

BeanConnect™ for openUTM V2.0

Issue December 2006

Connection of J2EE Application Servers with openUTM

Pages 4

BeanConnect™ for openUTM connects applications based on J2EE application servers such as the Oracle Application Server 10g with applications of the TP monitor openUTM in accordance with the J2EE Connector Architecture (JCA) from SUN. New applications which are based on the J2EE technology can thus be integrated into an existing IT environment using Java resources.

In addition, Java classes are contained for access to openUTM from any Java program.

JCA V1.5 conformable, bidirectional adapter with transaction security

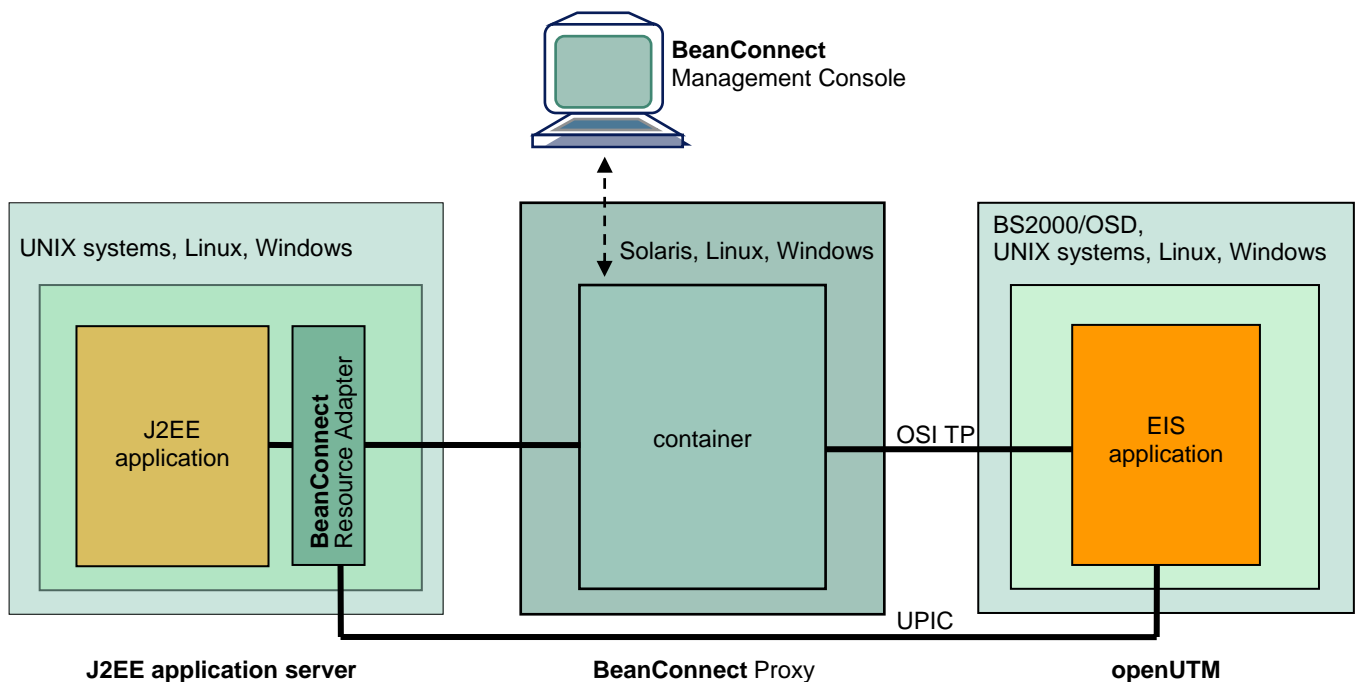
The J2EE Connector Architecture (JCA) is part of the Java 2 platform, Enterprise Edition (J2EE) as well as part of the Java Platform, Enterprise Edition 5 (Java EE 5). Based on this architecture user can write new applications using J2EE technology capabilities and can also encapsulate existing Enterprise Information Systems (EIS) with Enterprise JavaBeans or JavaServer Page technologies.

BeanConnect for openUTM V2.0 implements the connection of openUTM applications as EIS systems.

BeanConnect for openUTM V2.0 uses the JCA V1.5, which is part of the J2EE V1.4 and Java EE 5.

