these tables, the program selects that data and exports it as required. Figure 13 shows the integration model when using SQL selects.

Figure 13 Using SQL Selects



When using this mechanism, make sure that:

- the Select statement polls the table requesting new records since the last time it polled. You need to include logic to ensure that you do not miss records.
- your code is able to detect which records are being updated and the attributes that have changed in cases where the table you are monitoring has records that are being constantly updated.
- your Select statement is able to identify the tables in the SAP database. An SAP database can have many thousands of tables, therefore finding the exact table, or tables, for your Select statement can be difficult.

There is also the issue that this is a pull mechanism, where the program is trying to pull the data from the database at each polling interval.

Push Integrations

Implementing a push mechanism can take more initial effort when configuring SAP, but it does produce a more efficient integration compared to the pull integration mechanisms.

The following sections describe the push integration mechanisms available:

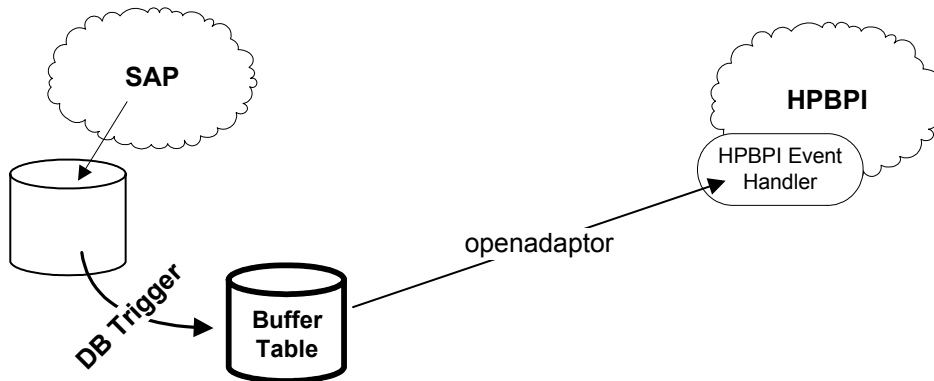
- [Custom DB Triggers](#) on page 96
- [IDocs](#) on page 97
- [ALE](#) on page 98 [User Exits](#) on page 100
- [XI](#) on page 102

Custom DB Triggers

By placing database triggers within an SAP database, you can capture the changes to the data and have the trigger write these out to a secondary buffer table. You can then configure the standard openadaptor database source to monitor this buffer table and send the business data events into HPBPI.

This trigger mechanism is shown in [Figure 14](#).

Figure 14 Using Database Triggers



Although this mechanism is effective, it is unlikely that any SAP Basis administrator is going to allow you to add database triggers to their SAP database. Typically, the SAP Basis administrator does not allow alterations to the SAP database in case it causes issues with their SAP support contract.

Another consideration when looking at this option is that the SAP database has many thousands of tables and finding the exact database table, or tables, on which to place one or more triggers can be difficult.

IDocs

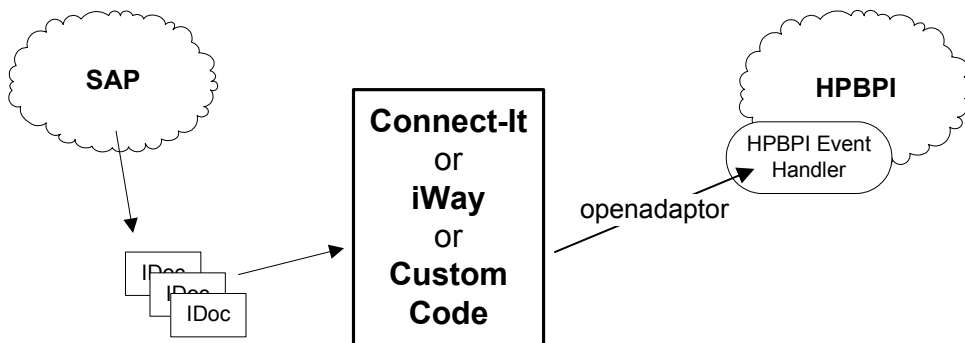
The various work flows within SAP can be configured to create IDocs as specific transactions occur; therefore, it is possible that the information you need in order to monitor an HPBPI business flow is already being written out as a series of IDocs. If this is the case, then you all you need to do is identify the appropriate IDocs; however, it is quite likely that existing IDocs are not capturing the exact information you require for monitoring a flow. In this case, the SAP consultant can help you configure the applications to produce the necessary IDocs.

