About Oracle® Java CAPS Application Adapters

Part No: 821–2574
March 2011

120126@25097

Contents

About Application Adapters

The following topics provide information about the application adapters provided with Java CAPS:

- "About the Oracle Java CAPS Adapter for SWIFTAlliance Gateway" on page 5
- "About Oracle Java CAPS Adapter for Oracle Applications" on page 9
- "About Oracle Java CAPS Adapter for PeopleSoft" on page 10
- "About Oracle Java CAPS Adapter for SAP ALE" on page 10
- "About Oracle Java CAPS Adapter for Siebel EAI" on page 12
- "About the Oracle Java CAPS Adapter for SAP BAPI" on page 14
- "About Oracle Java CAPS Adapter for WebSphere MQ" on page 17

About the Oracle Java CAPS Adapter for SWIFTAlliance Gateway

SWIFTAlliance Gateway is a modular software package that is installed on top of the SWIFTNet Link (SNL) software, and is designed to enable application-to-application communication. Using the SWIFTNet interactive services, InterAct and FileAct, messages and files are typically exchanged between a customer application (client) and a central application (server) over the Secure IP Network (SIPN). SWIFTAlliance Gateway can handle large volumes of information and is therefore suitable for use with both client and server applications.

The subtopics listed here provide information about SWIFT Alliance Gateway and its Java CAPS adapter.

- "Introduction to SWIFTNet" on page 6
- "SWIFTAlliance Gateway" on page 6
- "SWIFTAlliance Gateway Remote API" on page 6
- "SWIFTNet Messaging Services" on page 7
- "The Oracle Java CAPS SWIFT Alliance Gateway Adapter" on page 8

Introduction to SWIFTNet

SWIFTNet is a global business messaging network for secure connectivity between institutions that participate in the financial services industry. As such, SWIFTNet is designed to satisfy institutional community requirements for interoperability of mission-critical financial software solutions.

SWIFTNet provides an assurance of infrastructure reliability, availability, access control, correspondent and message authentication, message integrity, and confidentiality, to business applications that are interconnected among a community of institutions. Optionally, SWIFTNet also provides non-repudiation support, message validation, store-and-forward, and role-based access control.

SWIFTAlliance Gateway

SWIFTAlliance Gateway is an interface product for SWIFTNet. It incorporates all the functionality of the SWIFTNet Link. Additionally, it provides several different connectivity and usability features for SWIFTNet users, providing solutions to a variety of system integration problems.

SWIFTAlliance Gateway is designed to concentrate traffic from multiple SWIFTAlliance WebStations. It provides a graphical user interface for the administration of the SWIFTAlliance Gateway and related SWIFTNet security administration functions.

SWIFTAlliance Gateway can serve as a message concentrator, receiving messages from various other applications for passage through SWIFTNet. It can receive these messages through host adapters, including a WebSphere MQ host adapter, for interfacing with business applications running on a variety of different types of computing platforms.

SWIFTAlliance Gateway Remote API

SWIFTAlliance Gateway Remote API (RA) is a software package that establishes a communication link with the RA Host Adapter component of SWIFTAlliance Gateway, either from a SWIFTNet application existing on a remote computer or from a SWIFTNet application existing on the computer where SWIFTAlliance Gateway is installed.

Using Remote API, applications developed to run directly on top of SNL software can use SWIFTAlliance Gateway transparently as a concentrator for their SWIFTNet traffic, thereby implementing the single window concept RA offers two sets of APIs: SWIFTNet Link specific, and SWIFTAlliance Gateway specific. Message flow, from an RA instance to SWIFTAlliance Gateway, is managed by the Remote API Host Adapter (RAHA), a subcomponent of SWIFTAlliance Gateway's Application Interface (AI).

SWIFTNet Messaging Services

SWIFTNet offers four messaging services, SWIFTNet InterAct, FileAct, Browse, and FIN. Of these four, the SWIFTAlliance Gateway specifically addresses FileAct and InterAct in client mode, with both Real Time and Store-and-Forward transfers.