BeanConnect for openUTM V2.0 supports a variety of communication paradigms:

It enables outbound communication, i.e. communication is initiated by the application on the application server, and inbound communication, i.e. communication is initiated by the openUTM application. The communication can be transactional or non-transactional and can be executed dialog based as well as asynchronously.



Components

BeanConnect for openUTM V2.0 consists of the following components:

- BeanConnect Resource Adapter,
- BeanConnect Proxy,
- BeanConnect Management Console.

The **BeanConnect Resource Adapter** makes the JCA interface (JCA V1.5) available to the user. It is embedded (deployed) in the J2EE application server and runs as part of the J2EE application server within the same address space.

The **BeanConnect Proxy** represents the transactional connection between the resource adapter within the application server on the one hand and the openUTM application on the other hand. It includes a container based on the transaction monitor openUTM.

The **BeanConnect Management Console** is the Java GUI for configuring and administering the BeanConnect Proxy. BeanConnect doesn't contain components which run on the openUTM platform itself.

Interfaces and protocols

BeanConnect for openUTM V2.0 collaborates with the J2EE application server to provide system level mechanisms, the transactions, security and connection pooling mechanisms. For this JCA V1.5 defines certain system contracts between application server and EIS which are implemented by BeanConnect for openUTM V2.0.

Contracts for outbound communication:

- Connection Management
- Transaction Management
- Security Management

Contracts for inbound communication:

- Transaction Inflow
- Message Inflow

Contracts for the resource adapter lifecycle management and thread management:

- Lifecycle Management
- Work Management

Common Client Interface

The J2EE Connector Architecture defines a Common Client Interface (CCI) for EIS access. The CCI defines a standard API for application components. BeanConnect for openUTM V2.0 supports the CCI and offers an additional interface which clearly decreases the programming effort.

For communication to the EIS system the following protocols are used:

Outbound communication:

- OSI TP protocol for distributed transactions
- UPIC protocol (a proprietary protocol for clients of openUTM applications).

Inbound communication:

- OSI TP protocol for distributed transactions
- UPIC protocol
- Transport protocols such as the openUTM socket protocol and the RFC1006 protocol.

Mapping of COBOL data structures onto Java classes

By means of the supplied tool Cobol2Java the integration of BS2000/OSD-COBOL applications in Java programs is simplified by object-oriented mapping of COBOL data structures to Java classes.

Java classes for access to openUTM (openUTM-JConnect)

openUTM-JConnect are Java classes for any applications written in Java for access to openUTM server applications. The previously independent product openUTM-JConnect is part of BeanConnect for openUTM.

openUTM-JConnect supports all functions of the UPIC protocol and allows additionally the plug in of SLL security functions.

License structure of the product

- Processor licenses are assigned per processor core on which the J2EE application server with the BeanConnect Resource Adapter is running. Processor licensing allows any number of users (end user, developers, testers).
- If openUTM-JConnect is used, the licenses per processor running the Java applications are assigned.
- The cores of multi-core processors are counted by a factor of 0.5.
- Licenses for developers and testers are assigned per user (developer or tester) for a system without processor licensing.
- The software and documentation are delivered on CD-ROM.

Using BeanConnect with Oracle Application Server and other J2EE application servers.

BeanConnect for openUTM V2.0 is only released and maintained in connection with the Oracle Application Server. For other J2EE application servers the product can be used as well if they are JCA V1.5 conformable. The interoperability with the requested J2EE application server can be assured with an additional service package. The additional service package consists of two parts

- BeanConnect AS Interoperability Check
- BeanConnect AS Interoperability Service.

These service packages are described in separate data sheets. "BeanConnect AS Interoperability Check" contains an initial test to secure the interoperability between BeanConnect and the required application server and to fulfil the proprietary properties of the application server. "BeanConnect AS Interoperability Service" is a one-year-service-package to complement the standard service of BeanConnect for the use of BeanConnect with the requested J2EE application server.

