

# Community

[Ask a Question](#) [Write a Blog Post](#)[Login](#)

Sathwik Reddy G K

December 7, 2015 | 13 minute read

## ABAP Outbound Proxy

5 17 49,242

Follow

Like

RSS Feed

### Introduction

The Proxy is used as an Interface to transfer or receive from or to the ABAP system.

Depending on the direction of flow of data Proxy can be divided as

#### 1. Client Proxy:

Proxy used for transfer of data from ABAP end. The classes generated when we create proxy will be for Outbound.

We need to call the method of Proxy class to transfer the data from ABAP end.

## 2. Server Proxy:

Proxy for inbound interface is called Server Proxy.

This document briefly explains the step by step procedure needs to be followed to pass data available in an internal table in SE38 executable program to PI using Proxy. This is for Client Proxy (Outbound).

# Step By Step Procedure for Creating Proxy

**Step1:** The Proxy Service Interface has to be created by PI consultant from PI side. This will further reflect in ABAP side in T-Code "SPROXY".

**Step2:** Create the Proxy from T-Code "SPROXY".

a. The Proxy created from PI side will be reflected with following details in SPROXY

(All details are for example please refer the screen shot 1 to check where it will be available)

**Software Component:** "Example 1.0 of INV"

**Namespace:** <http://example/xyz/InvoiceDeltaLoad>

**Service Interfaces:** "S168\_InvoiceDeltaReplication\_XXX\_IB" for Inbound

"S168\_InvoiceDeltaReplication\_OB" for Outbound.

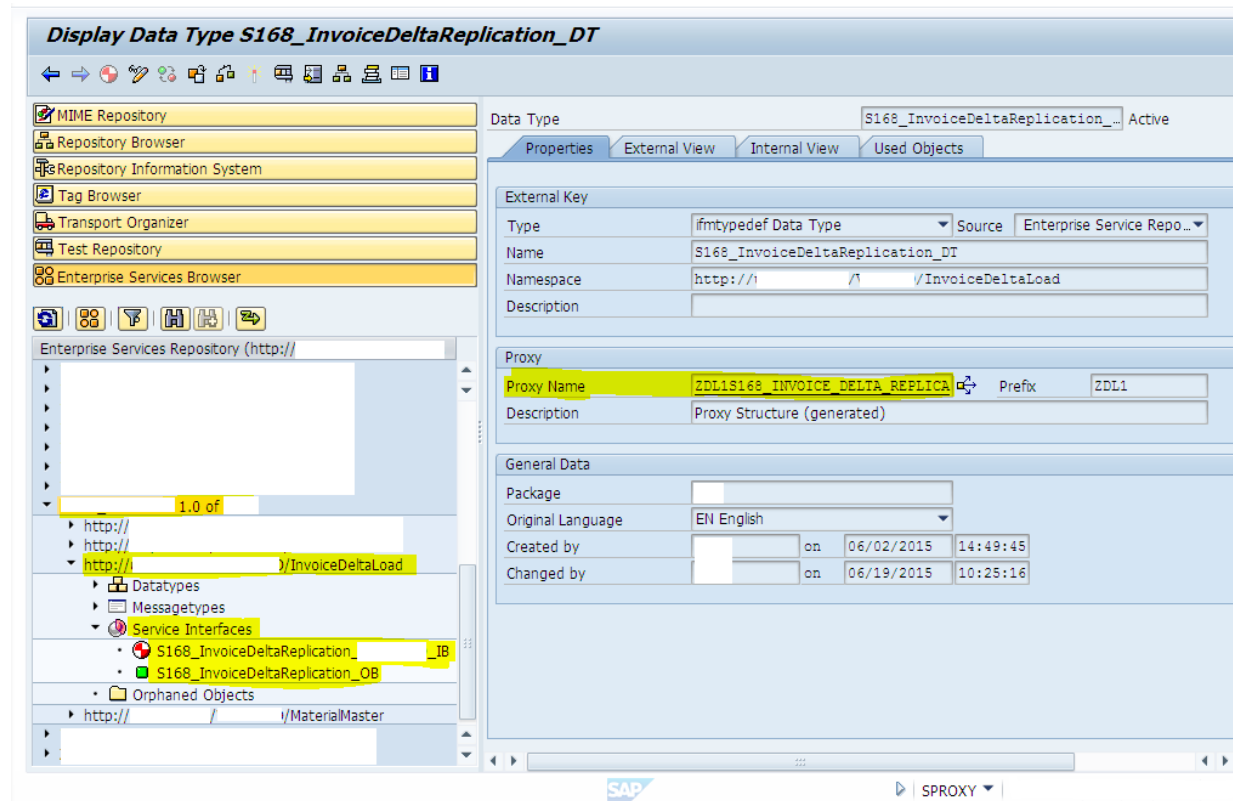


Fig1

b. Proxy Name will be available once we generate the Proxy.

To generate proxy we need to right click on Service Interface "S168\_InvoiceDeltaReplication\_XXX\_IB" for Inbound and "S168\_InvoiceDeltaReplication\_OB" for Outbound and click on "Create Proxy".