SWIFTNet InterAct

SWIFTNet InterAct provides secure and reliable exchange of individual structured financial messages. SWIFT customers' messaging requirements vary from customer to customer but also from message to message. SWIFTNet InterAct offers you a broad range of telecommunication modes.

Store-and-Forward Messaging

SWIFTNet InterAct's store-and-forward capability is designed for messages that are destined for a large number of correspondents, many of whom may not be online at the time of transmission. It removes the uncertainty and inconvenience of worrying about whether or not your correspondents are online at the time you send the message. The message is delivered as soon as the recipient is ready to receive it. As a result, it provides an ideal way to send individual instructions, confirmations, and reports to large numbers of correspondents, some of whom may be in different time zones.

Real-Time Messaging

Real-time messaging offers a low-cost alternative to store-and-forward for messages which are destined for correspondents that are online at the time of transmission. As a result, it is ideal for sending individual instructions, confirmations, and reports to a few large correspondents, or for messages to market infrastructures.

SWIFTNet FileAct

SWIFTNet FileAct provides secure and reliable transfer of files, such as batches of structured financial messages or large reports. Typical applications include repetitive credit transfers such as pension or salary payments, securities value-added information and reporting, and regulatory reporting. SWIFTNet FileAct offers a variety of messaging modes.

Store-and-Forward File Transfers

SWIFTNet FileAct's store-and-forward capability ensures that your correspondents receive your message whether or they are online at the time of transmission. Messages are delivered when the recipient is ready to receive it. Store-and-Forward is an ideal way to send individual instructions, confirmations and reports to large numbers of correspondents, some of which may be in different time zones.

Real-time File Transfers

Real-time messaging provides a lower-cost alternative to store-and-forward for files that are destined for correspondents that are online at the time of transmission. This makes it ideal for sending files to a few large correspondents or market infrastructures.

The Oracle Java CAPS SWIFT Alliance Gateway Adapter

The Adapter for SWIFTAlliance Gateway (referred to as the SWIFT AG Adapter) enables the Oracle Java™ Composite Application Platform Suite (Java CAPS) to communicate with SWIFTAlliance Gateway 5.0.

The SWIFT AG Adapter is comprised of the following components:

- **Connector module**: A JCA 1.5 Resource Adapter, allows you to exchange messages or files across SWIFTNet, SWIFT's secure IP network.
- **NetBeans module**: Incorporates the adapter into Java CAPS and provides necessary design time and runtime functionality within the Suite.
- **SWIFT AG Object Type Definition**: Exposes SWIFTNet methods and attributes for use within a Java Collaboration to perform connectivity and business logic.

In addition to the OTD, the SWIFT AG Adapter provides Connectivity Map and External System configuration for design time configuration.

SWIFT AG Adapter Features

The SWIFT AG Adapter includes the following features:

- Supports InterAct and FileAct Services in client mode, with both Real Time and Store-and-Forward messaging
- Supports both synchronous and asynchronous operation modes
- Provides support for all the SWIFTNet Link (SNL) Primitives
- Supports dynamic configuration of InterAct and FileAct primitive attributes from the Java Collaboration Editor
- Supports dynamic configuration of SWIFT AG Remote API transport properties

SAGOutboundAdapter Object Type Definition

The adapter provides a SWIFTAlliance Gateway specific OTD (Object Type Definition), which exposes methods, attributes, and configuration properties. When it is incorporated in a Java Collaboration, the SAGOutboundAdapter OTD allows you to build powerful business logic into your Projects.

The SAGOutboundAdapter OTD is comprised of the following nodes:

- **Configuration**: enables dynamic configuration of the adapter at runtime
- **Constants**: provides various SNL constants
- **Primitives**: provides all of the SNL Primitives for advanced users
- **RemoteApis**: provides user access to the Remote API's client APIs
- **Services**: provide the InterAct and FileAct client implementations to support Real Time and Store-and-Forward messaging

In addition to the OTD, the SWIFT AG Adapter provides Connectivity Map and External System parameters for design time configuration.

