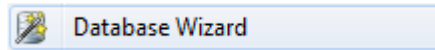


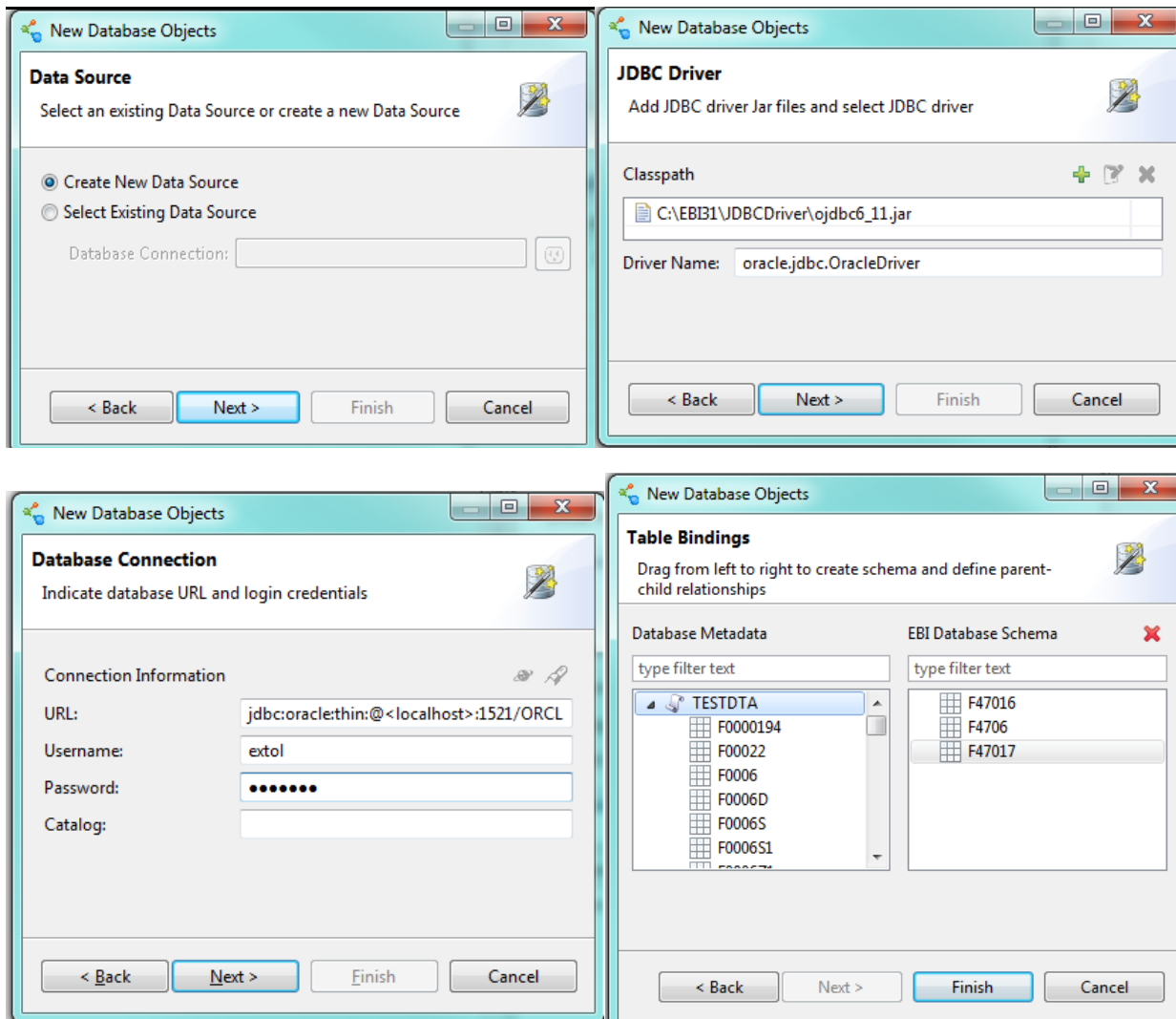
EBI 3.1 Outbound Process

1 DATABASE WIZARD



The wizard will use an existing Data Source and create a Database Adapter and Database Schema for EBI 3 users.

File > New > Database Wizard



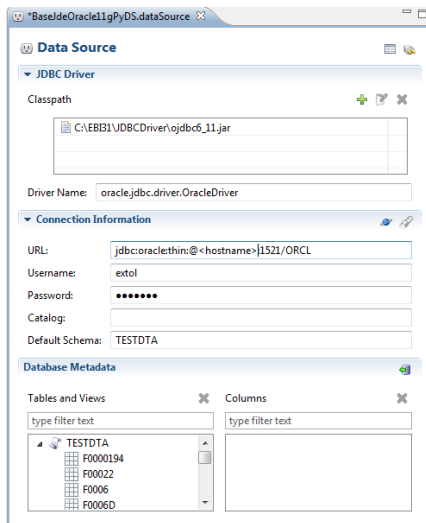
The wizard consists of four steps:

