Leading the **Integration** Revolution

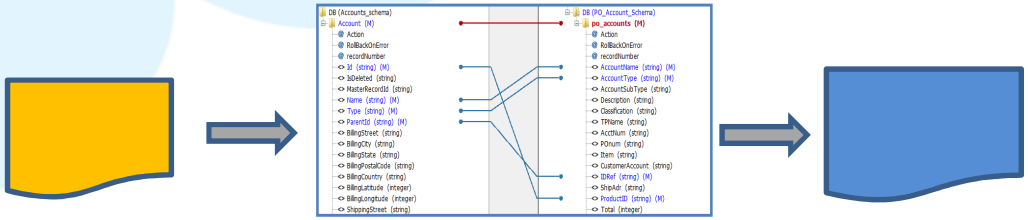
Your business problems have changed.
Why hasn't your integration solution?

ADEPTIA

Dynamic Mapping Service API

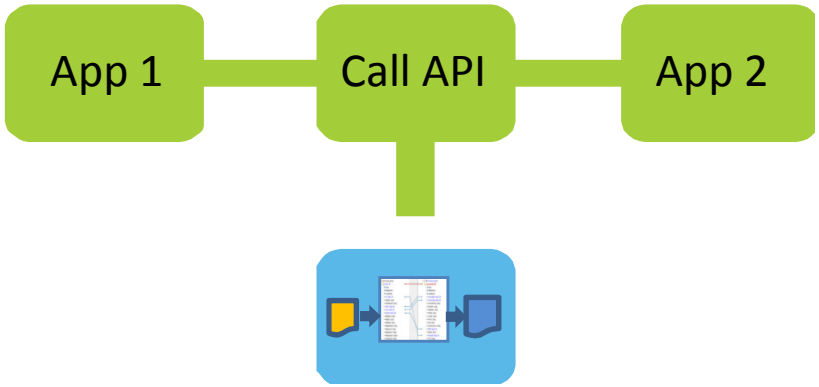
Use Case: Dynamic Mapping Service

Step 1: Create a dynamic process that can take any source data, executes the appropriate Mapping to convert or transform the source into the desired output.



Any Source data | Convert | Output

Step 2: Wrap this service into a REST API that can be called by applications automatically.

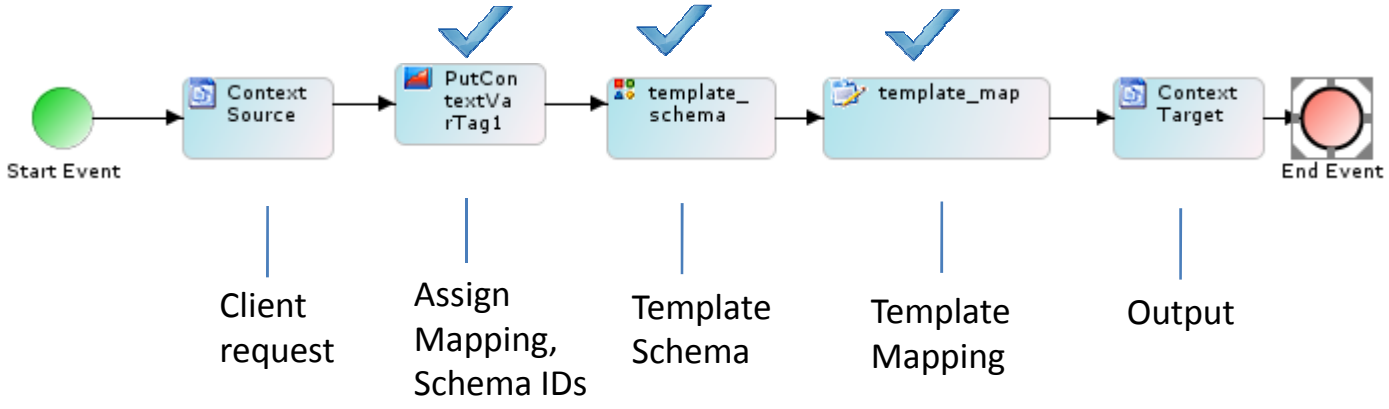


Step 1: Create a dynamic process (example 1)

Refer to the Use Case video on how to build the process flow.

Here's an example of a dynamic process template.