An IDoc typically contains the data from a single transaction and writes this as a multi-line file; therefore, you need some mechanism to transform this IDoc into one, or more, business data events. This is where you can use products such a Connect-It or iWay.

Both Connect-It and iWay have tools that enable you to interrogate the IDoc and pull out the necessary data, formatting this in a way suitable for passing to openadaptor. Alternatively, you might decide that the format of the IDoc is such that you can write a custom program to extract the necessary data and pass this into openadaptor.

This mechanism is shown in [Figure 15](#).

Figure 15 Using IDocs with Connect-It, iWay or Custom Code



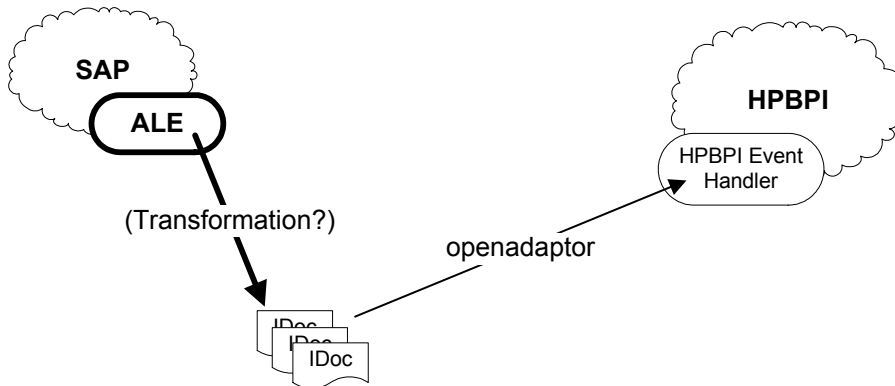
ALE

Application Link Enabling enables a SAP Basis administrator to monitor one or more SAP business objects. When a new instance of a business object is created or updated, ALE can be configured to send an IDoc to a destination PORT.

Within ALE, a destination PORT can be any output device, such as a file or a socket. It is also possible to configure ALE to complete a data transformation before sending the IDoc to this PORT.

Your ALE configuration is shown in [Figure 16](#).

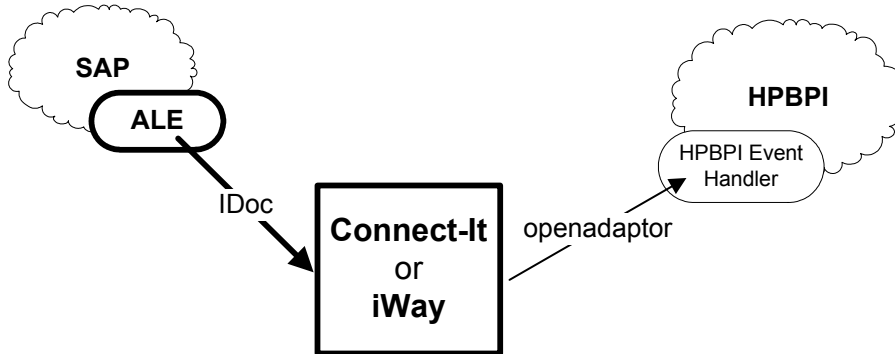
Figure 16 Using ALE



Using this mechanism, you might be able to configure the necessary data transformation within ALE such that the resultant IDocs are in a form that is readable by openadaptor. Alternatively, you might decide to have ALE simply produce the IDocs and then chose to use something like Connect-It, or iWay, to do the necessary transformations, before passing the data to openadaptor.

This mechanism is shown in [Figure 17](#).