About Oracle Java CAPS Adapter for Oracle Applications

The Oracle E-Business Suite 11i is a comprehensive enterprise resource planning (ERP) software package built upon Oracle's database technology. It is presented within an Internet environment, using online transaction processing to address the global requirements of today's typical enterprise.

The E-Business suite includes a large number of Product Families, grouped into software modules corresponding to what were once stand-alone computer systems used by individual departments. These Product Families are identified by their major business functions, such as:

- Financials
- Human Resources
- Manufacturing
- Marketing
- Sales

These Product Families are integrated together to share a common database, allowing a company's various departments to quickly and easily share information and communicate with each other.

Oracle Applications Basic Operation

The basic architecture of an Oracle system contains a set of base objects which are held in highly normalized core tables within the Oracle database. A de-normalized view of these base objects is provided in a set of Open Interface Tables (OITs), also maintained in the database. Data is passed from the Open Interface Tables to the core tables under the control of the Concurrent Manager.

In a typical scenario, an operator schedules an import job by means of the Oracle front end, which initiates the following procedure:

1. Data is passed from the Open Interface Tables to the core tables under the control of Import Jobs scheduled by the Concurrent Manager.
2. It then invokes the Oracle Concurrent Manager, which:
3. Validates the data in the Open Interface Table, based on a set of stored SQL procedures.
4. Inserts the validated rows into the Oracle Applications Database.

There are several limitations to this very basic scheme:

- Once data is in the Open Interface Table, it cannot be withdrawn or corrected.
- Data failing the validation process may be handled in different ways—some import scripts update the original rows with error codes, while other scripts log errors to a file, requiring user intervention.
- Only the default validation rules provided by Oracle are used in the validation process, and may not address specific customer requirements.
- There is no easy way to insert batches of data as a transactional unit—for example, where all inserts from a batch must succeed (if *any* fail, then *all* must fail).

About Oracle Java CAPS Adapter for PeopleSoft

PeopleSoft's Enterprise Resource Planning (ERP) software is a full-function application package that offers business applications for financials, human resources, customer relations, supply chain management, materials management, and business analytics. PeopleSoft provides what it calls "pure-Internet" architecture: Web-based applications designed to streamline a company's operations by integrating systems to effectively connect it's various departments, customers, and suppliers.

Java CAPS and the PeopleSoft Adapter enable PeopleSoft to easily and transparently integrate with legacy systems, enterprise applications, and other platforms. The Adapter for PeopleSoft exposes JCA and Web services compliant interfaces for the purpose of application and business integration.

About Oracle Java CAPS Adapter for SAP ALE

SAP ALE (Application Link Enabling) is a technology for exchange of business data between multiple SAP R/3 systems or SAP R/3 and customer applications. The vehicle for data exchange is an IDoc (Intermediate Document), which is basically a SAP defined message structure that serves as a container for the different types of application data being transmitted.

ALE provides SAP customers with a program distribution model and technology that enables them to transfer IDocs across various platforms and systems.

The SAP IDoc Format

IDocs are used as containers for information, and are used to exchange business data between systems.

Several hundred IDocs are supplied with each SAP R/3 system, serving as templates for a wide variety of applications. The IDoc hierarchy is represented by the following terminology:

- Message Types are related to specific applications such as Orders.
- IDoc Types are different versions of standard Message Types, such as Orders for specific items or services.

The Oracle Java CAPS SAP ALE Adapter

The SAP ALE IDoc Object Type Definition (OTD), when used with the SAP BAPI Adapter in Transactional Remote Function Call (tRFC) mode, enables Java CAPS projects to exchange data with SAP R/3 software using SAP's Intermediate Documents (IDocs) via the Application Link Enabling (ALE) interface.

The next two sections provide an overview of how to use the IDoc OTD and the SAP BAPI Adapter to send or receive IDocs to SAP R/3.

Inbound Data Flow: SAP R/3 to Java CAPS

During routine operations, an application on the SAP R/3 system generates a transaction designated for an external system. The ALE interface converts the data from the internal data format to the IDoc format, and sends it via tRFC to the SAP BAPI Adapter, acting as a RFC server.