Base process

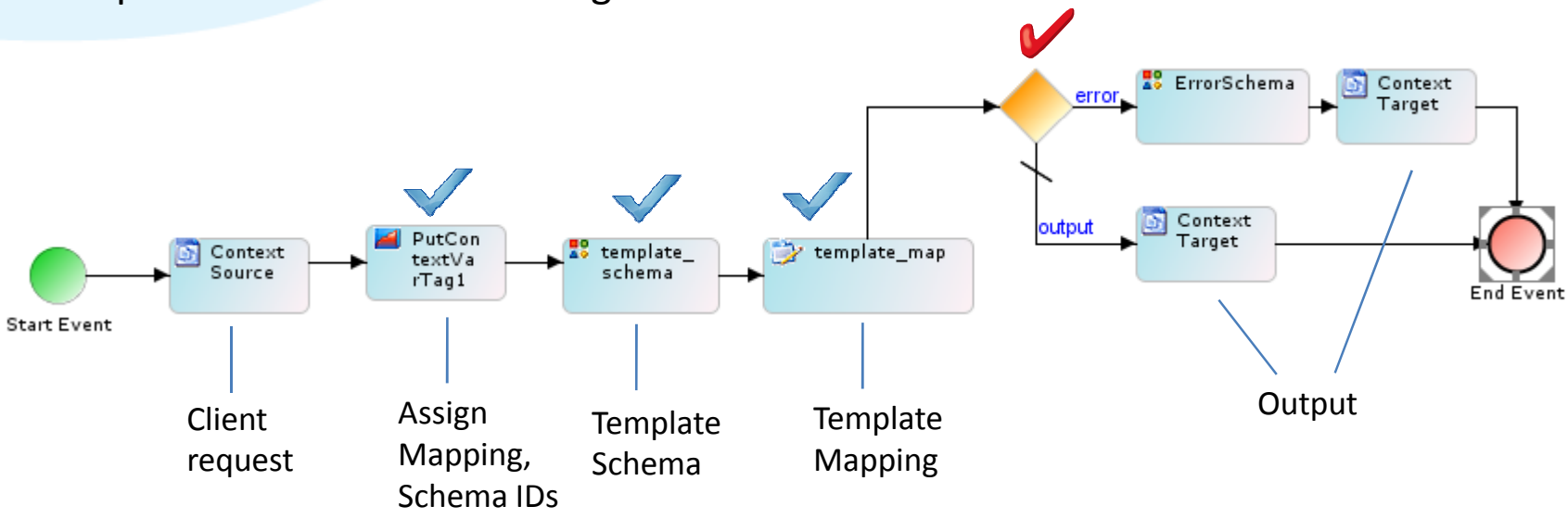


Step 1: Create a dynamic process (example 2)

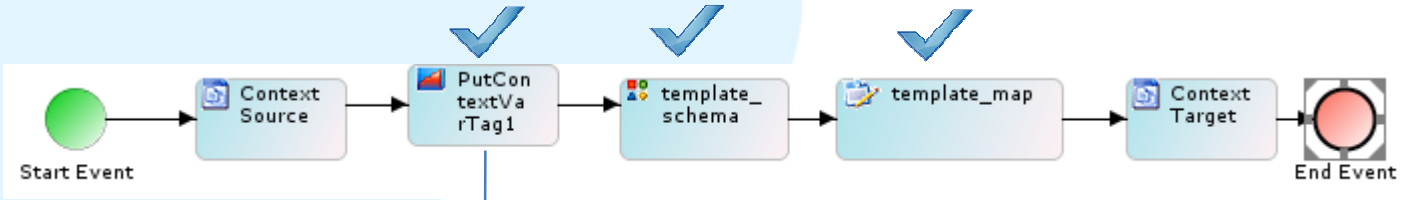
Refer to the Use Case video on how to build the process flow.

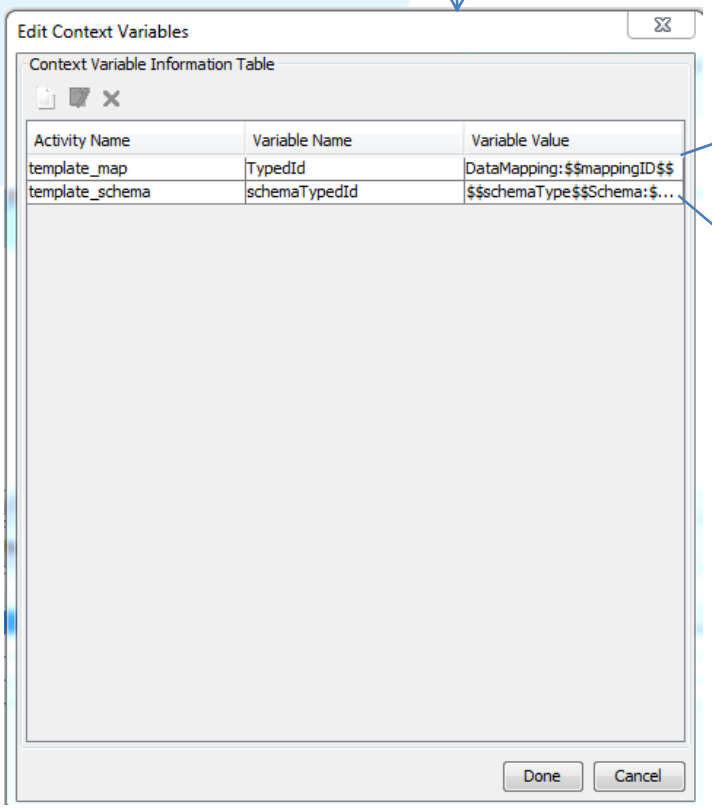
We can add Error Handling steps after the gateway and make the process extensible as per your business needs.