- Data Source:** Select an existing Data Source or create a new Data Source. Options: Create New Data Source, Select Existing Data Source. Database Connection: [text field].
- JDBC Driver:** Add JDBC driver Jar files and select JDBC driver. Classpath: C:\EBI31\JDBCDriver\ojdbc6_11.jar. Driver Name: oracle.jdbc.OracleDriver.
- Database Connection:** Indicate database URL and login credentials. Connection Information: URL: jdbc:oracle:thin:@localhost:1521/ORCL, Username: extol, Password: [masked], Catalog: [text field].
- Table Bindings:** Drag from left to right to create schema and define parent-child relationships. Database Metadata: TESTDTA (F0000194, F00022, F0006, F0006D, F0006S, F0006S1, F0006T). EBI Database Schema: F47016, F4706, F47017.

1.1 DATA SOURCE BaseJdeOracle11gPyDS.dataSource

The wizard will create this object for you. This object is what will connect to the Data Source using a JDBC Driver and the credentials to connect to the Database.

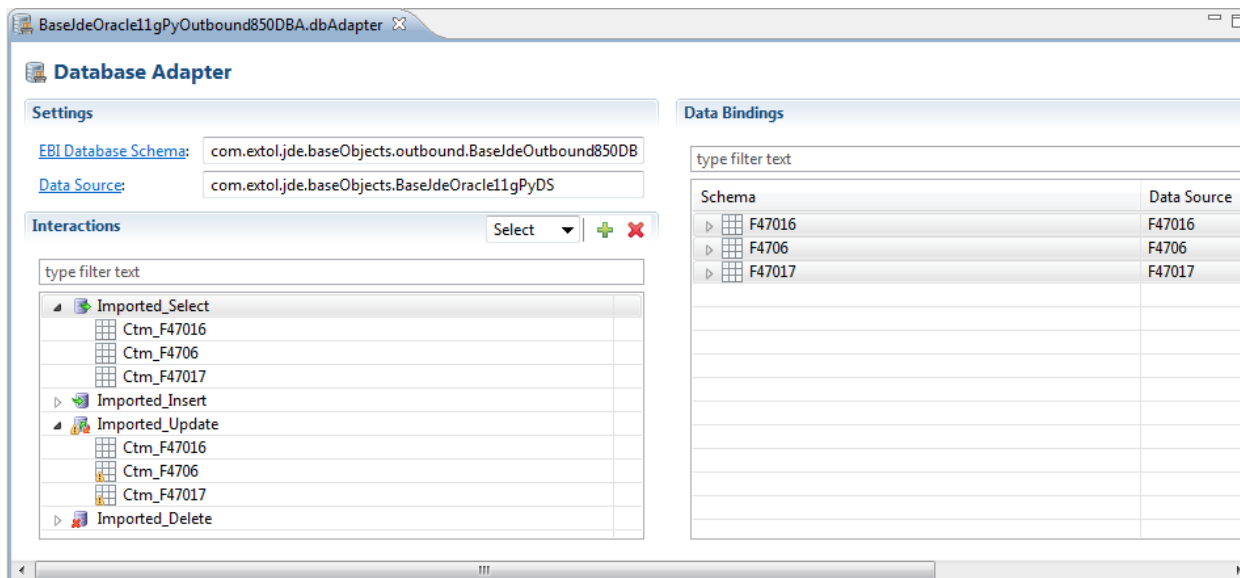
File > New > Data Source



1.2 DATABASE ADAPTER BaseJdeOracle11gPyOutbound850DBA.dbAdapter

The wizard will create this for EBI users. This object allows EXTOL to select the data in the Database and update the records as processed. We are selecting where SYEDSP is blank in this example.

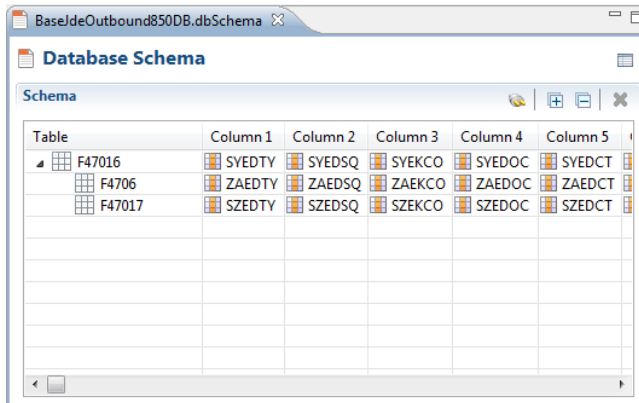
File > New > Database Adapter



1.3 SOURCE SCHEMA

The wizard will create the Database Schema for EBI users. The Database will be used as the source schema in our Ruleset.