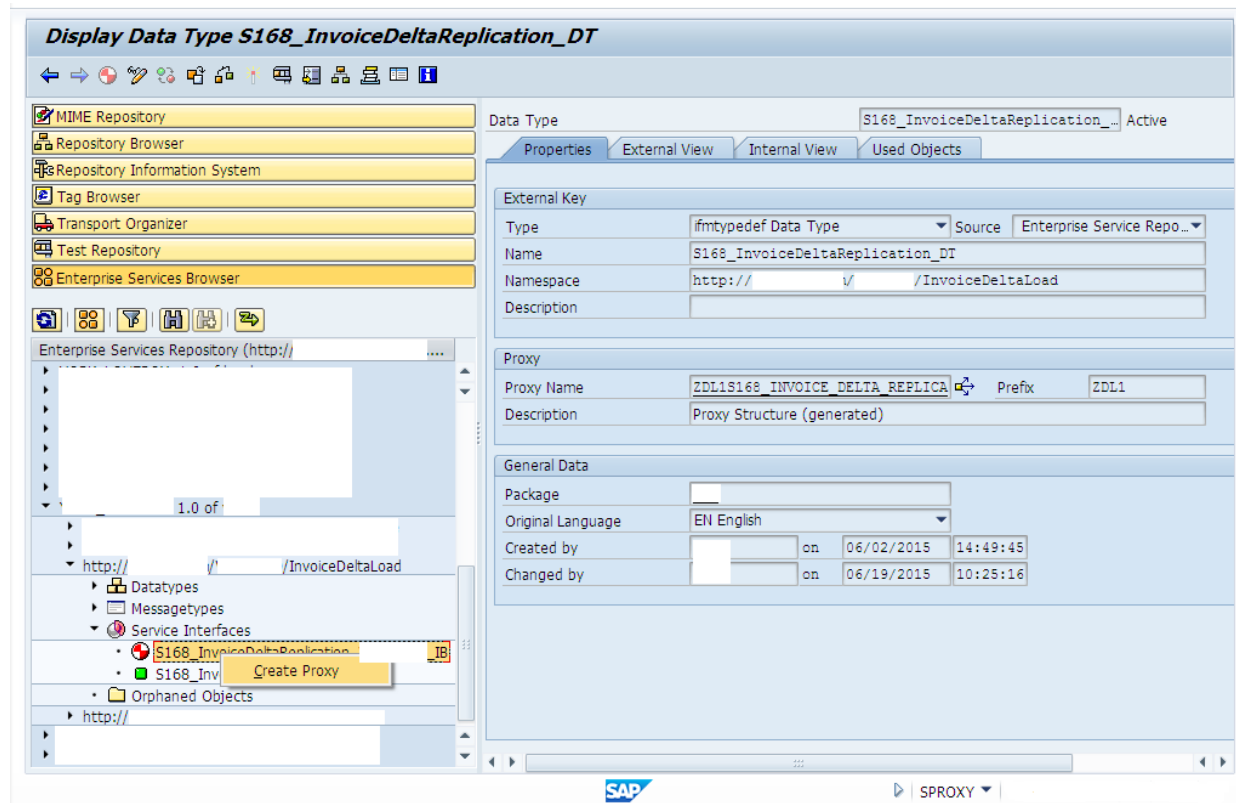


Fig 2

c. The following dialog will popup when we select “Create Proxy”.

- We need to give the “Package” and “Transport Request” to which the Proxy needs to be saved.
- A prefix needs to be given which will be used as prefix for all the Objects generated while creating proxy. If we click on continue and then select on finish the proxy will be generated. Then we need to **save and Activate the Proxy**.
- Once the proxy is created for a service interface then that Interface will be marked as Green.

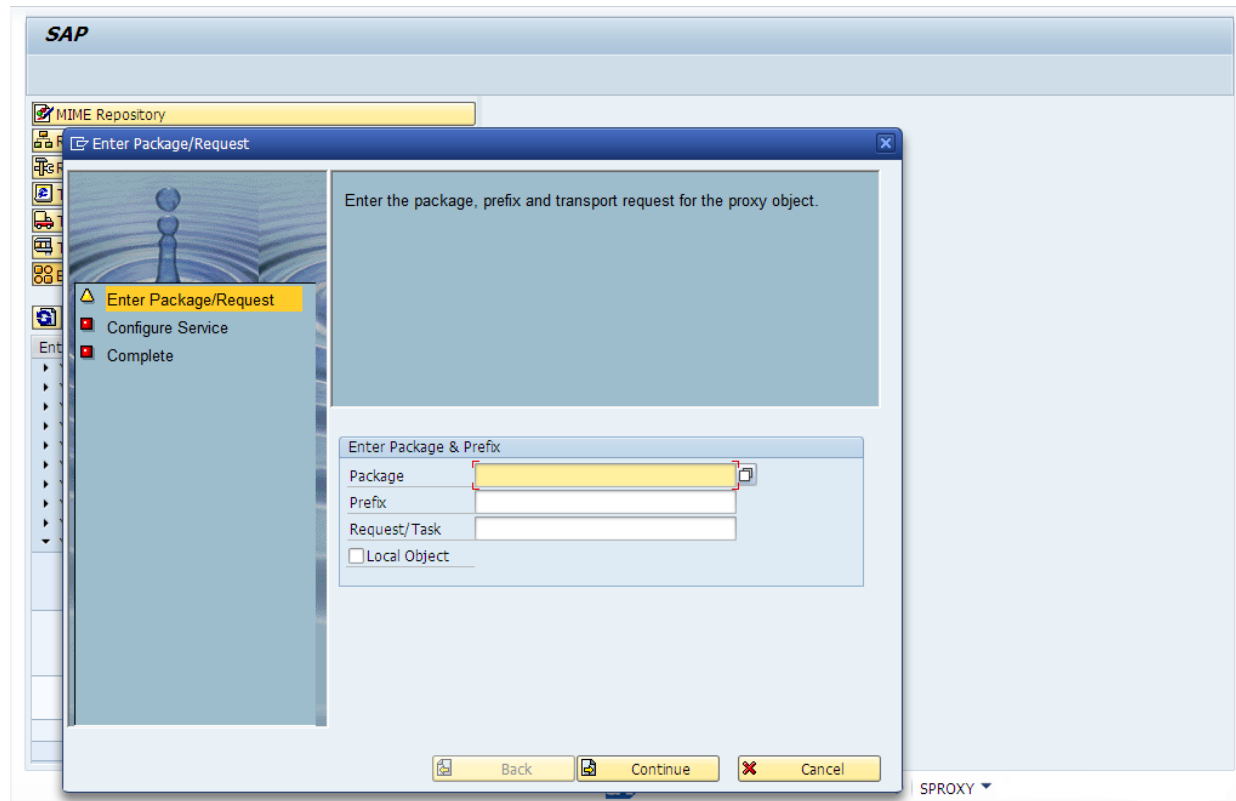


Fig 3

As you can see in “Fig 1” the outbound Proxy “S168\_InvoiceDeltaReplication\_OB” was already created so it is marked Green. In the Right hand side of the “Fig 1” we can find the Proxy Name when we double click on the generated Service Interface. In this case since the Outbound Proxy was already generated we can find the Proxy name of that.