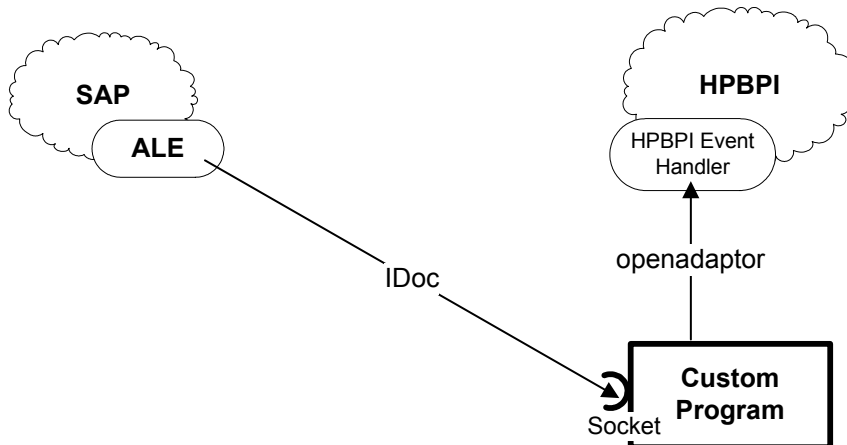
Figure 17 Using ALE with Connect-It or iWay



Another option is to use ALE to send the data directly to a socket on the HPBPI machine. You can then have a custom program listening on the socket that completes any final transformations before invoking openadaptor to pass the data into HPBPI.

This mechanism is shown in [Figure 18](#).

Figure 18 ALE Sending to a Socket on the HPBPI Machine



Using the ALE mechanism:

- You can complete some of the data transformation within ALE.
- Your custom program on the HPBPI machine can complete any transformations you require.
- Your custom program can invoke openadaptor and act as a local source, in openadaptor terminology.

This enables your custom program to pass the input data record directly to the openadaptor SocketSink and therefore directly to the Business Event Handler. If necessary, your adaptor can carry out further filtering or data transformations.

Using ALE is one of the more efficient integration mechanisms.

User Exits

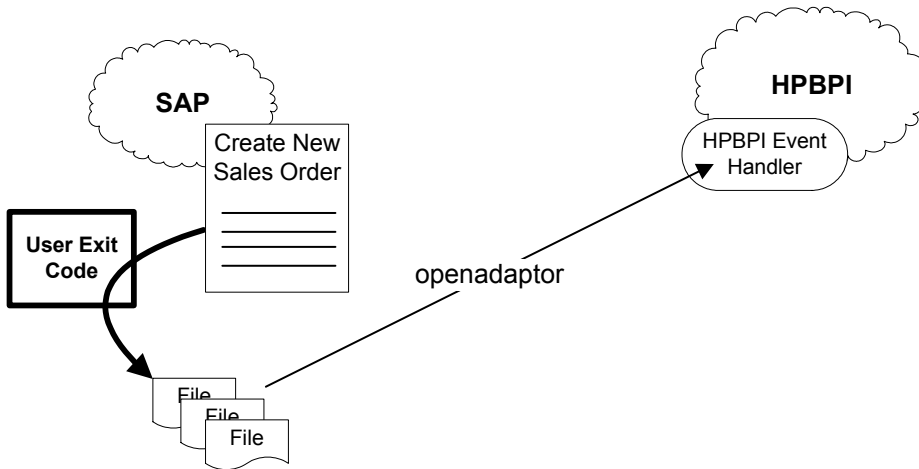
SAP user exits enable you to embed your own code (typically written in ABAP) within an existing SAP work flow; for example, you might want to embed a user exit within a `Create new sales order` flow, and capture the following data whenever a new sales order is created:

- sales order identifier
- customer identifier
- total value of the order

Your user exit code can then output the data as required by your integration.

Figure 19 shows an example of the architecture when the user exit code writes out each new sales order as a separate comma-delimited file.

Figure 19 User Exit Code Writing to Files



You can have more than one user exit for a work flow, which enables you to capture whatever data you require. You can also output the data to any location you need, and using whatever format is appropriate.

In Figure 19, you can configure the openadaptor file, FileRollSource such that it can access the directory in which your user exit code is writing its files. You then need to make sure that the user exit code creates the file without any filename extension, and when the file is complete, renames this file to have a file extension (for example, .txt). FileRollSource can then be configured to process only those files ending with the .txt file extension.

To write and embed user exit code, the SAP consultant needs to be able to write the ABAP code and identify where, within the SAP work flow, to insert this code. The SAP Basis administrator then needs to provide the necessary software unlock keys to enable the user exits to be embedded in the appropriate work flows.