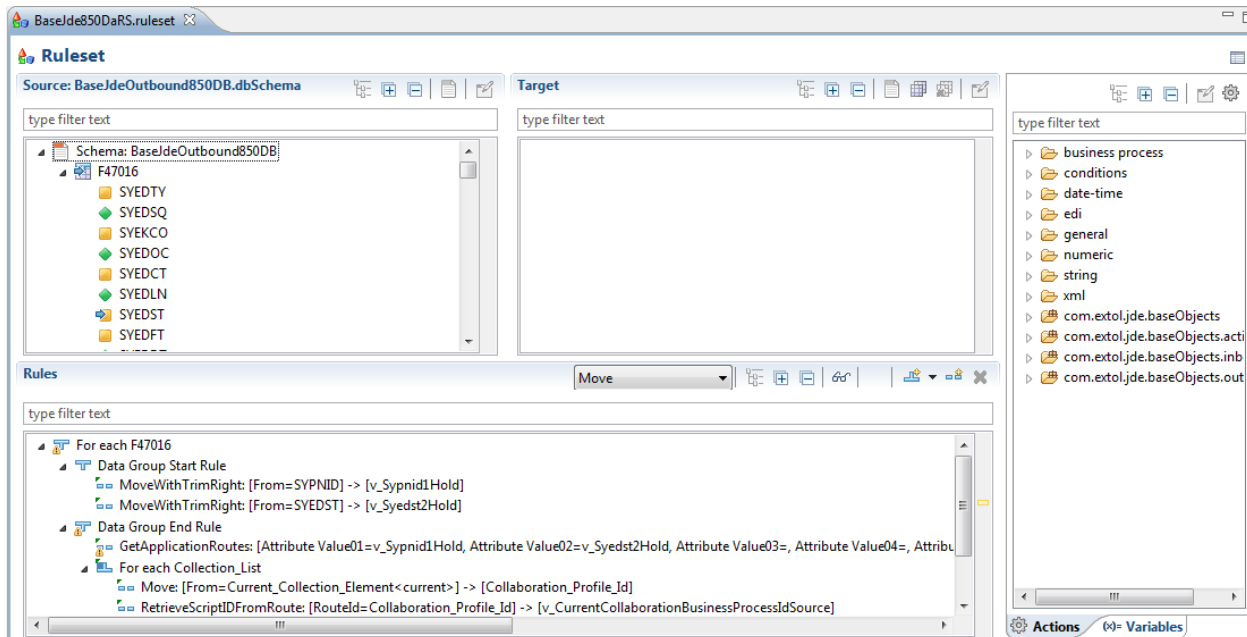
File > New > Database Schema



2 DATA ANALYSIS RULESET

Once the Source Schema is created EBI users can create your Data Analysis Ruleset. This object is placed in the Launcher Business Process and reads the defined database record/fields to determine the outbound routing.

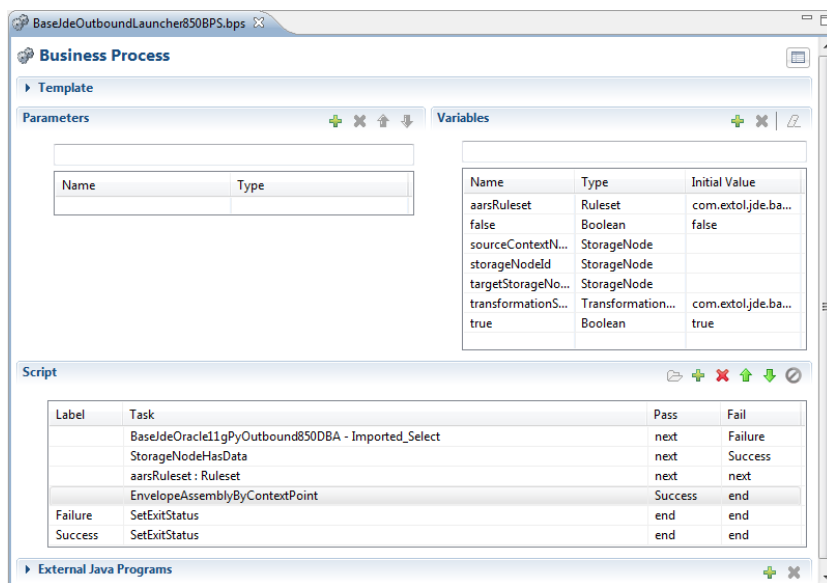
File > New > Ruleset



3 BUSINESS PROCESS LAUNCHER

This Business Process is what kicks off the Outbound Process and where the Data Analysis Ruleset is placed.

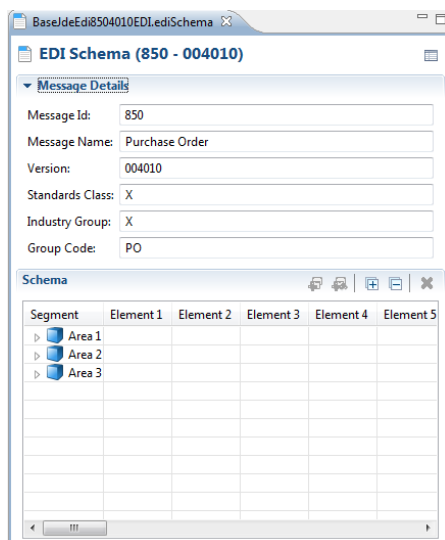
File > New > Business Process



4 TARGET SCHEMA

For this example we will be using an 850 EDI Schema. The x12 will be used as the target schema in our Ruleset.