**Proxy Name:** ZDL1CO\_S168\_INVOICE\_DELTA\_REPL

d. The objects created when we generate a Proxy is as follows

#### i. Data Type

"S168\_InvoiceDeltaReplication\_DT"

## ii. Message Type

"S168\_InvoiceDeltaReplication\_MT"

## iii. Proxy Class.

Proxy Class name and Proxy Name will be same.

(For example "ZDL1CO\_S168\_INVOICE\_DELTA\_REPL")

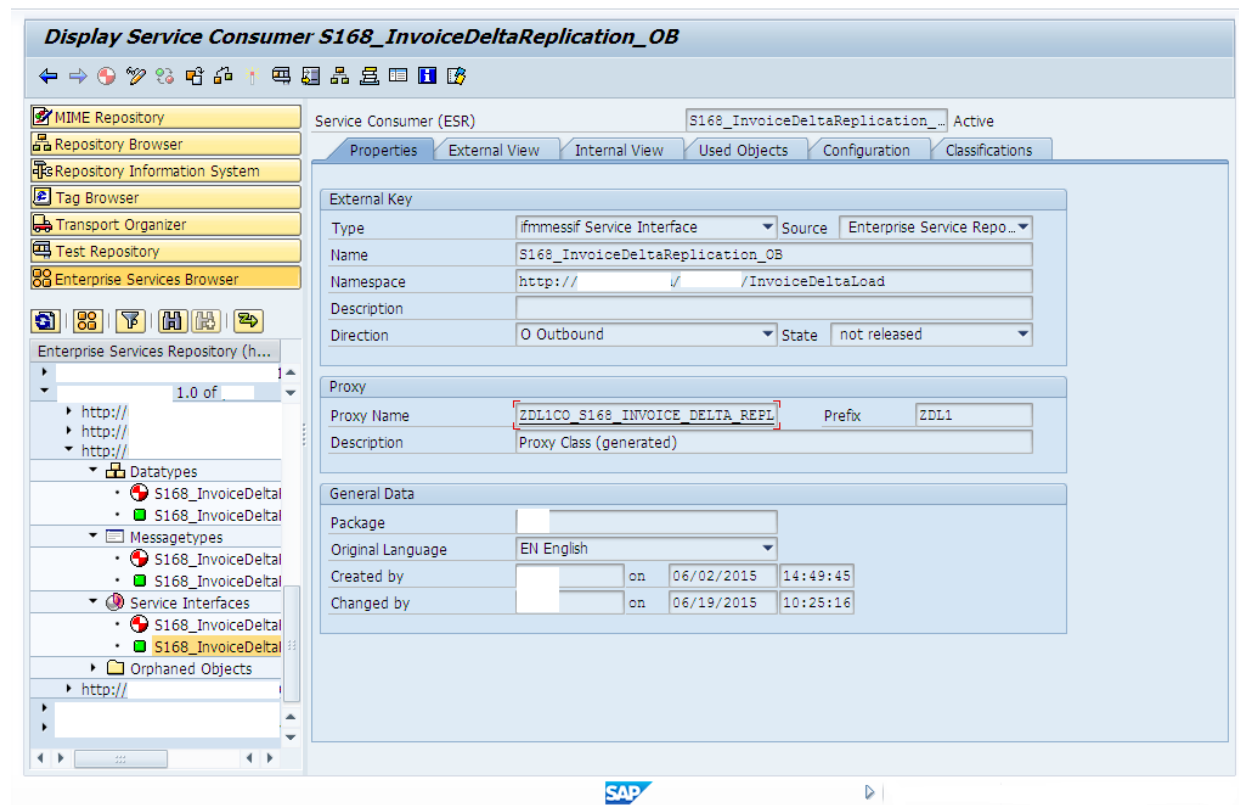


Fig 4

## Coding required in ABAP end

**Step1:** Code to be written in SE38 to pass the data from internal table to the Proxy.

a. Following Screen Shot shows the methods available in Proxy (ZDL1CO\_S168\_INVOICE\_DELTA\_REPL).