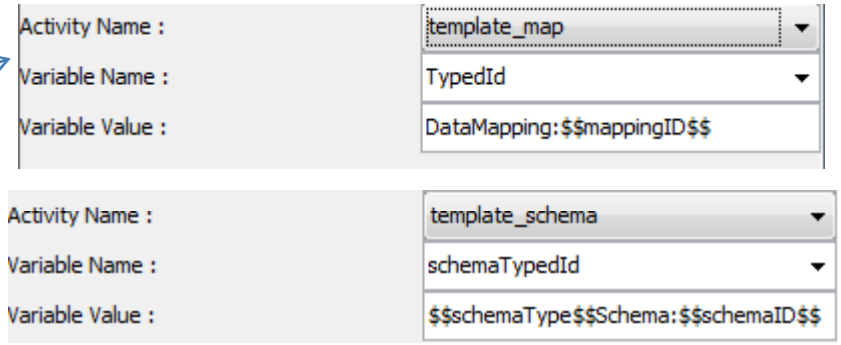
Base process with error handling



Overriding the Activity IDs in process flow

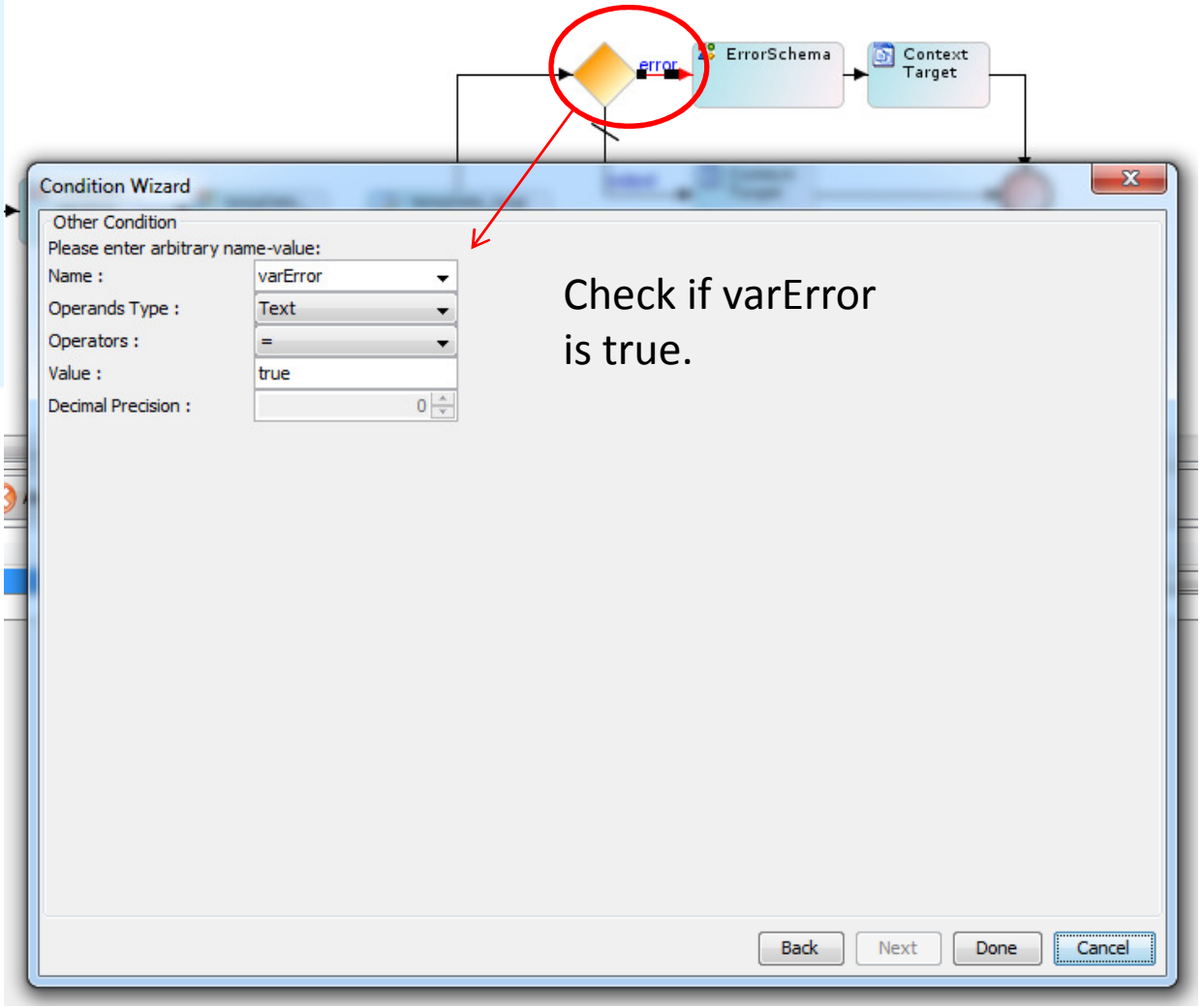