XI

If you are using SAP XI, or NetWeaver (including XI), then a SAP consultant can choose to send SAP data across the XI bus. This involves sending and receiving XML data.

The SAP consultant can write a specific client to accept messages destined for HPBPI, or possibly use a file-based XI client to accept the messages destined for HPBPI, and simply write these as files into a directory. You could then configure an openadaptor FileRollSource to monitor this directory and send the data into the Business Event Handler.

Index

Symbols

- /HPOV/*S
 - SAP Transport object, 24
- /HPOV/BIAADAPTORS
 - SAP Transport object, 24
- /HPOV/BSM
 - SAP Transport object, 24

A

- ABAP
 - HPBPI Accelerator for SAP, 10
- Accelerator
 - architecture, 44
 - configuring, 41
 - installation procedure, 26
 - licensing, 35
 - permanent license, 36
 - re-installing, 40
 - removing from your machine, 38
 - trial extension license, 35

Adaptor

- and Business Events, 42
- and HPBPI, 63
- architecture, 44
- configuring, 41, 45
- example using SAP Materials IDOC, 64
- introduction, 10
- Java Service Wrapper, 57
- logging options, 62
- properties file, 45
- source component, 43
- starting as a Windows Service, 56
- starting on HP-UX, 59
- starting on Windows, 56
- starting using biaeventadaptor.bat, 56
- starting using runAdaptor, 56
- stopping on HP-UX, 61
- testing, 48
- tracing, 23

Adaptor properties

- Business Event properties, 53
- Event Data properties, 54
- Licensing properties, 48
- Polling properties, 47
- Processing Type properties, 51
- SAP Connection properties, 49, 50
- Sink Class property, 54
- Source Class Name property, 46
- Status Query properties, 52
- Test Mode
 - InhibitHistory, 48
- Tracing properties, 50, 51
- Update Time properties, 51

Advanced Business Application
Programming
SAP terminology, 16

ALE
SAP terminology, 15

Application Link Enabling
SAP terminology, 15

Architecture
for HPBPI Accelerator for SAP, 44

B

Basis Administrator
SAP terminology, 18

biaeventadaptor.bat, 56

BPI.rmiregistryport
Licensing property, 48

BPI.serverhostname
Licensing property, 48

BPI Accelerator for SAP licensing, 35

Business Application Programming
Interface
SAP terminology, 16

Business Event properties
Idoc.event.fieldn, 53

Business Events
adaptor configuration, 42

C

Checker script
validating the Accelerator installation,
33

ClassName
Sink Class property, 54
Source Class Name property, 46

Configuring the adaptor, 45

Configuring the HPBPI Accelerator for SAP,
41

D

deltatime
Update Time property, 51

Development class
SAP Transport class, 24

E

EDIDC
IDoc status, 42
table of fields, 81

EDIDS
IDoc status, 42

encodePassword script, 55

Event Data properties
eventname, 54

eventname
Event Data property, 54

Example
name mapping, 75
SAP Materials IDoc, 64

Exchange Infrastructure
SAP product, 13
SAP terminology, 16

F

Function group
SAP Transport class, 24

Function module
IDOC_GET_STATUS, 43
SAP Transport class, 24

H

HPBPI

- and SAP integration, 19
- configuration for the adaptor, 63

HPBPI Accelerator for SAP

- ABAP, 10
- architecture, 44
- disk space requirements, 21, 22
- installation procedure, 26
- introduction, 9
- software requirements, 22

I

IDoc

- how the adaptor identifies require IDocs to monitor, 45

Idoc.event.fieldn

- Business Event property, 53

Idoc.processing.type

- Processing Type property, 51

IDoc.query

- Status Query property, 52

IDOC_GET_STATUS

- function module, 43

IDocPollSource logging, 62

IDoc status

- EDIDS and EDIDC tables, 42
- table of status codes, 85

Importing SAP Transport files, 24

InhibitHistory

- Test Mode, 48

Installation

- description, 19
- on HP-UX, 27
- on Windows, 26
- required SAP libraries, 28
- requirements for an HP-UX system, 21
- requirements for a Windows system, 20
- tasks to complete before starting the installation, 20