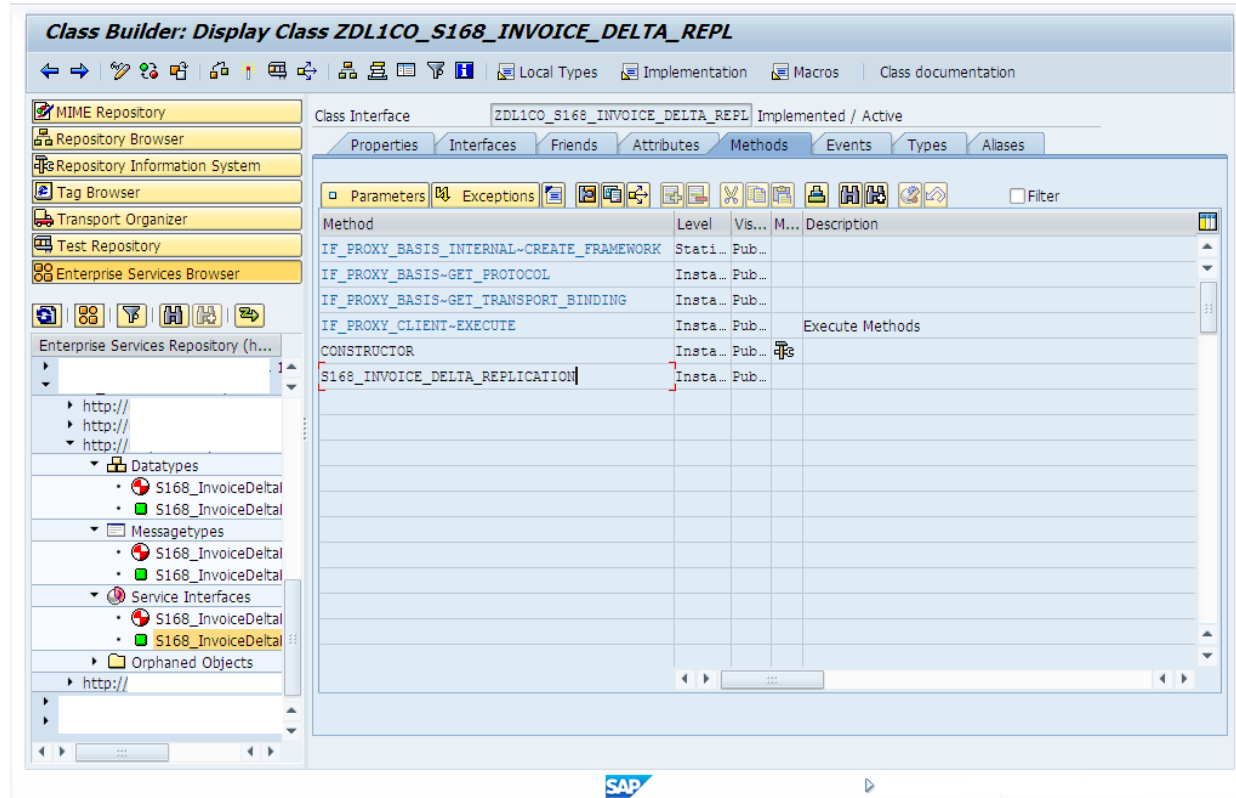


Fig 5

- b. Call the Method “S168\_INVOICE\_DELTA\_REPLICATION” and Pass the data to the Importing parameter of the Method. Below Screen Shot shows the importing parameter of the Method.

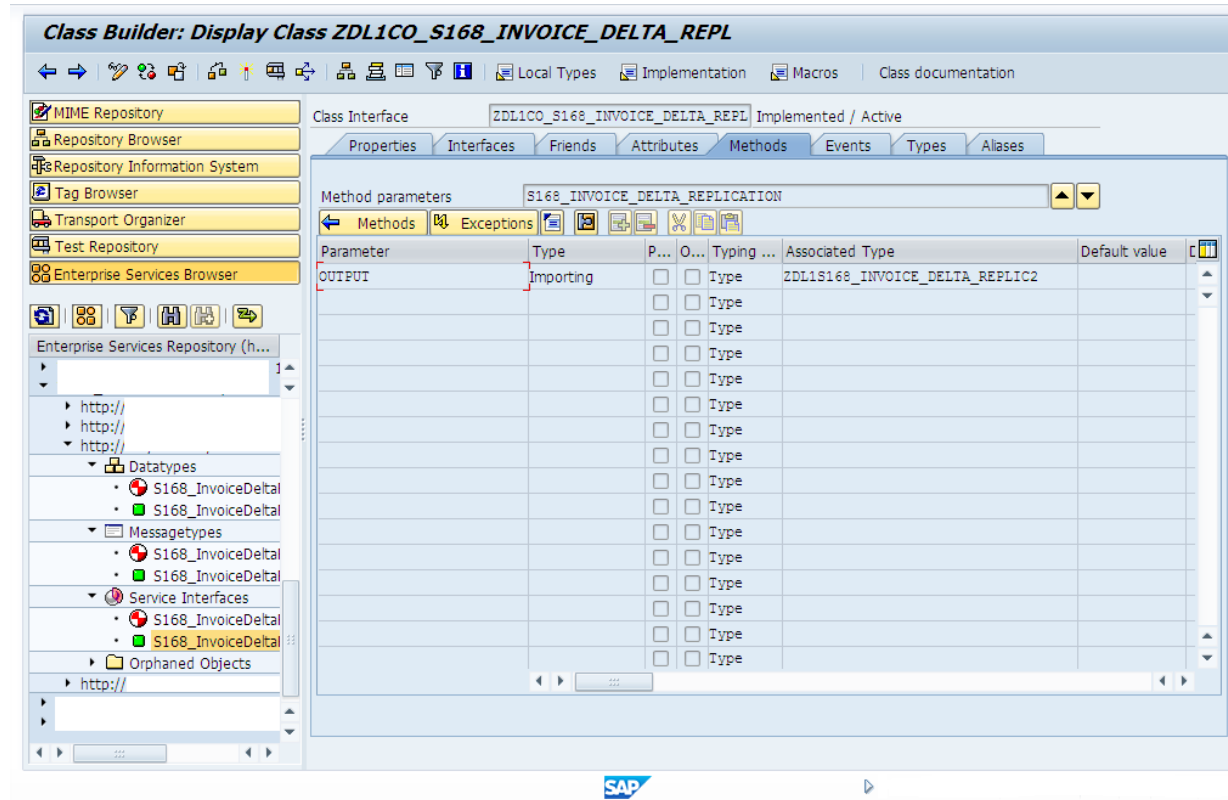


Fig 6

- c. The Importing Parameter “OUTPUT” will be a Deep Structure.

Following Screen Shots show the deep structure of “OUTPUT”.