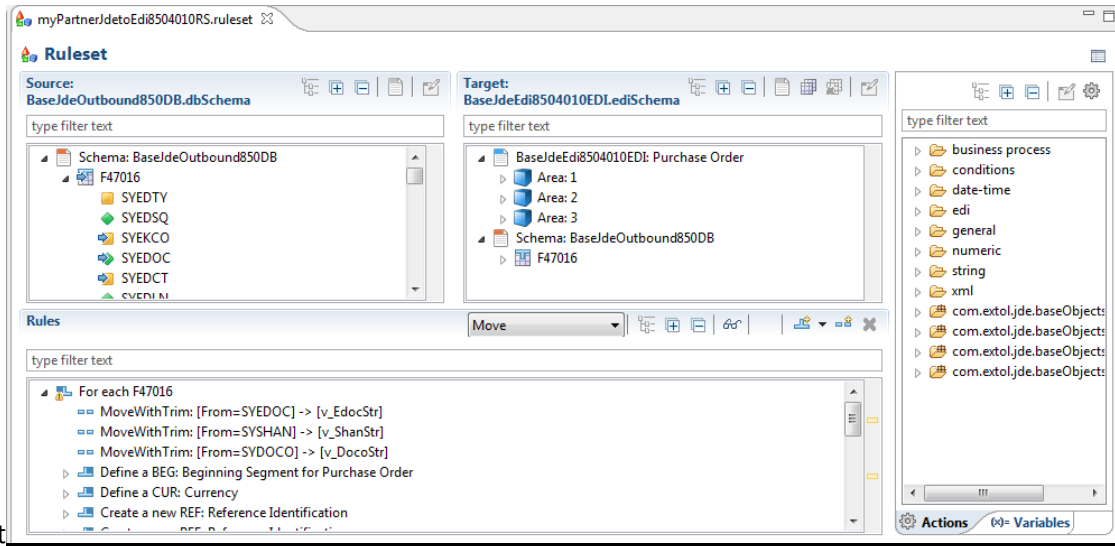
File > New > EDI Schema



5 RULESET

myPartnerJdetoEdi8504010RS.ruleset

Once both schemas are created we can create our Ruleset. This Ruleset will select data from a Database and transform the data into an EDI document.

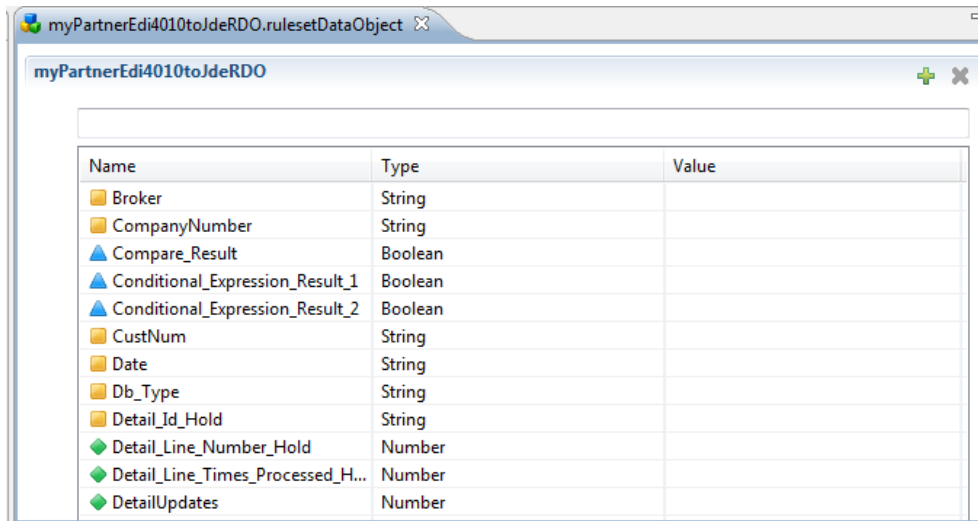


5.1 RULESET DATA OBJECTS

myPartnerEdi4010toJdeRDO.rulesetDataObject

Ruleset Data Object (RDO) is to define variables that are utilized by one or more EBI Rulesets.