The Java CAPS Project's business rules receive the IDoc data from the SAP BAPI Adapter, performs any necessary processing or routing, and sends the information to another adapter connected to the recipient system. Any necessary data transformation required for the target application is performed in your Project Collaborations.

1. The adapter reads in the required configuration parameters and establishes a network connection with the SAP R/3 system. The adapter acts an RFC server, receiving IDocs from the SAP R/3 system.
2. When the IDoc is sent from SAP R/3 via tRFC, the SAP BAPI Adapter uses the RFC OTD, IDOC_INBOUND_ASYNCHRONOUS, to receive the IDoc data.
3. IDoc data received by the IDOC_INBOUND_ASYNCHRONOUS OTD can be marshaled out of the OTD and unmarshaled into a IDoc OTD.
4. A file-based TID (Transactional ID) database is used to track transactions that have been committed successfully or rolled back.
5. If identified successfully, the process moves on to the next step. If not, the adapter composes the appropriate response and logs an exception in the log file.

6. If the Collaboration or Business Process fails, an exception is logged in the log file raised back to SAP R/3.
7. The adapter then repeats the procedure beginning with step 2.

Outbound Data Flow: Java CAPS to SAP R/3

In the outbound mode, you must first get the data into the IDoc OTD using its unmarshal method. From the IDoc OTD, you unmarshal the data into the IDOC_INBOUND_ASYNCHRONOUS RFC OTD which sends the IDoc to SAP R/3 using tRFC protocol.

1. When the Collaboration or Business Process starts to run, the adapter is initialized with its configuration properties.
2. The data is unmarshaled to the IDoc OTD before being sent to the SAP BAPI Adapter's RFC OTD---IDOC_INBOUND_ASYNCHRONOUS.
3. The SAP BAPI Adapter transmits the data to SAP R/3.
4. The SAP BAPI Adapter associates the next TID (from a persistent resetable counter) with the transformed outbound message and sends it via tRFC to the SAP R/3 host.
5. If no exceptions are raised by the receiving SAP R/3 host, the next TID is incremented.
6. The adapter repeats the procedure beginning with step 2.

Messages are sent to the SAP R/3 host via Transactional RFC (tRFC). With tRFC, the receiving SAP R/3 system relies on an unique Transactional ID (TID) sent with the message to ascertain whether or not a transaction has ever been processed by it before. The SAP BAPI Adapter assumes that all messages handled are new and assigns a new TID to each message.

Note – If you have IDoc data in a byte array format you may unmarshal it directly to the IDOC_INBOUND_ASYNCHRONOUS OTD without using the IDoc OTD first.

About Oracle Java CAPS Adapter for Siebel EAI

The Siebel EAI Adapter enables the application to exchange messages with the Siebel EAI interface via a Web server using open standards such as HTTP and XML. There are two distinct processes involved in using the Siebel EAI Adapter:

- A design-time process, in which you obtain information about the Siebel Interface Object; and
- A runtime process, in which you use the Project to exchange data with Siebel EAI.

Design-Time Process

The design-time process, which is an integral part of Project development, is primarily concerned with extracting metadata from the Siebel application. This metadata is then used to format the messages propagated by the adapter.

This process uses the Siebel EAI OTD Wizard, which prompts you for information to find and connect to the desired Siebel instance. The Wizard then connects to Siebel and extracts the business services that are exposed through the Siebel Web Engine. These services are presented to you for selection of the appropriate service and operation.

When the service and operation have been selected, an OTD representing the selections is generated and saved in the repository.

Runtime Process

During runtime, the Siebel EAI Adapter's components relay the contents of web requests to Java Collaborations or Business Processes for further processing and subsequent hand-off to an outbound Siebel EAI Adapter.

In routine operation, the Siebel EAI Adapter uses HTTP to post a Siebel XML-formatted message to Siebel. It also specifies one of the following actions to be performed on the XML message:

- Delete
- Upsert (Insert/Update)
- Query

The result is that a corresponding Workflow is executed to process the message. A Siebel Workflow is a customized business application for managing and enforcing business processes.