**BeanConnect
is part of the
comprehensive
product offering of
openSEAS**



Technical data																												
TECHNICAL REQUIREMENTS HARDWARE	<p>The hardware which is compatible to the operating system versions listed below is supported. The BeanConnect components may run on one system or may be distributed to more systems. The functionality of BeanConnect remains unchanged when different operating systems or operating system versions are coupled</p> <p>CPU for BeanConnect Proxy: at least 450 MHz; recommended at least 1 GHz.</p> <p>Main memory demand for BeanConnect Proxy: at least 512 MB; recommended at least 1 GB.</p> <p>Hard disk:</p> <p>BeanConnect Resource Adapter: RAInstall.jar approx. 16 MB; extracted approx. 24 MB. BeanConnect Proxy / Management Console:</p> <table border="0"> <tr> <td>Central components:</td> <td>Solaris:</td> <td>approx. 153,0 MB</td> </tr> <tr> <td></td> <td>Linux:</td> <td>approx. 68,0 MB</td> </tr> <tr> <td></td> <td>Windows:</td> <td>approx. 60,0 MB</td> </tr> <tr> <td>Container:</td> <td>Solaris:</td> <td>approx. 55,0 MB</td> </tr> <tr> <td></td> <td>Linux:</td> <td>approx. 55,0 MB</td> </tr> <tr> <td></td> <td>Windows:</td> <td>approx. 55,0 MB</td> </tr> <tr> <td>Console:</td> <td>Solaris:</td> <td>approx. 0,1 MB</td> </tr> <tr> <td></td> <td>Linux:</td> <td>approx. 0,1 MB</td> </tr> <tr> <td></td> <td>Windows:</td> <td>approx. 3,0 MB</td> </tr> </table> <p>Advice: For outbound communication with the UPIC protocol the BeanConnect Proxy and the BeanConnect Management Console are not needed.</p>	Central components:	Solaris:	approx. 153,0 MB		Linux:	approx. 68,0 MB		Windows:	approx. 60,0 MB	Container:	Solaris:	approx. 55,0 MB		Linux:	approx. 55,0 MB		Windows:	approx. 55,0 MB	Console:	Solaris:	approx. 0,1 MB		Linux:	approx. 0,1 MB		Windows:	approx. 3,0 MB
Central components:	Solaris:	approx. 153,0 MB																										
	Linux:	approx. 68,0 MB																										
	Windows:	approx. 60,0 MB																										
Container:	Solaris:	approx. 55,0 MB																										
	Linux:	approx. 55,0 MB																										
	Windows:	approx. 55,0 MB																										
Console:	Solaris:	approx. 0,1 MB																										
	Linux:	approx. 0,1 MB																										
	Windows:	approx. 3,0 MB																										
TECHNICAL REQUIREMENTS SOFTWARE	<p>BeanConnect Resource Adapter</p> <p>Platforms:</p> <ul style="list-style-type: none"> • Windows2000, Windows Server 2003, Windows XP • Solaris (SPARC) as of V8.0 • RedHat Enterprise Linux 3 (AS or ES) • SUSE Linux Enterprise Server 8 (SLES8), 9 (SLES9)) • HP-UX as of 11i • HP Tru64 as of V5.1B <p>Software requirements:</p> <ul style="list-style-type: none"> • J2SE JDK as of 1.4.2_08 resp. software configuration of the application server • Application server with JCA 1.5 conformance <p>BeanConnect Proxy and BeanConnect Management Console</p> <p>Platforms:</p> <ul style="list-style-type: none"> • Windows2000, Windows Server 2003, Windows XP • Solaris as of V8 • RedHat Enterprise Linux 3.0 (AS or ES) • SUSE LINUX Enterprise Server 8 (SLES8), 9 (SLES9) <p>Software requirements:</p> <ul style="list-style-type: none"> • J2SE JDK as of 1.4.2_08 <p>Enterprise Information System (EIS)</p> <ul style="list-style-type: none"> • openUTM as of V5.1 on all platforms (BS2000/OSD, Unix systems, Linux, Windows). • On the BS2000 platform the product openUTM-D as of V5.1 is 																											

	required for transactional communication.
IMPLEMENTATION LANGUAGE	Java, C
USER INTERFACE	English
INSTALLATION	By the users themselves using the description in the manual
DOKUMENTATION	Manual on CD-ROM and downloadable from the web
USER REQUIREMENTS	Knowledge of the programming language Java and general knowledge of J2EE application server
TRAINING	Courses on the programming language Java and on J2EE application servers are recommended. Workshops about BeanConnect are offered by Fujitsu Siemens Computers.
CONDITIONS	This software product is supplied to the customer under the conditions of usage for software products on single / continuous payment.
WARRANTY	Class: A
	Delivery format: Java class files and machine language
	Examples are delivered in source code and are subject to warranty class B.