Integrating with SAP

- options available, 89

Intermediate Document

- SAP terminology, 15

J

Java Service Wrapper

- starting the adaptor, 57

K

Key

- and Accelerator licensing, 35

L

Licensing

- Accelerator, 35
- permanent license key, 36
- property
 - BPI.rmiregistryport, 48
 - BPI.serverhostname, 48
- requirements, 35
- trial extension, 35

Logging options, 62

N

Name mapping

- configuration example, 75

NetWeaver
SAP product, 14
SAP terminology, 16

P

Password encoding utility, 55

Polling properties

PollPeriod, 47

PollSource, 47

PollPeriod

Polling property, 47

PollSource

Polling property, 47

Processing Type properties

Idoc.processing.type, 51

Properties

SAP.client, 49

SAP.gatewayhost, 50

SAP.gatewayservice, 50

SAP.hostname, 49

SAP.language, 49

SAP.password, 49

SAP.sid, 49

SAP.systemnumber, 49

SAP.user, 49

Properties file

for the adaptor, 45

R

R2

SAP product, 12

R3

SAP product, 12

Re-installing the Accelerator, 40

Remote Function Call

SAP terminology, 17

Removing the HPBPI Accelerator, 38

RFC

SAP terminology, 17

runAdaptor, 56

S

SAP

Exchange Infrastructure, 13

history of SAP applications, 12

integration with HPBPI, 19

NetWeaver, 14

R2, 12

R3, 12

System/R, 12

Web Application Server, 13

SAP.client

SAP Connection property, 49

SAP.gatewayhost

SAP Connection property, 50

SAP.gatewayservice

SAP Connection property, 50

SAP.hostname

SAP Connection property, 49

SAP.language

SAP Connection property, 49

SAP.password

SAP Connection property, 49

SAP.sid

SAP Connection property, 49

SAP.systemnumber

SAP Connection property, 49

SAP.user

SAP Connection property, 49

SAP Connection, 49, 50

SAP integration

options available, 89

SAP JCo package

obtaining, 30

- SAP libraries
 - installing, 28
- SAP Service Marketplace
 - obtaining required SAP libraries, 28
- SAP SID
 - and user profiles for Accelerator, 31
- SAP terminology, 15
 - Advanced Business Application Programming, 16
 - Application Link Enabling (ALE), 15
 - Basis Administrator, 18
 - Business Application Programming Interface (ABAP), 16
 - Exchange Infrastructure, 16
 - Intermediate Document, 15
 - NetWeaver, 16
 - Remote Function Call (RFC), 17
 - User Exit, 17
 - Work Flow, 17
- SAP Transport
 - Development class, 24
 - Function group, 24
 - Function module, 24
 - importing, 24
- SAP User
 - setting up, 31
- Service Wrapper logging, 62
- Sink Class property
 - ClassName, 54
- Source Class Name property
 - ClassName, 46
- Starting
 - adaptor
 - Java Service Wrapper, 57
 - adaptor as a Windows Service, 56
 - adaptor on HP-UX, 59
 - adaptor on Windows, 56

- Status codes
 - for IDocs, 85
- Status Query properties
 - IDoc.query, 52
- Stopping
 - adaptor on HP-UX, 61
- System/R
 - SAP product, 12

T

- Terminology
 - for SAP, 15
- Testing the adaptor, 48
- Trace
 - files, 23
- Tracing.SAP.queryconditions.enable
 - Tracing property, 51
- Tracing.SAP.querytable
 - Tracing property, 51
- Tracing.SAPRFC.tracelevel
 - Tracing property, 50
- Tracing.SAPRFC.tracepath
 - Tracing property, 50
- Tracing properties
 - Tracing.SAP.queryconditions.enable, 51
 - Tracing.SAP.querytable, 51
 - Tracing.SAPRFC.tracelevel, 50
 - Tracing.SAPRFC.tracepath, 50
- Transformation pipe
 - example configuration, 75

U

- Update Time properties
 - deltatime, 51
- User Exit
 - SAP terminology, 17

User profile
for Accelerator within SAP, 31

V

Validating the Accelerator installation
using the Checker script, 33

W

Web Application Server
SAP product, 13

Work Flow
SAP terminology, 17

X

XPL logging, 62

XPL Viewer, 23