OUTPUT(Parameter) – ZDL1S168\_INVOICE\_DELTA\_REPLIC2 (Associated Type)



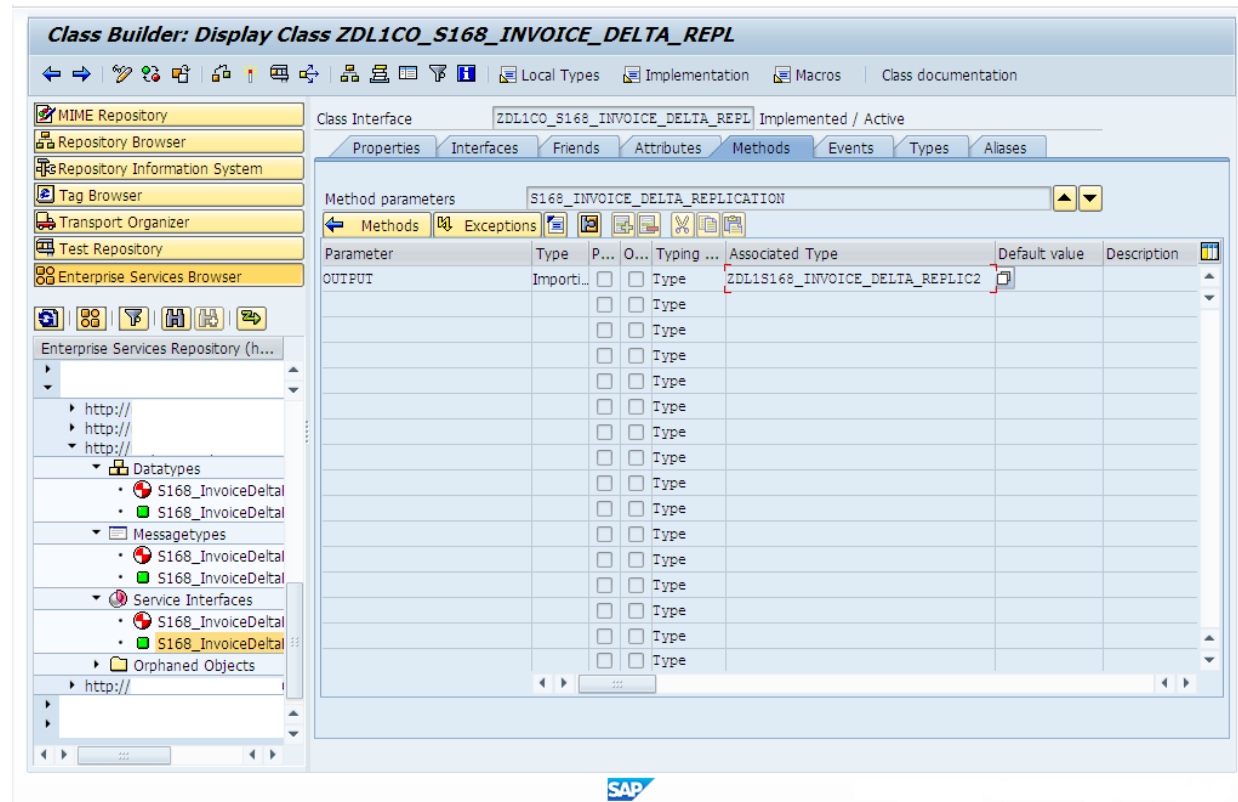


Fig 7

### Components of ZDL1S168\_INVOICE\_DELTA\_REPLIC2

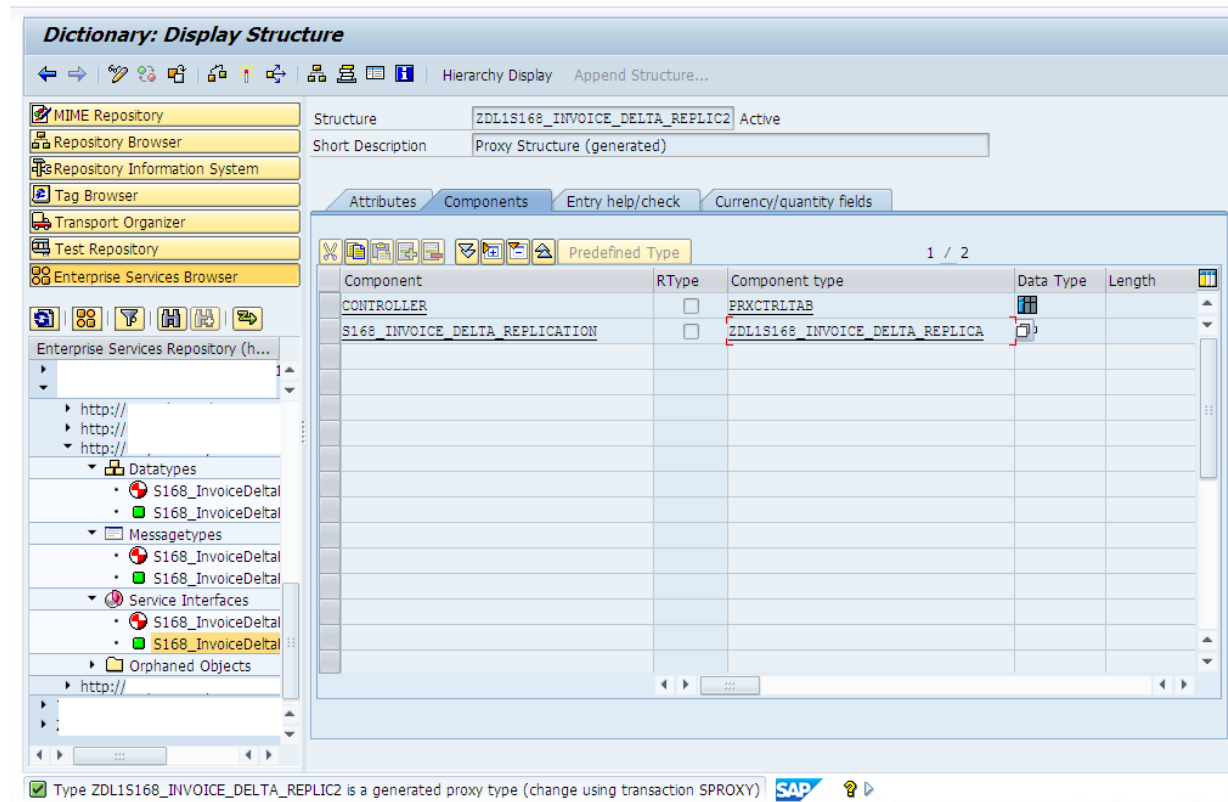


Fig 8

S168\_INVOICE\_DELTA\_REPLICATION(Component) – ZDL1S168\_INVOICE\_DELTA\_REPLICA(Component type)

