RFP 134 Connectors

Draft

10 Pages

Abstract

Resource adapters and resource managers, defined by the Java EE Connector Architecture specification, are an important component of many enterprise architectures. They are used both as a means to integrate non-standard resources into standard containers, as well as a pattern for integrating a standard resource into container infrastructure. This RFP defines the requirements for integrating connectors into OSGi, and describes the use cases and expected features that such an integration should include.

0 Document Information

0.1 Table of Contents

0 Document Information **2**
0.1 Table of Contents 2
0.2 Terminology and Document Conventions 3
0.3 Revision History 3

1 Introduction **3**

2 Application Domain **4**
2.1 Resource Adaptor 4
2.2 Client API 5
2.3 Container SPI 5
2.4 Directionality 5
2.5 Resource Adapter Metadata 5

3 Problem Description **5**

4 Use Cases **6**
4.1 Transactional Pooled Data Sources 6
4.2 Messaging Systems 6
4.3 Non-relational Access 6
4.4 Non-managed applications 7
4.5 Local Transactions 7
4.6 Secure Access 7

5 Requirements **7**
5.1 Configuration 7
5.2 Integration 8
5.3 Dynamicity 8
5.4 Resources 8
5.5 Compliance 8
5.6 Environment 9

6 Document Support **9**
6.1 References 9
6.2 Author's Address 9
6.3 End of Document 10

0.2 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 6.1.

```
Source code is shown in this typeface.
```

0.3 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	04/19/10	Mike Keith, Oracle michael.keith@oracle.com
0.1	04/28/10	Mike Keith, Oracle Changed RES0003 from MAY to MUST Added dependency on annotation processing RFP 135
0.2	05/21/10	Mike Keith Added ENV0004 as input from discussion at London F2F
0.3	05/26/10	Mike Keith Changed RES0002 to show outbound interactions are optional. Changed RES0004 to include other client interfaces in addition to CCI. Added CNF0004.

1 Introduction

Many enterprise technologies have been discussed within the Enterprise Expert Group, resulting in a series of specifications describing each of their integrations with OSGi. What has not been specified up to this point is a coordinated plan for integrating them into an overarching container environment. In some cases this will require further effort and evolution of the current OSGi specifications that describe those technology integrations, but in

other cases they can and often should be integrated as resource adapters, using the Java Connector Architecture.

In this RFP we outline some of the resources that are being specifically targeted as resource adapters, and outline the requirements that should be considered in the design of a functioning connector container.

2 Application Domain

Resource adapters are a critical part of applications built using Java Enterprise Edition (Java EE). The Connector model is a popular resource management and access development model that provides a well known API for applications to access various resource managers (such as databases, messaging systems, and legacy EIS and ERP systems) and SPI for resource adapters to exploit container services (such as transaction processing, security, connection management, work management, etc.). The Connector specification [3,4] describes how standard resource adapters function in an enterprise environment and defines standard contracts that allow bi-directional connectivity between enterprise applications and EISs.

The connector architecture enables an EIS vendor to provide a standard resource adapter for its EIS, shielding the user from any proprietary EIS communication protocols that may be used by the resource vendor for connectivity.

2.1 Resource Adaptor

A resource adapter is a collection of objects packaged up in a resource adapter archive (RAR). The contents of the archive include some or all of the following:

1. Outbound Connection Factories and related classes
2. Inbound Connection Factories and related classes
3. Utility classes
4. Resource files used by Java classes
5. Descriptive meta information as annotations, or in an XML descriptor file (ra.xml), that tie the above elements together

A resource adapter can list its dependencies on external libraries in the manifest as described in the JAR specification. During deployment of the resource adapter, the connector container must make the correct versions of the extensions available to the application following the rules defined by Optional Package Versioning [5].

The Java interfaces, implementation, and utility classes required by the resource adapter must be packaged as one or more JAR files as part of the resource adapter module. A JAR file must use the .jar file extension. These JAR files can reside anywhere inside the RAR file. Platform-specific libraries required by the resource adapter must also be packaged within the resource adapter module. The Connector specification provides a detailed description of the format of a RAR file.

Resource adapters can be packaged and signed in a resource archive (RAR) file using the standard Java archive tools.

2.2 Client API

Clients invoke operations on the resource by means of a Common Client Interface (CCI). They begin by obtaining a ConnectionFactory, from which a Connection object is procured. From the Connection object, InteractionSpec artifacts can be created and configured. By setting properties on the InteractionSpec the client can operate on the resource.