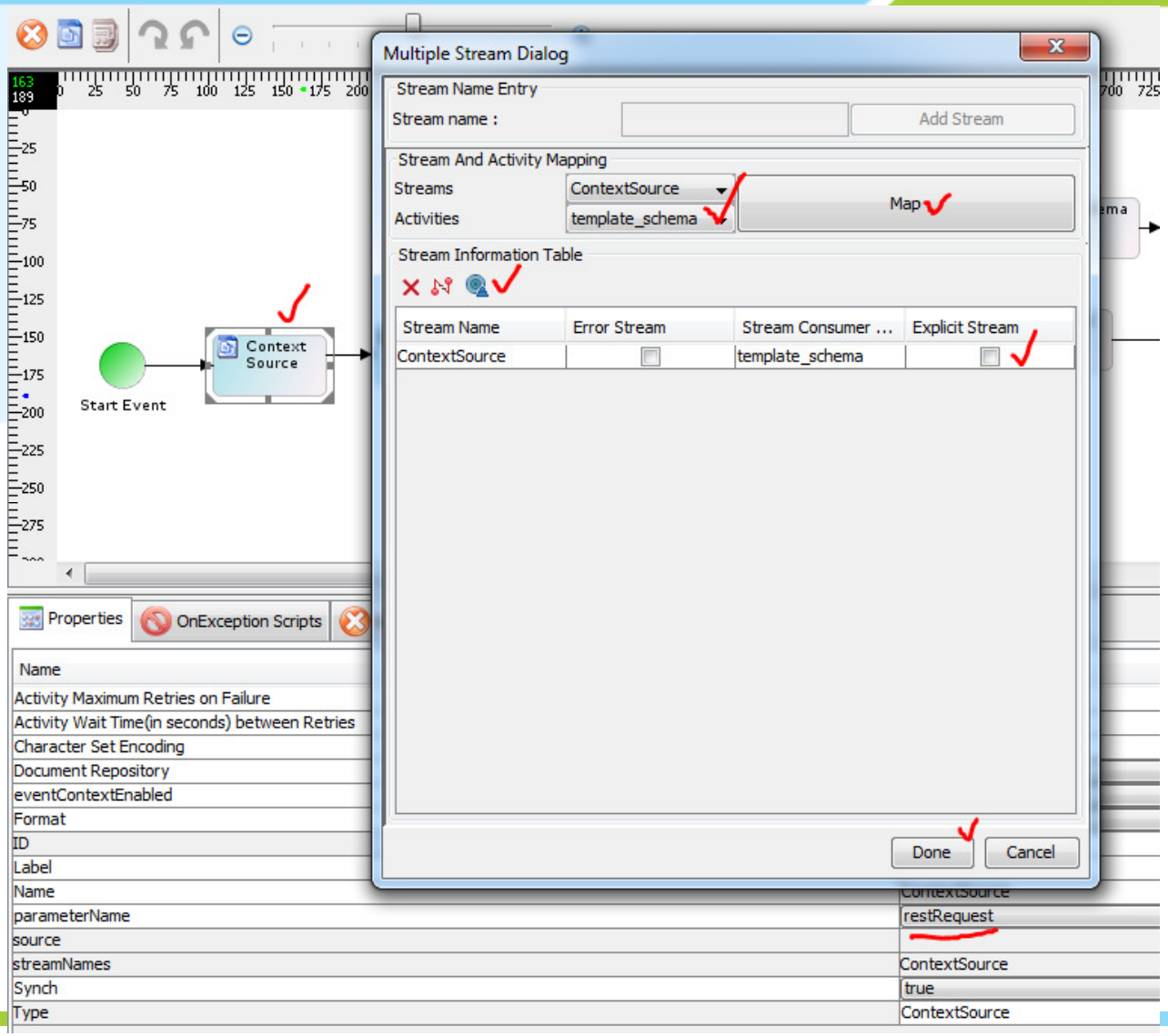
Put-Context-Var allows activity IDs to be overridden at run-time. In this flow it is overriding the “template” schema and mapping placeholder