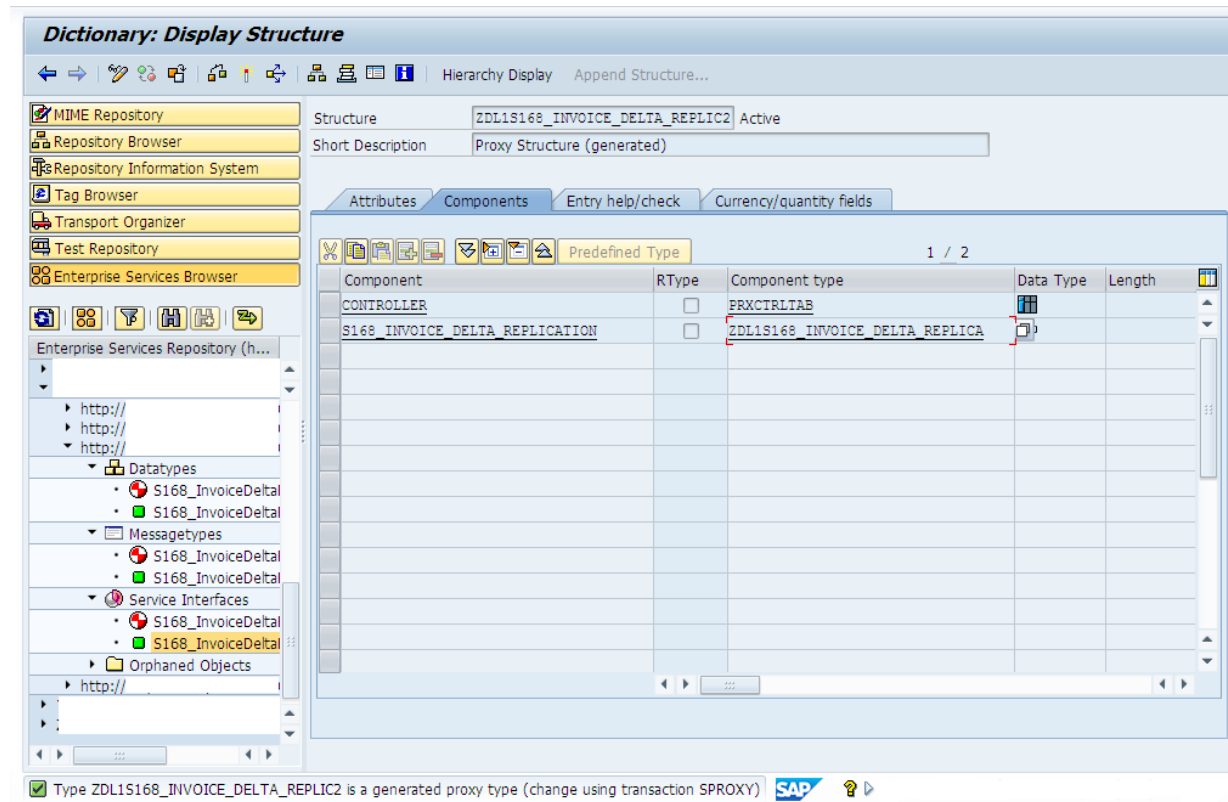


Fig 9

Components of ZDL1S168\_INVOICE\_DELTA\_REPLICA

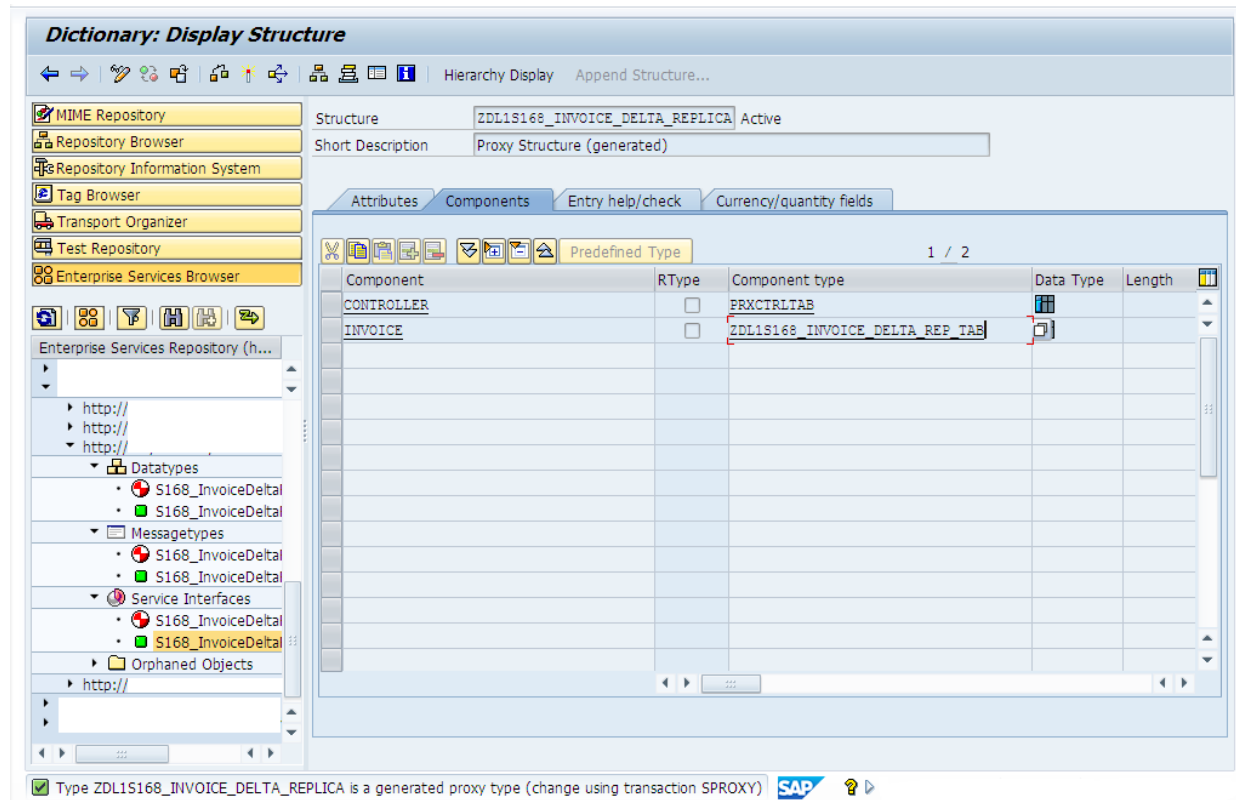


Fig 10

“ZDL1S168\_INVOICE\_DELTA\_REP\_TAB” will be the table type of line type

“ZDL1S168\_INVOICE\_DELTA\_REPLIC1” the Structure of the Proxy table to which the data has to be passed. This structure of Line type must be similar to the structure of internal table of the program from which the data has to be passed.

The Structure of Line Type “ZDL1S168\_INVOICE\_DELTA\_REPLIC1” in this case is

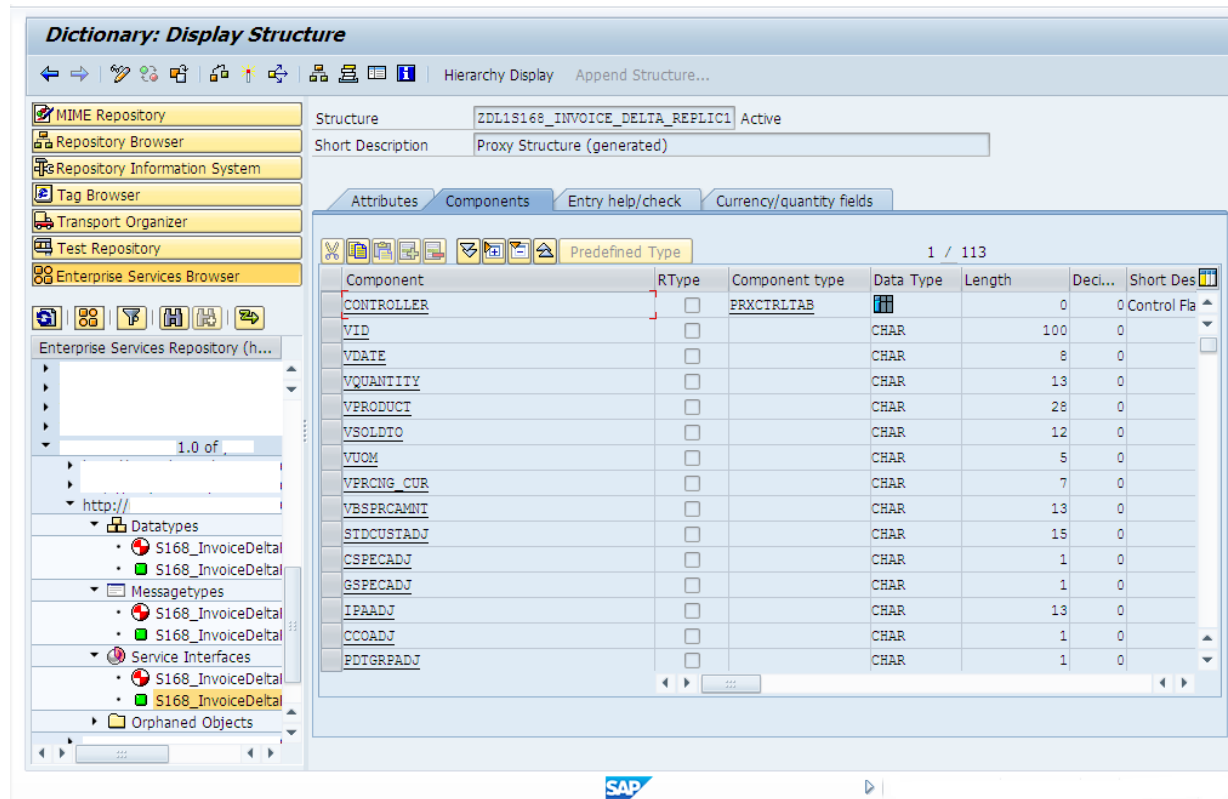


Fig 11

d. Now before passing the data to the Method we need to pass the entire data available in the Internal table

to the Internal table which is of type, table type "ZDL1S168\_INVOICE\_DELTA\_REP\_TAB" explained above.

e. Pass the data from step d to "OUTPUT-S168\_INVOICE\_DELTA\_REPLICATION- INVOICE"

(Deep Structure) and then pass the "OUTPUT" to Method "S168\_INVOICE\_DELTA\_REPLICATION"

of class "ZDL1CO\_S168\_INVOICE\_DELTA\_REPL".

## Sample ABAP code

Following is the Example code needs to be written in ABAP end.

Here it\_fin\_delta[] is internal table in which the data to be passed to Proxy will be available.

```
DATA: it_output          TYPE zdl1s168_invoice_delta_replic2,

      it_delta           TYPE zdl1s168_invoice_delta_rep_tab,

