|  |  |
| --- | --- |
|  |  |
| OTSS Integratin with OTCS (using Java Connectors) |
|  |  |

Table of Contents

[1 Document Control 2](#_Toc423363024)

[1.1 Version Control 2](#_Toc423363025)

[2 Why Java Connectors for this solution? 3](#_Toc423363026)

[3 Eclipse Java Development 4](#_Toc423363027)

[4 Stream Serve Project Development 5](#_Toc423363028)

[5 DC Package 8](#_Toc423363029)

# Document Control

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Issued | Author | Description |
| 0.1 | 29th JUNE 2015 | Krishna Chaithanya Kandala(OpenText Corp) | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Why Java Connectors for this solution?

As we all know the Integration of stream serve with content server happens via ELS Archiving.

But it has its own limitations like, we can’t pass more than one parameter as input to content server. It takes only parent ID with username and password.

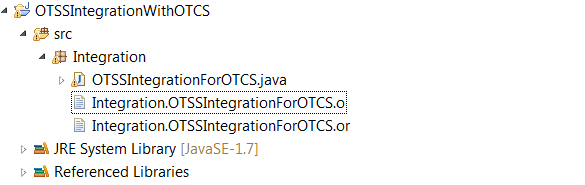
**Problem:** What if we need to pass more input parameters to content server? Sometimes we may force to pass more input parameters depending on the requirement, one such requirement is creating the revisions for the PDF file until new customer is being stored in the content server that means storage is dependent on his customer ID, and whenever the new customer comes new folder will be created for him depending on his customer ID)?

**Solution:** Java Connectors

**Inputs required:** An URL from Content server team where it might be accepting few more parameters when compared to ELS archive (No need to worry about the web service creation it will be provided to you by OTCS team and what other input it is required to be passed before you call URL)

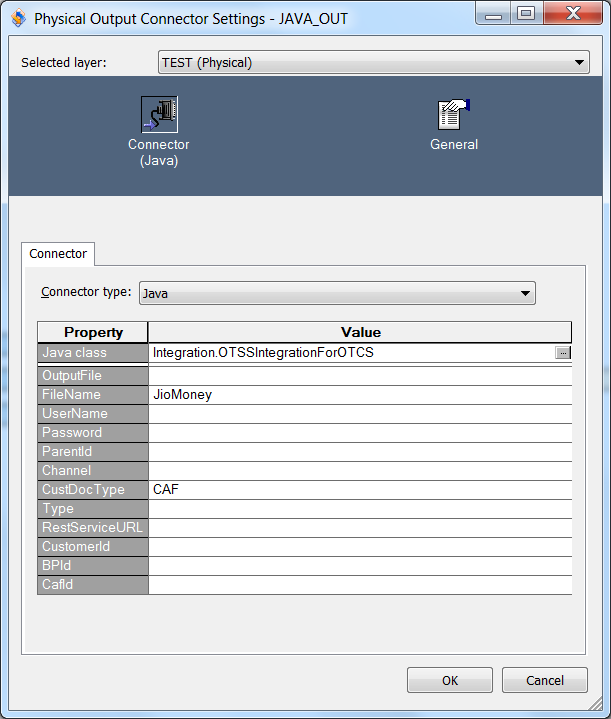
# Eclipse Java Development

Before we go for the design center project where we start out development for the project, we have to concentrate on below points for any java connector related project.

1. Install Eclipse
2. Create a sample java project
3. With 3 files
   1. OTSSIntegrationForOTCS.java
   2. Integration.OTSSIntegrationForOTCS.o
   3. Integration.OTSSIntegrationForOTCS.or
4. 
5. Make sure the Java file name and .o, .or file names to be identical like above
6. Once the project structure is ready,
7. Implement the java connectors as mentioned below.
8. 
9. Once the project is error free create or export a jar file from the eclipse and save it in the disk.

# Stream Serve Project Development

Follow below steps while design the project using design center,

1. Now you have 4 files in your hand
2. 3 eclipse files + one jar file
3. Copy .O &.OR files to below location.
4. C:\Program Files (x86)\OpenText\StreamServe\Applications\StreamServer\5.6.1\Tools\System\Java
5. Open the design center, import the jar file in you resource set like below
6. 
7. “JavaConnectorsForcontentServer.jar” is the jar name that we exported from eclipse and imported in to the resource set of design center all other remaining jars are supporting jars for the project, to give support of calling the web service from the project.
8. Attached the referenced libraries for eclipse and for design center project.
9. Like any other sample project, Saving a pdf file to disk drive just create one project, but create the platform like below
10. 
11. Whatever the field values you see in the above box, all are created using .O file that you created before.
12. 
13. The above fields are from run time environment, because of .OR file
14. Once the project is ready, Export the project and deploy the project in control center
15. Enable Java Configuration on the application that you created in control center
16. Choose the vendor to ORACLE.
17. Start the application

# DC Package

Attached a DC package + Eclipse Java project.



**Note that referenced libraries or jar files attached here are chosen by me on the basis of java project implementation, it depends on the java project coding.**

**Or you can get the jar files from those who developed the content service web service as they have used it for testing while storing the document to OTCS☺**