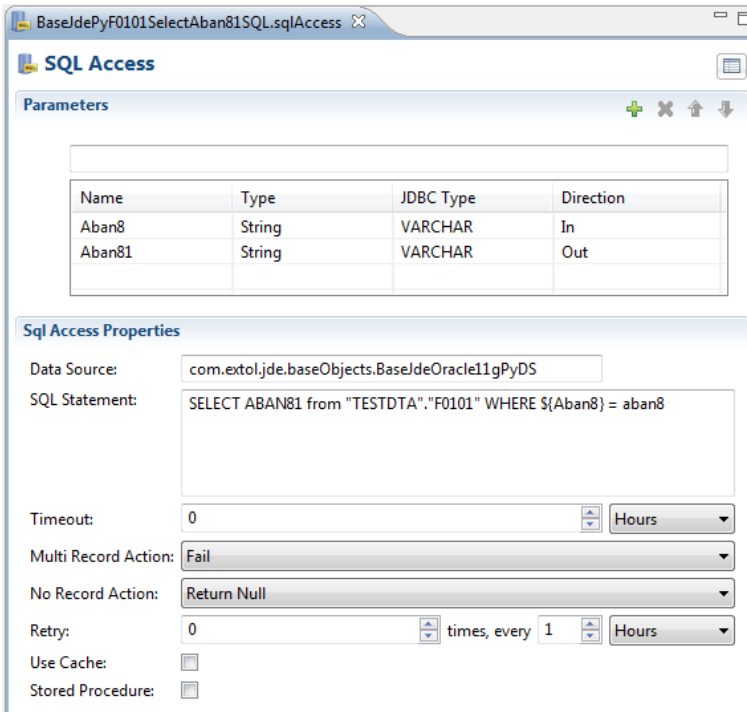
File > New > Ruleset Data Object



5.2 SQL ACCESS [BaseJdePyF0111SelectWwan8SQL.sqlAccess](#)

These are used in a Ruleset to go out to other Database tables with certain input values and return back specific fields. This example goes out to the F0101 table with ABAN8 and returns ABAN81.

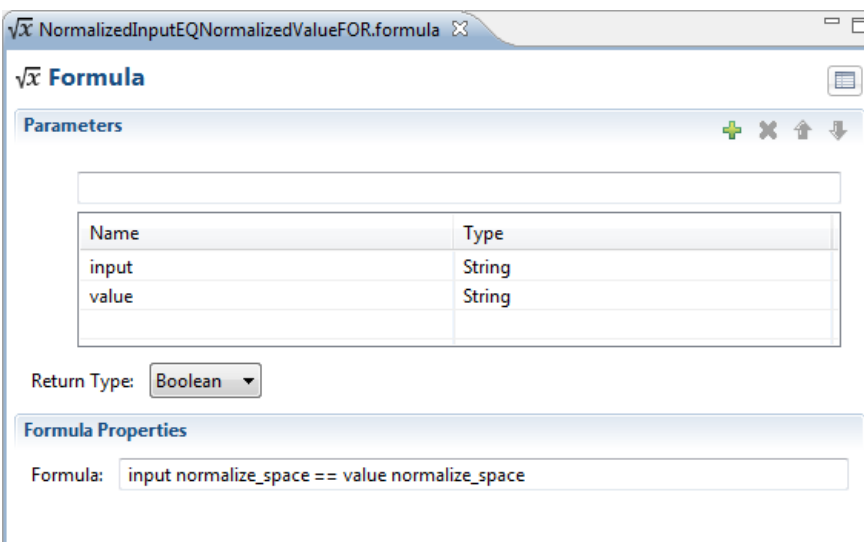
File > New > SQL Access



5.3 FORMULAS [√x NormalizedInputEQNormalizedValueFOR.formula](#)

These are used in a Ruleset to set conditions on composite rules or simple rules. This example sees if a certain value equals another value.

File > New > Formulas

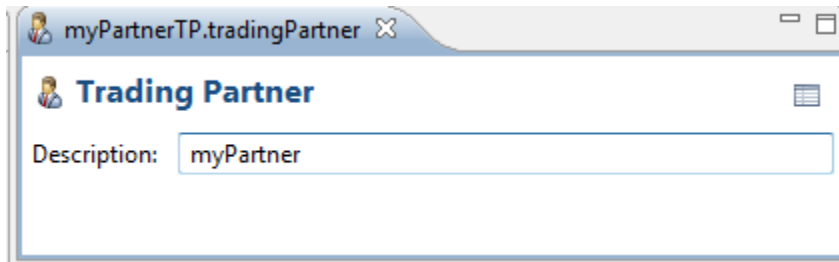


6 TRADING PARTNER

myPartnerTP.tradingPartner

This object acts as a pointer to help users indicate who their trading partner is.