activities with actual activity IDs passed from the Web Service request.

Gateway condition to check errors

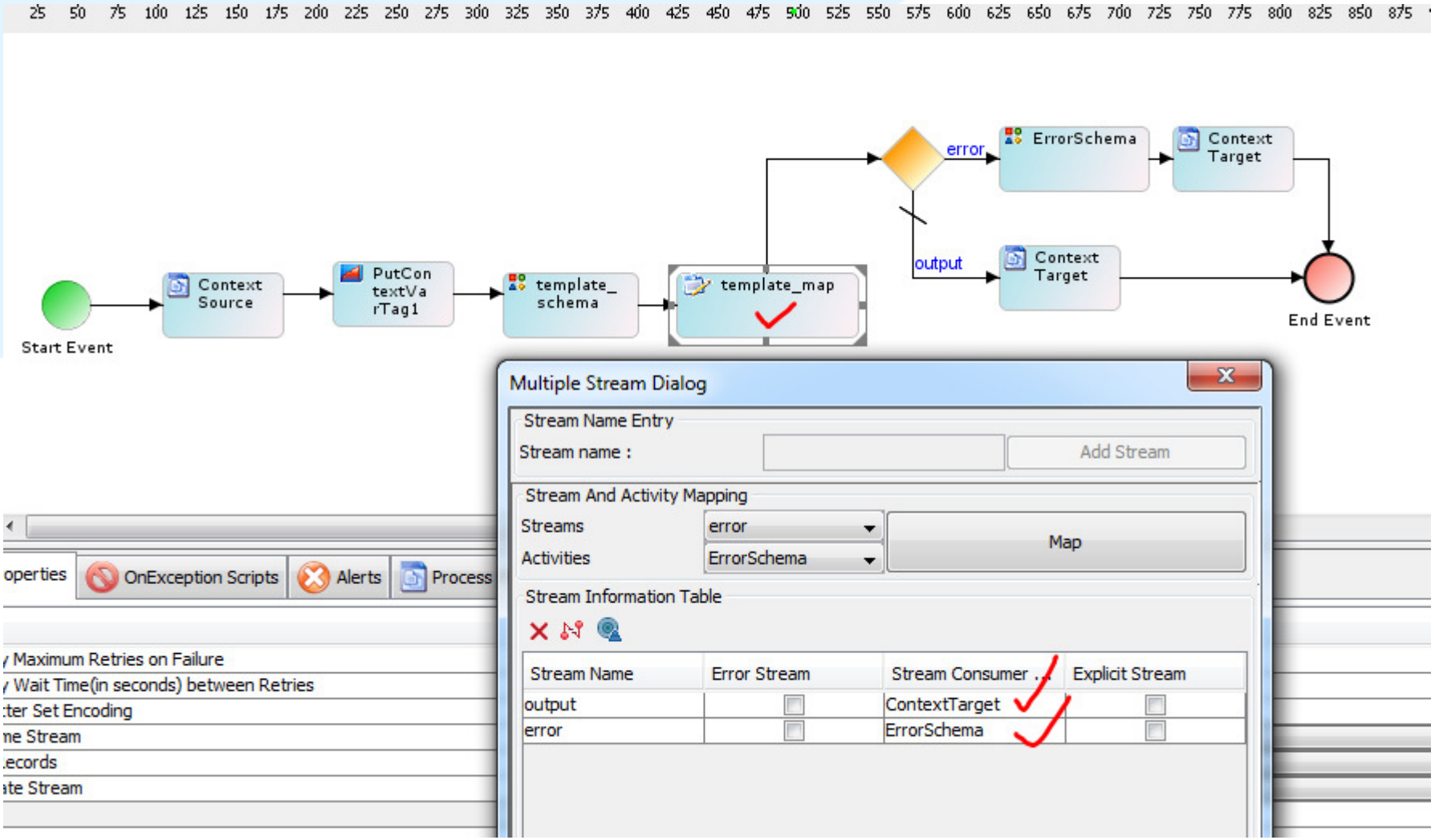


Check if varError is true.