      wa_delta           TYPE zdl1s168_invoice_delta_replic1,

      z01_otc_delta_class TYPE REF TO zdl1co_s168_invoice_delta_repl,

      z_otc_cl_exception  TYPE REF TO cx_ai_system_fault,

      l_error             TYPE string.

IF it_fin_delta[] IS NOT INITIAL.

LOOP AT it_fin_delta INTO wa_final.

MOVE-CORRESPONDING wa_final TO wa_delta.
```

```
APPEND wa_delta TO it_delta.

CLEAR:wa_final,wa_delta.

ENDLOOP.

it_output-s168_invoice_delta_replication-invoice = it_delta.

CREATE OBJECT z01_otc_delta_class.

CREATE OBJECT z_otc_cl_exception.

CLEAR:l_error.

TRY.

    CALL METHOD z01_otc_delta_class->s168_invoice_delta_replication

        EXPORTING

            output = it_output.

    COMMIT WORK.

CATCH cx_ai_system_fault INTO z_otc_cl_exception.

    l_error = z_otc_cl_exception->errortext.

    MESSAGE l_error TYPE c_e.

EXIT.
```

```
ENDTRY.  
  
IF sy-subrc EQ 0.  
  
    MESSAGE text-i05 TYPE c_s.  
  
ENDIF.  
  
  
ELSE.  
  
WRITE: text-i06.  
  
ENDIF.
```

## Issue which can be faced during development with solution

### Issue:

There was an issue experienced during the development for the Outbound Proxy. The issue was that the Data sent from ABAP side was getting stuck in the Queue. If we open the T-Code SMQ2 we can see that the queue is getting stuck and if we open the queue and process the data



manually then the data was being sent to the PI side.