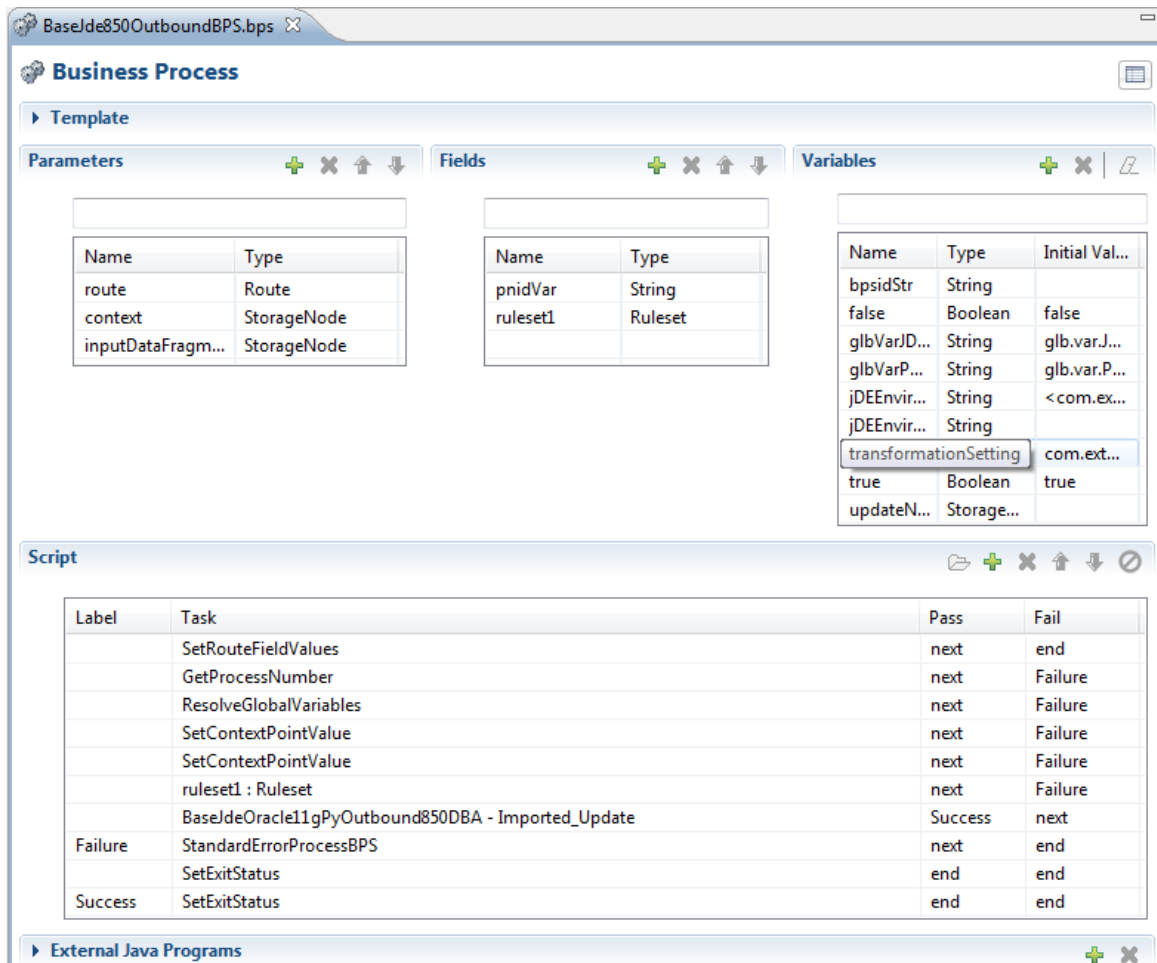
File > New > Trading Partner



7 OUTBOUND BUSINESS PROCESS BaseIde850OutboundBPS.bps

This object performs tasks on the data. In this example we have a Ruleset and Database Adapter to translate and update the data in the database.

File > New > Business Process



Parameters

Name	Type
route	Route
context	StorageNode
inputDataFragm...	StorageNode

Fields

Name	Type
pnidVar	String
ruleset1	Ruleset

Variables

Name	Type	Initial Val...
bpsidStr	String	
false	Boolean	false
glbVarJD...	String	glb.var.J...
glbVarP...	String	glb.var.P...
jDEEnvir...	String	<com.ex...
jDEEnvir...	String	
transformationSetting		com.ext...
true	Boolean	true
updateN...	Storage...	

Script

Label	Task	Pass	Fail
	SetRouteFieldValues	next	end
	GetProcessNumber	next	Failure
	ResolveGlobalVariables	next	Failure
	SetContextPointValue	next	Failure
	SetContextPointValue	next	Failure
	ruleset1 : Ruleset	next	Failure
	BaseIdeOracle11gPyOutbound850DBA - Imported_Update	Success	next
Failure	StandardErrorProcessBPS	next	end
	SetExitStatus	end	end
Success	SetExitStatus	end	end