Context Source properties and multiple stream

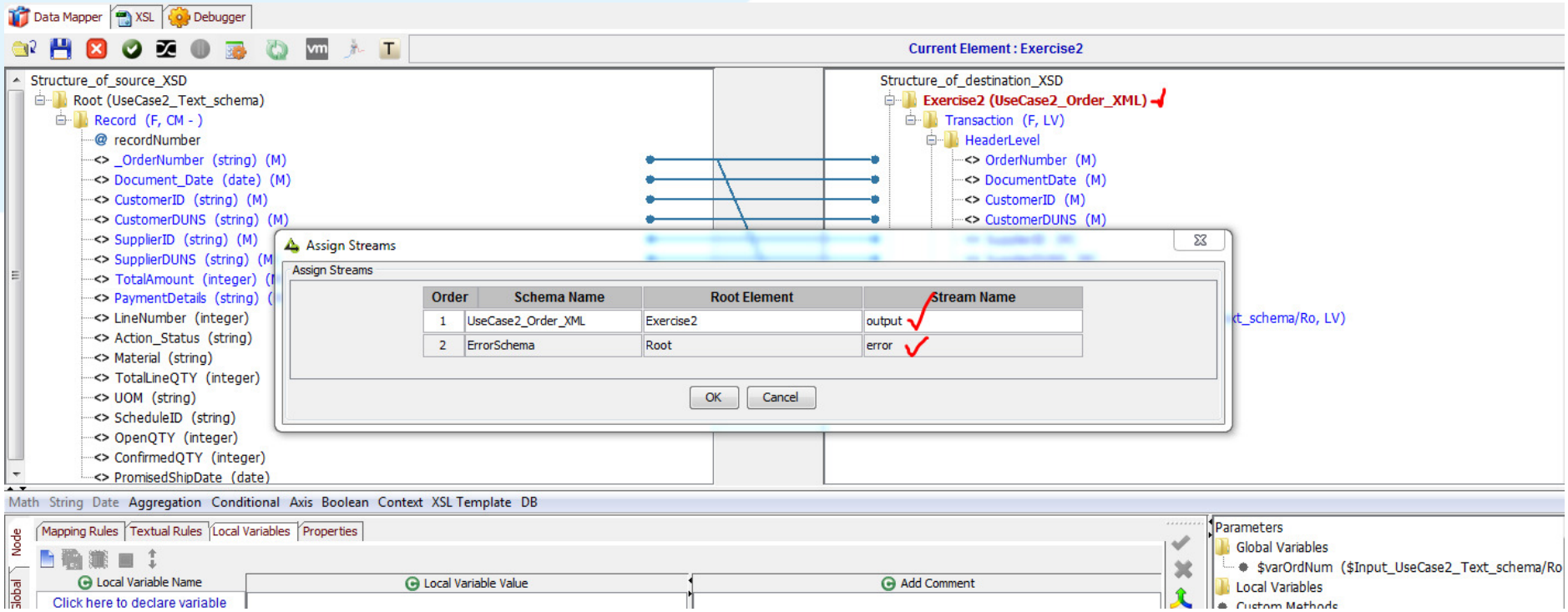


Template Map multiple stream option



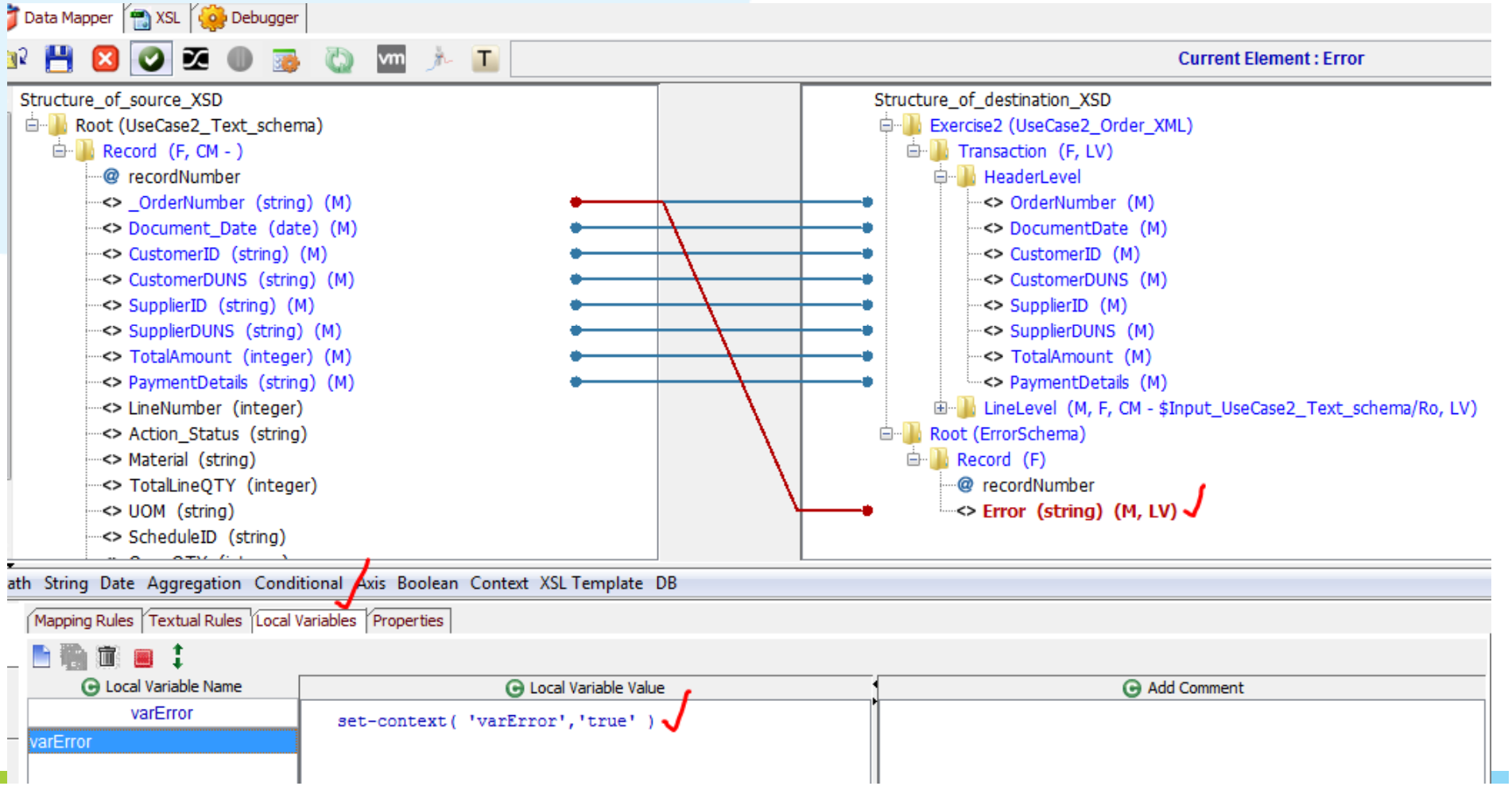
Two output streams in the Mapping

In the template and the actual map name the two output streams as ‘output’ and ‘error’