2.3 Container SPI

A server/container accesses a resource adapter through an SPI (Server Provider Interface) that governs how the container obtains connections from the resource. It is through the SPI that containers can provide pooling of connections, transactionality, either through global JTA transactions or local transactions private to the resource, security enforcement, as well as management of resource threads.

2.4 Directionality

Interactions with the resource adapter may be in any of three forms or directions:

1. Outbound interactions – The interaction is initiated by the application, and the resource adapter is called from within an application thread.

2. Inbound interactions – The interaction is initiated by the resource adapter calling into the application. The thread of control is obtained by the resource adapter from the container through the Work SPI.

3. Bidirectional interactions – Both inbound and outbound interactions occur.

Resource adapters may support only Outbound, or they may additionally support inbound or bidirectional interactions with the applcation.

2.5 Resource Adapter Metadata

The resource adapter metadata may be specified either as annotations on the resource classes or in an ra.xml deployment descriptor file. The metadata that needs to be specified includes the following types of configuration and deployment information:

- Initialization parameters for resource adapters and configuration
- Inbound and Outbound Connection and ConnectionFactory definitions, message listeners, and activation specifications
- Transaction support
- Security role mapping

3 Problem Description

The OSGi enterprise specification does not currently include support for resource adapters.

A typical resource adapter can include various application attributes declaratively in annotations or in a deployment descriptor. While, in theory, each of these features could be implemented programmatically, enterprise developers are accustomed to writing resource adapters and deploying them to get quick and easy integration.

The problems associated with connector support are the classic ones of a container class accessing classes and resources in other bundles without having any declared dependencies on them or knowledge of what the bundles contain. An extender or observer agent needs to be responsible for recognizing and accessing deployed resource adapter archives. Part of this involves being able to process annotations (for 1.6 connectors), a problem that is being discussed in RFP 135 Annotation Processing.

The agent must be able process the classes and resources in the archive and initialize them in accordance with the connector architecture. The adapter classes also need to be integrated with transactions, security and pooling mechanisms that exist in the container. The container must also manage and control the life cycle of the adapter classes.

Directionality support is dependent upon the resource adapter, but inbound interactions may need to be considered specially.

Since the resource adapter may also have a CCI that is consumable by clients (typically posted in JNDI) there must also be a standard way to make the resource adapter classes available as services.

4 Use Cases

4.1 Transactional Pooled Data Sources

A data source of the kind exported by the JDBC service is not normally expected to support connection pooling and the properties and settings that go along with pooling connections. In a container environment, data source connections are expected to be registered as XA resources in the global transaction so that all database operations on that connection are atomic with respect to the active JTA transaction context. A connector for these data sources will enable them to be managed or proxied by a container, with the container able to vend out connections according to the relevant container-level data source settings.

4.2 Messaging Systems

Messaging and queuing systems, such as JMS, are typically bound to transactions so that they can manifest ACID characteristics in their operations. When a container supports a message queue, for example, it often makes guarantees that messages are queued/dequeued atomically. Containers want to support container configuration settings for the messaging infrastructure they offer to applications and integrate the messaging middleware with the transaction system.

4.3 Non-relational Access

Many enterprise applications use resource adapters to access non-relational data, such as XML and other data formats. The ability to conveniently access legacy EIS is critical to many corporate IT applications. The resource

adapter provides a standard API to non-standard data sources, opening up a path from new/modern application code to an existing outdated data store, without having to code to the proprietary data store API.

A resource adapter for a given EIS is installed in the framework. An application accesses the EIS through its CCI, that it looks up in the service registry. When the application is finished, the connections are closed and the adapter resources are cleaned up.

4.4 Non-managed applications

Some applications use resource adapters in non-managed environments, with no controlling container. Such an application should be able to look up the CCI for a given resource as a service and invoke it to access the resource. The usual restrictions will apply, such as the lack of integration with transaction or security services.

4.5 Local Transactions

Resource adapters may not support XA transactions, and hence may not provide XAResource implementation. For these kinds of resources, clients and containers will use the local resource transaction obtainable from the connection.

4.6 Secure Access

Access to an EIS containing confidential employee information is limited to a certain limited subset of individuals. An application must be running under a given principal in order to be able to query it. When the application attempts to access the EIS, it is discovered that the current principal is not authorized (has not been authenticated) to access the data and the application receives an error in return.