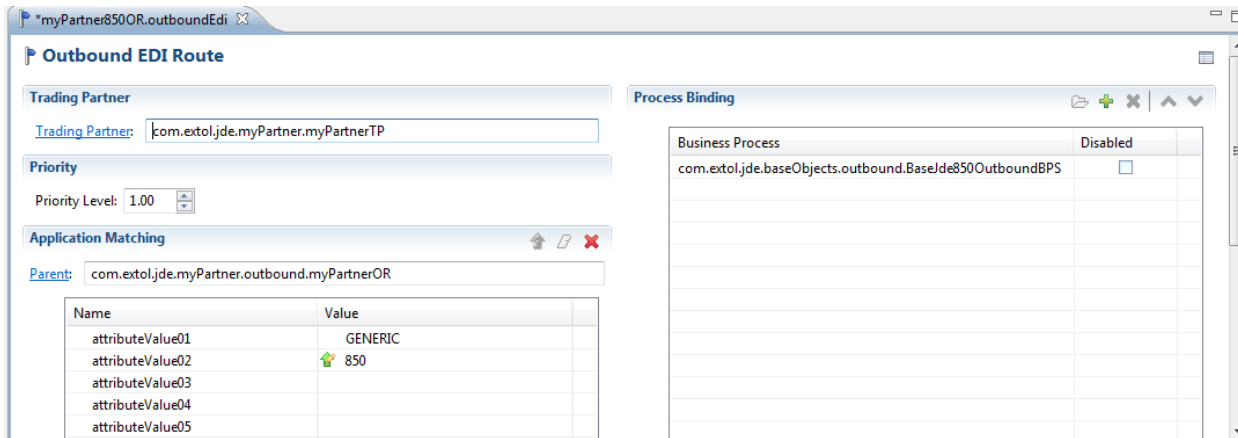
External Java Programs

8 OUTBOUND EDI ROUTE

myPartner850OR.outboundEdi

This object integrates with the Data Analysis Ruleset to determine how to properly route the outbound data. The Outbound EDI Route will kick-off the EBI Outbound Business Process. This example is searching for GENERIC in SYPNID and 850 in SYEDST. **Note:** Under 'Process Bindings' more than one Business Process Scripts can be executed.

File > New > Outbound EDI Route

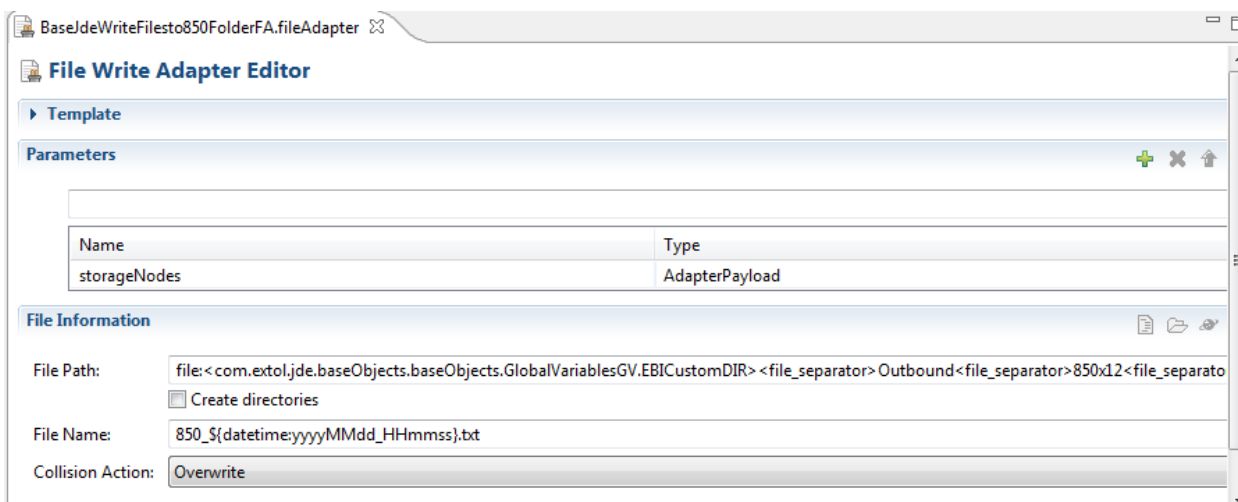


9 FILE WRITE ADAPTER

BaseJdeWriteFilesto850FolderFA.fileAdapter

This object is used to write the data to a specific file path and create your data. In this example we are creating a .txt document for our EDI data. **Note:** the File Path implements global variables for path determination.