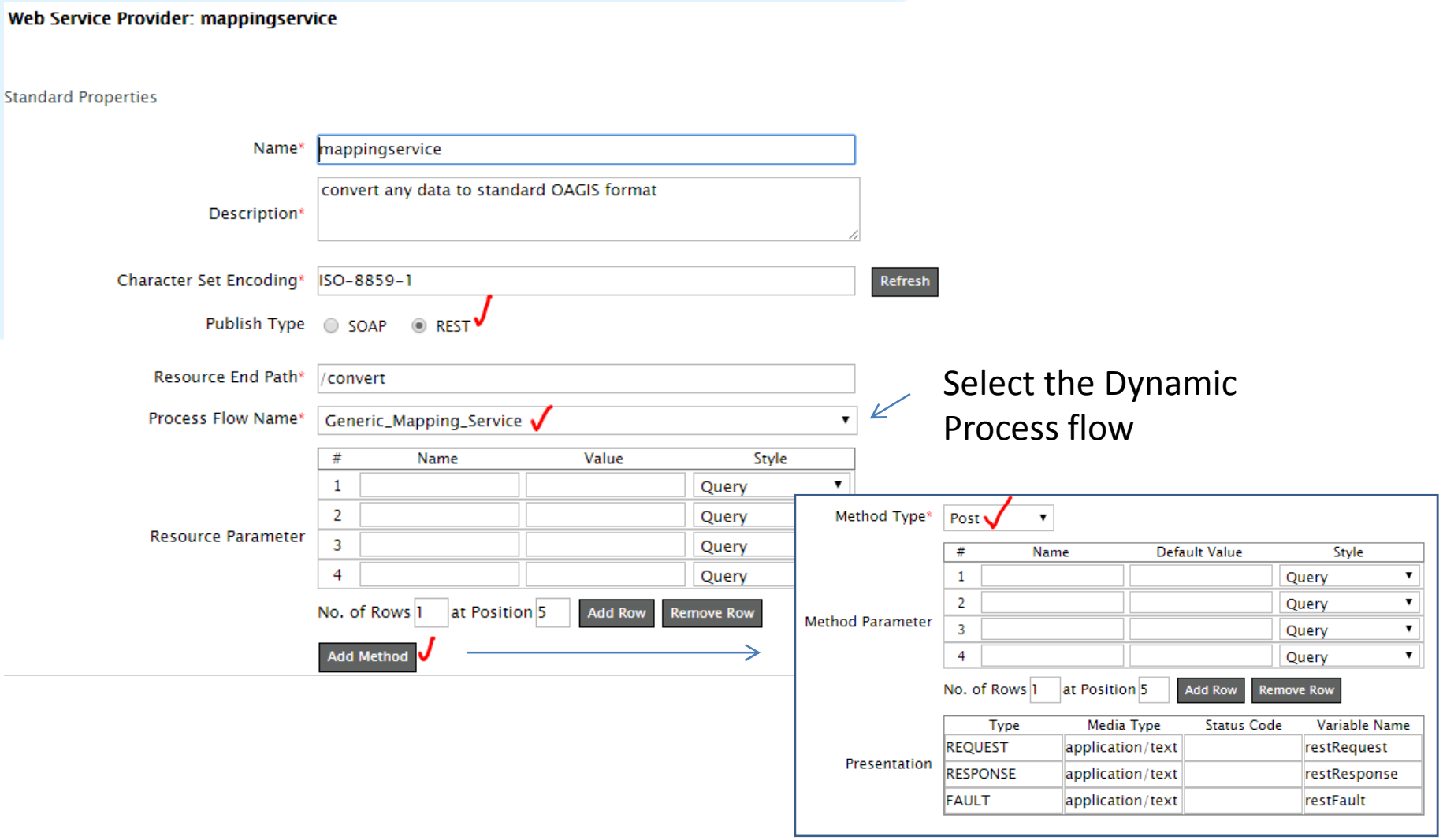


Context variable in actual Mapping

Set context variable with value as true. Use for each and apply any error rule on this schema's record node that would result in this error schema to be populated by error records at run-time.

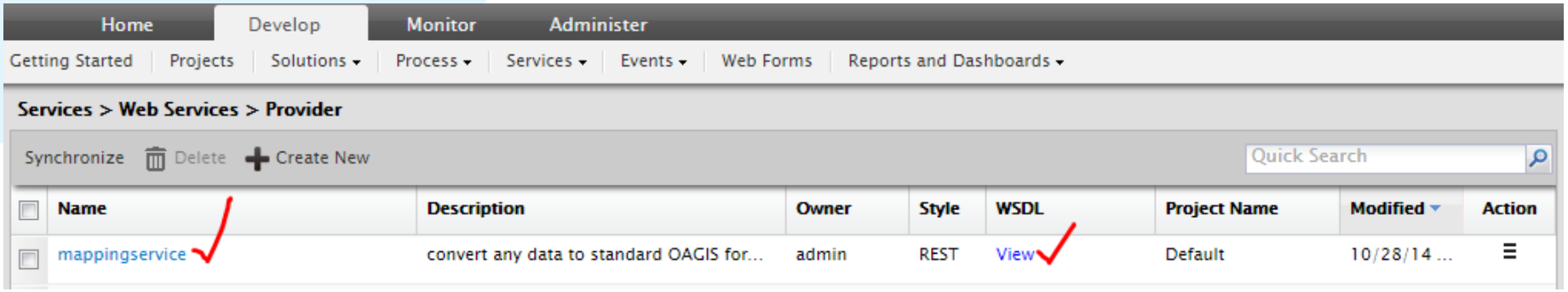


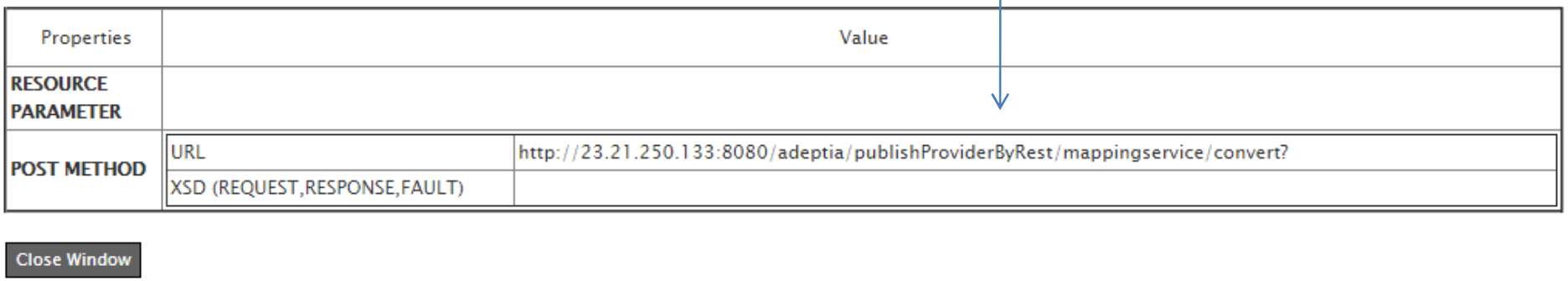
Step 2: Publish the dynamic process as a REST API