The Siebel EAI Adapter POSTs the message to the Web server. The Siebel Web Server Extension invokes the specified Business Service which, in turn, starts an internal Workflow.

The Workflow invokes the Siebel EAI XML Converter, which converts the information from XML into the Siebel internal format and presents it to the Siebel EAI Adapter. The information is then sent to the Siebel Server via the Siebel Object Manager.

If any data is to be returned, the EAI Siebel Adapter can pass the result to the EAI XML Converter and send the data back to the adapter as a Siebel XML message.

Workflow Templates

A set of Workflow Templates is included with the Siebel EAI Adapter. These Workflow Templates invoke the necessary Workflow Processes to map the data directly to or from the Siebel database. You can download these files from the Java CAPS sample site. For more information, see "Configuring Your System Before Creating the OTD" in *Developing OTDs for Oracle Java CAPS Application Adapters*.

Session and Sessionless Modes

You can run the Siebel EAI Adapter in either session or sessionless mode. When running in the default Sessionless mode, every message posted to Siebel is enveloped with the login method, negating the need for an explicit login. By contrast, when Siebel runs in Session mode, the collaboration must include both a login method at beginning and a logout method at the end. Session mode allows you to post multiple messages to Siebel within a loop between a single login and logoff statement. Session mode is only supported using the Java Collaboration Definition (JCD). You cannot use Session mode when using business processes in Business Process Manager.

Using the Siebel Message Header

Siebel EAI Adapter supports both Siebel integration objects and Application Service Interfaces (ASIs). A Siebel message header is required for most integration objects or ASIs. In a JCD, you can include the Siebel Message Header by invoking the appropriate methods provided in the Siebel EAI OTD. When creating business processes in Business Process Manager, the Siebel Message Header is automatically included when the appropriate web service operation (Query, Update, Insert, Delete) is selected. Also, be sure to set the integrationObjectName.

About the Oracle Java CAPS Adapter for SAP BAPI

This topic provides conceptual information about SAP BAPI and its Oracle Java CAPS adapter.

About SAP

SAP creates software for the Enterprise Resource Planning (ERP) business sector. The company main product is SAP R/3 which uses a three-tier application architecture—database, application server, and client—to facilitate real-time data processing.

About the Oracle Java CAPS SAP BAPI Adapter

The SAP BAPI Adapter enables Java CAPS Projects to exchange data with SAP R/3 software using Business Application Programming Interfaces (BAPIs), RFCs, and IDocs.

The SAP BAPI Adapter uses the SAP Java Connector (SAP JCo) to allow Java applications to access BAPIs and RFCs.

The functionality of the SAP BAPI Adapter simplifies the process of determining the requisite IMPORT, EXPORT, CHANGING, and TABLE parameters—collecting all the necessary data using the correct type and format, calling the Remote Function Module (RFM) that represents the BAPI, and then extracting and parsing data from the EXPORT and TABLE parameters.

Before it can be invoked, a BAPI or RFM requires the following parameters:

- **IMPORT** parameters: data provided to the BAPI
- **EXPORT** parameters: data returned by the BAPI
- **CHANGING** parameters: data provided to and/or returned by the BAPI/RFC
- **TABLE** parameters - data provided to and/or returned by the BAPI/RFC

The detailed metadata for these parameters such as descriptions of their value types and mandatory or optional nature, can be found under SAP transaction **SE37**.

The metadata for a BAPI/RFC in SAP R/3 is extracted by the BAPI wizard, which uses it to build the BAPI/RFC OTD. This OTD is used in Java Collaborations and Business Process Manager Business Processes to invoke or receive the BAPI/RFC call.

The SAP BAPI Adapter Data Flows

When the SAP BAPI Adapter communicates with the SAP R/3 software, it uses the RFC protocol. The list below shows the RFC types of communication used:

- Outbound (Java CAPS to SAP R/3): non-transactional (regular) RFC and transactional RFC (tRFC)
- Inbound (SAP R/3 to Java CAPS): non-transactional and transactional RFC (tRFC)

Outbound Data Flow: Java CAPS to SAP R/3

Outbound communications occur when the adapter receives data from Java CAPS and sends it to SAP R/3 by calling a specific BAPI or RFM. The figure below shows a non-transactional outbound process.