File > New > File Adapter

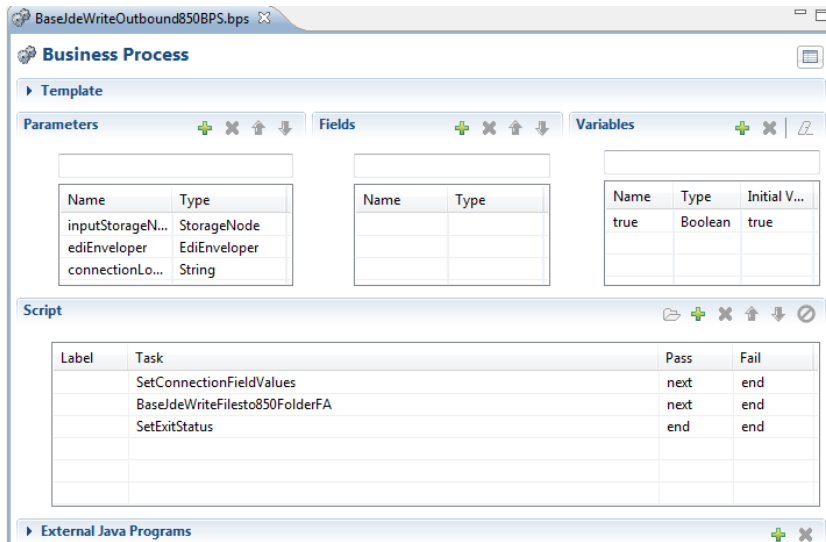


10 OUTBOUND WRITE BUSINESS PROCESS

BaseJdeWriteOutbound850BPS.bps

This Business Process writes out the x12 850 data once it has been generated. In this example we are using our File Write Adapter we just created. The Business Process can be generalized so we can use it for any outbound document; not just for the outbound x12 850.

File > New > Business Process

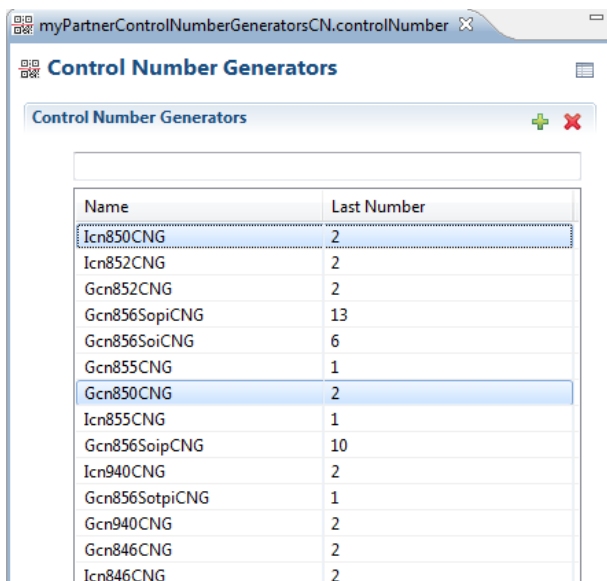


11 CONTROL NUMBER GENERATORS

myPartnerControlNumberGeneratorsCN.controlNumber

This object assigns sequential numbers to your trading partner's interchanges and group control numbers.

File > New > Control Number Generators



The EDI Enveloper will envelope the trading partner information. Once the data is enveloped, the object points to the Outbound Write Business Process and Control Number Generators. This object will launch the Business Process that will send the file to the trading partner; in this case, write out the file.

File > New > EDI Enveloper

The screenshot displays the X12 EDI Enveloper application window. The interface is divided into several sections:

- Trading Partner:** Shows the parent EDI Enveloper path and the trading partner ID: `com.extoljde.myPartner.outbound.myPartnerEdi4010OutboundENV` and `com.extoljde.myPartner.myPartnerTP`.
- Envelope Fields:** A table listing various EDI fields and their values.

Name	Value
(ISA01) Authorization Qualifier	00
(ISA02) Authorization Information	
(ISA03) Security Qualifier	00
(ISA04) Security Information	
(ISA05) Sender Id Qualifier	ZZ
(ISA06) Sender Id	SENDERID
(ISA07) Receiver Id Qualifier	ZZ
(ISA08) Receiver Id	RECEIVERID
(ISA12) Interchange Version Id	00401
(ISA14) Interchange Ack Requested	0
(ISA15) Test Production Flag	T
Element Delimiter	* (Hex value: 2a)
Segment Delimiter	~ (Hex value: 7e)
Repetition Separator	U (Hex value: 55)
Sub Element Delimiter	> (Hex value: 3e)
(GS01) Group Code	PO
(GS02) Application Sender Id	SENDERID
(GS03) Application Receiver Id	RECEIVERID
(GS07) Controlling Agency	X
(GS08) Version Release [1:6]	004010
(GS08) Industry Id Code [7:12]	
(ST01) Message Id	850
- Envelope Settings:**
 - Target Envelope: X12
 - Change Enveloping Limits
 - Enveloping Limits:
 - Interchanges in Connection: 1 (No Max)
 - Groups in Interchange: 1 (No Max)
 - Messages in Group: 1 (No Max)
 - Control Number Generators:
 - Interchange: `com.extoljde.myPartner.myPartnerControlNumberGeneratorsCN.Gcn850CNG` (Auto counter)
 - Group: `com.extoljde.myPartner.myPartnerControlNumberGeneratorsCN.Icn850CNG`
 - Message: Click to set Control Number... (checked)
- Document Settings:**
 - Encoding Character Set: UTF-8
 - End of Line Character: Carriage Return / Line Feed
 - Enable Fixed Wrapping
 - Fixed Wrap Length: 80
- Connection Process:**
 - Business Process: `com.extoljde.baseObjects.outbound.BaseJdeWriteOutbound850BPS`
 - Fields table:

Name	Type	Value
- Late Acknowledgement Setting:**
 - Enable Late Acknowledgement
 - Late Threshold: 0 Hours