Using EntireX Java ACI

This chapter covers the following topics:

- Introduction
- Delivered ACI Examples

Introduction

EntireX Java ACI is a Java class library (Java package) that provides access to the EntireX Broker ACI for Java programmers. You can visualize it as the Java "language binding" for the EntireX Broker ACI. In this way, it is related to, for example, the C and Natural language interfaces. It covers the whole Broker ACI which enables you to write applications for both client/server and publish-and-subscribe communication models.

EntireX Java ACI comes as a single Java package: *com.softwareag.entirex.aci*. All of the EntireX Broker ACI functionality is wrapped in a system of collaborating classes. Thus, it should appeal equally well to both the experienced Java programmer and to those familiar with the EntireX Broker ACI. The package also contains the framework for RPCs.

The class library is packaged in the *entirex.jar* file which can be found in the *classes* subdirectory of the EntireX installation.

Since EntireX Java ACI is only a thin interface layer to communicate with EntireX Broker, little "local" knowledge is implemented in it. All essential information will be passed to and received from EntireX Broker itself.

Documentation of the classes consists of the documentation generated by Javadoc and the documentation of the core API shipped with the JDK. This provides quick recognition for those familiar with this format.

Delivered ACI Examples

The EntireX Java ACI examples in subdirectory *JavaACI* of the Developer’s Kit are:

- The *Client.java* and *Server.java* sources show how to write client and server applications for non-conversational and conversational EntireX Broker communication. The *Server.java* examples can be used as a prototype for writing arbitrary, multithreaded EntireX Broker servers. This example also shows the usage of an attach server written in Java.
- The *ChessWhite.java* and *ChessBlack.java* sources demonstrate the usage of units of work, persistence and persistent status with the Java ACI.
- The *Publisher.java* and *Subscriber.java* sources demonstrate the publish-and-subscribe message model. The publisher just publishes a number of messages to a topic. All subscribers receive the messages published to a topic. The subscriber demonstrates a non-durable subscription and the use of the PublicationListener.

To run the examples, refer to the *README.TXT* delivered with the examples.