The figure above shows the following steps for the outbound data flow:

1. The Collaboration or Business Process populates the appropriate BAPI or RFC **Import, Changing,** and **Table** parameter nodes on the BAPI/RFC OTD with data from an inbound OTD.

2. The adapter logs onto the SAP R/3 application using preconfigured properties.
3. The adapter calls the BAPI OTD's `execute()` method. Any work performed is immediately committed by SAP R/3 through autocommit.
4. The SAP R/3 applications returns successfully.

Inbound Data Flow: SAP R/3 to Java CAPS

For the inbound data flow, the SAP BAPI Adapter can receive data from SAP R/3 via RFC or tRFC. The sections below describe each protocol.

To enable the SAP BAPI Adapter to receive data from SAP R/3, configure the Environment properties with an RFC destination created within SAP R/3.

Inbound Data Flow via RFC

The sequence diagram uses a sample CostCenter OTD to describe the RFC inbound sequence.

The figure above shows the following steps for the inbound data flow via RFC:

1. The Business Process is activated when an RFM call is received from SAP R/3.
2. Finding that data from an RFM is available, the Business Process accesses all pertinent data nodes and sends the gathered information to other Java CAPS components.
3. The adapter returns the results of the RFM execution back to SAP.

Inbound Data Flow via tRFC

Communication via tRFC is the similar to RFC, except that it adds transactional verification steps prior to committing or rolling back. tRFC is preferred over RFC because of the additional reliability. By using unique TIDs associated with a BAPI/RFM call, SAP R/3 processes the data once, and only once. The figure below shows inbound data flow via tRFC.

The figure above shows the following steps for the inbound data flow via tRFC:

1. The Business Process is activated when an RFM call is received from SAP R/3.
2. Finding that data from an RFM is available, the Business Process accesses all pertinent data nodes and sends the gathered information to other Java CAPS components.
3. The adapter returns the results of the RFM execution back to SAP R/3.
4. If the RFM call returned successfully without exceptions, SAP R/3 informs the adapter that the data can be committed by calling onCommitTID().
5. The adapter updates the TID in the file database as being Committed, commits the data, and sends an onCommitTID() return to SAP R/3.
6. If the RFM call did not return successfully for any reason, SAP R/3 informs the adapter that the data must be rolled back by calling onRollbackTID().

7. The adapter sends an onRollbackTID() return to SAP R/3, confirming that the TID was not committed.

About Oracle Java CAPS Adapter for WebSphere MQ

This topic provides conceptual information about WebSphere MQ and its Java CAPS adapter.

About IBM's WebSphere MQ

WebSphere MQ (formerly MQSeries™) from IBM™ is a client-server message broker supporting an open API (application programming interface), available on a variety of operating systems including AIX™, Solaris™, HP-UX™, and Windows™. WebSphere MQ is "middleware" that provides commercial messaging and queuing services. Messaging enables programs to communicate with each other via messages rather than direct connection. Messages are placed in queues for temporary storage, freeing up programs to continue to work independently. This process also allows communication across a network of dissimilar components, processors, operating systems, and protocols.

About the Oracle Java CAPS WebSphere MQ Adapter

The Adapter for WebSphere MQ (referred to as the WebSphere MQ Adapter) allows the Oracle Java CAPS ESB system to exchange data with IBM's WebSphere MQ. Java CAPS ESB, using the WebSphere MQ Adapter, uses business logic within a Collaboration or Business Process to perform operations for data identification, manipulation, and transformation. Messages are tailored to meet the communication requirements of specific applications or protocols. Queues or Topics provide non-volatile storage for data within the Java CAPS ESB system allowing applications to run independently of one another at different speeds and times.

The WebSphere MQ Adapter transparently integrates existing systems with IBM's WebSphere MQ. This document explains how to install and configure the WebSphere MQ Adapter.