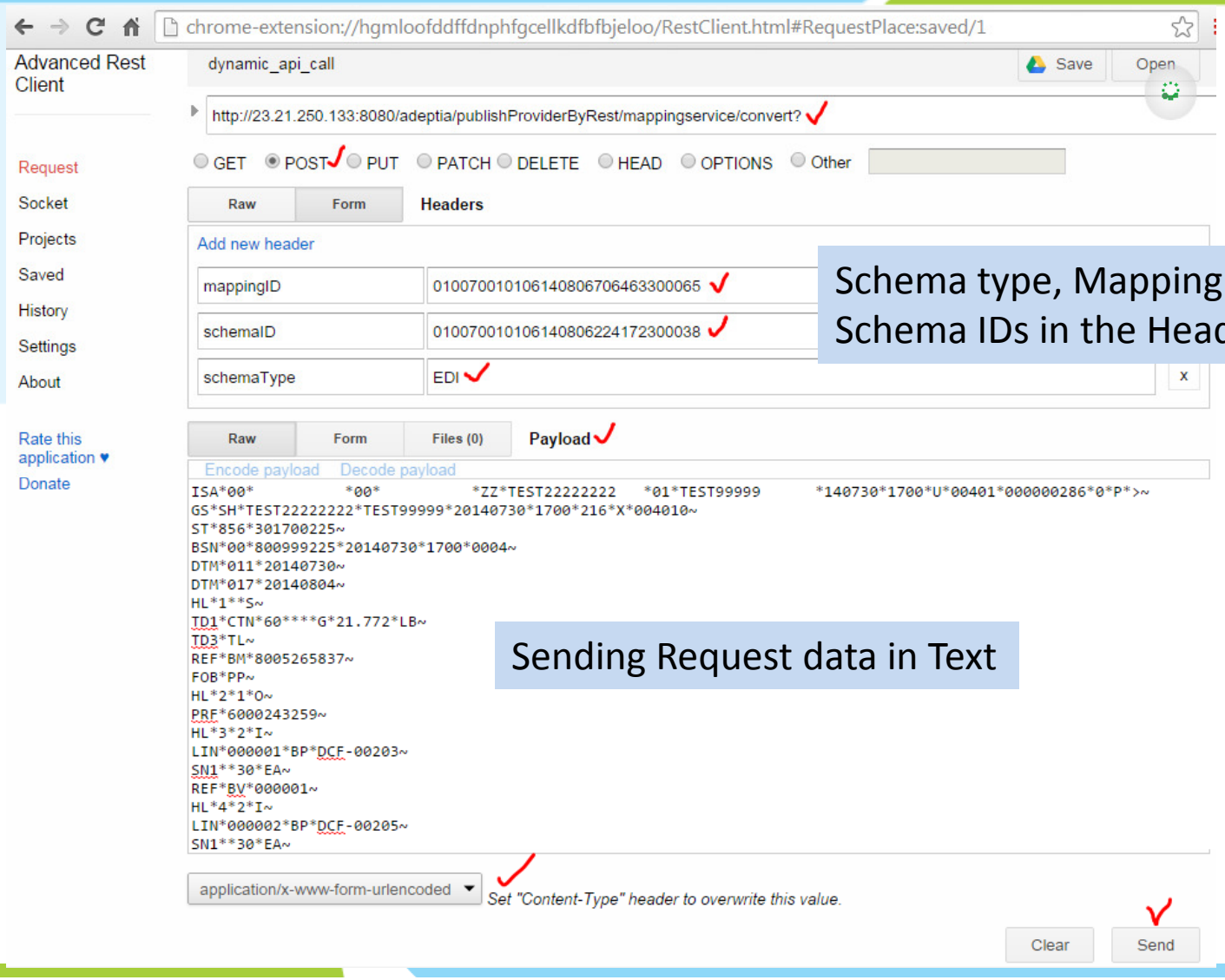
Step 2: Publish the dynamic process as a REST API

After saving the Provider go to Manage page and click on the View to see the REST endpoint. Now use this endpoint in your client application to pass the payload and the Activity IDs and get the mapping results.





Running the Mapping Service API: Request



Running the Mapping Service API: Response

Scroll to top

Status: 202 Accepted Loading time: 552 ms

Request headers:

schemaType: EDI
Origin: chrome-extension://hgmloofddffdnphfgcellkdfbfbjeloo
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36
schemaID: 0100700101061408062241723000 38
mappingID: 010070010106140806706463300065
Content-Type: application/x-www-form-urlencoded
Accept: */*
Accept-Encoding: gzip,deflate
Accept-Language: en-US,en;q=0.8
Cookie: m=open; m6=open; m7=open; m2=open; m3=open; yPos=!~0~!; JSESSIONID=ut74pbexk6fa1tgdwrtcpjiwh; JSESSIONID=g22jkqm8s5px1r7bc7iuhfkiw; lastService=WsProvider

Response headers:

Date: Tue, 28 Oct 2014 19:56:23 GMT
Content-Type: application/text; charset=ISO-8859-1
Transfer-Encoding: chunked

Request triggers the Mapping Service and sends the mapping result back as a response.

Raw | Parsed | **Response**

Open output in new window Copy to clipboard Save as file Open in JSON tab

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ns0:NotifyShipment xmlns:ns0="http://www.openapplications.org/oagis/9_4" xmlns:e2o="http://www.e2open.com/ns">
<ns0:ApplicationArea>
<ns0:Sender>
<ns0:LogicalID>TEST22222222   </ns0:LogicalID>
<ns0:TaskID>ASN</ns0:TaskID>
<ns0:ReferenceID>na</ns0:ReferenceID>
<ns0:ConfirmationCode>na</ns0:ConfirmationCode>
</ns0:Sender>
<ns0:Receiver>
<ns0:LogicalID>TEST99999      </ns0:LogicalID>
</ns0:Receiver>
</ns0:ApplicationArea>
<ns0:DataArea>
<ns0:Shipment>
<ns0:ShipmentHeader>
<ns0:DocumentID>
<ns0:ID>800999225</ns0:ID>
</ns0:DocumentID>
<ns0:DocumentDateTime>20140730</ns0:DocumentDateTime>
<ns0:Status>
<ns0:Code name="Payment">PP</ns0:Code>
</ns0:Status>
<ns0:BillOfLadingReference>
<ns0:DocumentID>
<ns0:ID>800999225</ns0:ID>
</ns0:DocumentID>
<ns0:Note type="PackCode">CTN</ns0:Note>
<ns0:Quantity>60</ns0:Quantity>
</ns0:BillOfLadingReference>
<ns0:ScheduledDeliveryDateTime>2014-08-04T00:00:00.000+0100</ns0:ScheduledDeliveryDateTime>
<ns0:ActualDeliveryDateTime>2014-07-30T14:00:27+00:00</ns0:ActualDeliveryDateTime>
```

Receiving Response data in XML

ADEPTIA

Thank you!

sales@adeptia.com