5 Requirements

The requirements for connector support are listed in the following categories below:

5.1 Configuration

- CNF0001 – The standard MUST NOT require that a particular configuration API or system, including the Configuration Administration Service, be supported.
- CNF0002 – The solution MUST support deploying resource adapters as RAR files, as described in the Connector Specification.
- CNF0003 – The solution MAY allow that resource adapters be packaged or deployed in formats other than RAR files (as described in the Connector Specification).
- CNF0004 – The solution SHOULD support resource adapter components packaged as a bundle.

5.2 Integration

- INT0001 – The solution MUST allow resource adapters to export packages and services to the OSGi Framework, as well as allow OSGi-aware resource adapters to access other OSGi services.
- INT0002 – The solution MUST allow resource adapters to specify which packages to import or bundles to require from the OSGi Framework.
- INT0003 – The solution SHOULD support resource adapter components interoperating with components written to other models such as Declarative Services and Blueprint. For example, the solution could support injection of component services into Java EE components based on additional metadata associated with the application components.
- INT0004 – It SHOULD be possible for the resource adapter to be a Blueprint, Declarative Services, or other similarly-defined component

5.3 Dynamicity

- DYN0001 – There MUST be a standard programming approach that makes it possible to deploy a resource adapter to a Connector container programmatically. It may be a traditional interface registered as an object or objects in the Service Registry, it may be a programming pattern that an OSGi bundle must follow in order to ensure that it is registered as a resource adapter via reflection, or it may be something else, such as a pattern that the bundle must use to register itself on a “whiteboard.”
- DYN0002 – It SHOULD be possible for a resource adapter bundle to remain installed when its Connector Container is dynamically replaced.
- DYN0003 – Resource adapters MAY stop functioning when their backing Connector Container is stopped.

5.4 Resources

- RES0001 – The solution MUST provide the ability to access resource connection factories of arbitrary type, such as database access (JDBC datasources), messaging systems (JMS connection factories), and other EIS and ERP systems.
- RES0002 – Outbound communication MUST be supported when the resource adapter supports it.
- RES0003 – Inbound or bidirectional communication MUST be supported when the resource adapter supports it.
- RES0004 – The solution SHOULD include support for accessing the CCI or other client-facing interfaces via a service for a given resource if the CCI or client interface exists for that resource.

5.5 Compliance

- COM0001 – A compliant Connector Container MUST NOT be impeded from also being compliant with the Connector 1.6 specification.
- COM0002 – A compliant Connector Container MUST support resource adapters implemented to the Connector 1.0, 1.5 and 1.6 specifications (and any subsequent releases that are backwards compatible with any of these versions).

- COM0003 – The Connector Container design MUST NOT require an OSGi Execution Environment greater than that which satisfies the signatures of the Connector 1.6 specification.

- COM0004 – A Connector Container MAY provide additional aspects of the technology that are required for resource adapter support to be properly integrated into an OSGi framework but MUST NOT make any syntactic changes to the Java interfaces defined by the Connector 1.6 or any previous Connector specification releases.

5.6 Environment

- ENV0001 – Resource adapters MUST be supported in managed environments, when a full-featured container is providing the backing support for integrating the resources with other well-known and system level services (such as security).

- ENV0002 – Resource adapters MUST be supported in non-managed environments. Client applications may access a resource directly through its CCI. When the services a resource depends upon are not present in the system then the resource adapter might not be supported.

- ENV0003 – Conditions that do not support a given adapter MAY cause a failure at install-time, or they MAY cause an error at run-time or invocation-time.

- ENV0004 – Resource adapter services SHOULD also contain defined and standardized service properties that declare what the resource adapter capabilities are.

6 Document Support

6.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

[3]. J2EE Connector Architecture Specification, v1.5, Sun Microsystems, Nov 2003.

[4]. Java EE Connector Architecture Specification, v1.6, Sun Microsystems, Dec 2009.

[5]. Optional Package Versioning, Sun Microsystems, 2002. *http://java.sun.com/j2se/1.4.2/docs/guide/extensions/versioning.html*

6.2 Author's Address

Name	Mike Keith
Company	Oracle
Address	45 O'Connor Street, Suite 400, Ottawa, Ontario, Canada
Voice	+1 613 288 4625
e-mail	michael.keith@oracle.com

6.3 End of Document