**Solution:**

We have to register the name of queue which will be created and in which the data will be stuck when we try to pass data.

For example consider the Queue name will be “XBTS1254”. Here you can check that whenever the data is passed it gets stuck in queue which starts with “XBTS”. So we need to register queue with name XBTS\*

1. Go to Transaction SMQR. This T-Code is used to Register the Queue.

**qRFC Monitor (QIN Scheduler)**

Registration Register without activation Deregistration qRFC Monitor

Scheduler Information	
Scheduler Status	: Inactiv.
Last Update (Every 2 Minutes)	: 25.06.2015 09:29:53
Name of AS Group (DEFAULT = All):	DEFAULT
Number of Entries Displayed	2
Host ID	:
Number of Active Connections	0

Cl.	Queue name	Type	Mode	Max. Runtime	Attempts	Pause	Destination with LOGON Data
<input checked="" type="checkbox"/> 110	R3A*	R	D	60	30	300	
<input type="checkbox"/> 110	XBIS*	R	D	60	30	300	

SAP

Fig 12

2. Click on Registration. The following POP up appears.

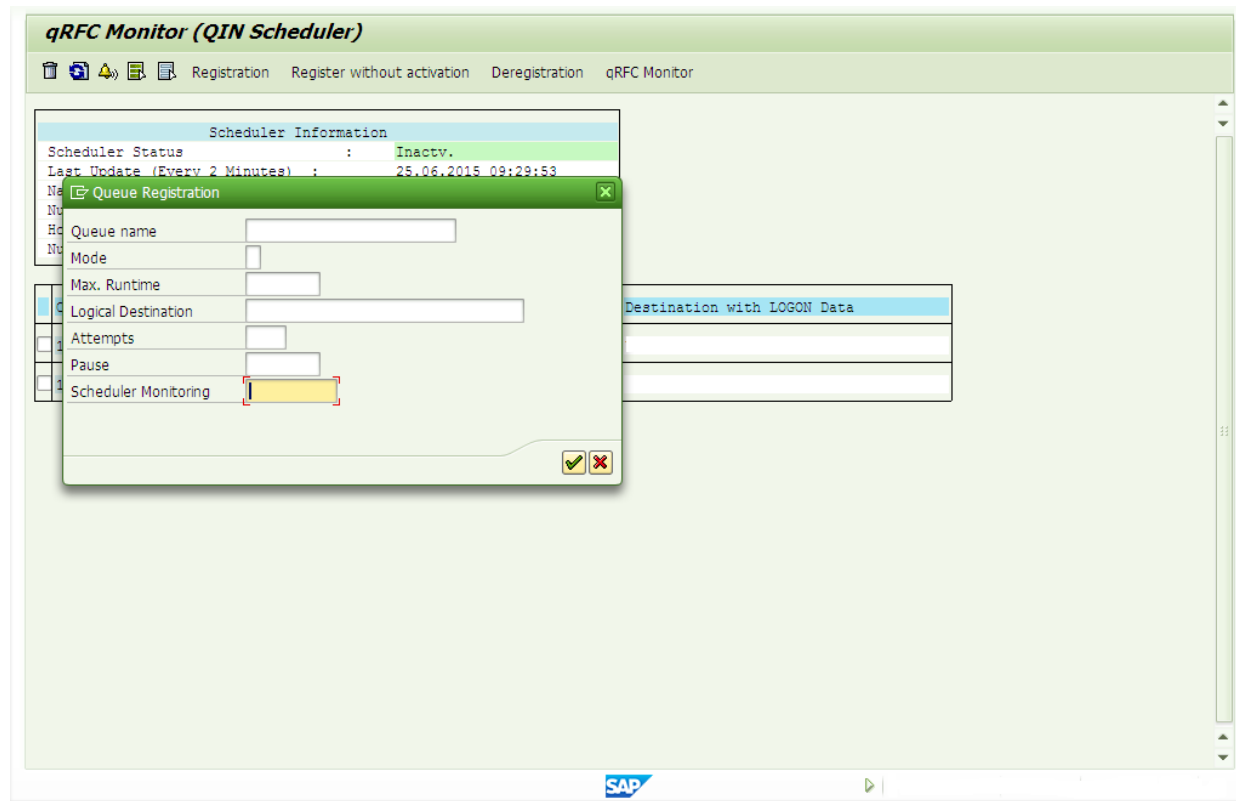


Fig 13

3. Queue name: XBTS\*(Example)

MODE: D (Preferable by SAP)

Max.Runtime: 60(Example)

**Logical Destination should be Empty. Else the data will still be stuck in Queue.**

Attempts: 30(Example)

Pause: 300(Example)

Scheduler Monitoring: 0(Example)

4. Once the Queue is registered with the name then the data will be passed directly to Proxy without getting stuck.

[Alert Moderator](#)

### Assigned tags

---

ABAP Development

abap

proxy

### Similar Blog Posts

---



[Consuming SAP HANA Procedures from ABAP \[In Side Car Approach\]](#)

By Former Member Sep 25, 2014

[Create dynamic proxy persistently in Java and ABAP](#)

By Jerry Wang Jan 28, 2017

## Error Handling of Outbound Proxy Calls

By Tobias Trapp Feb 18, 2013

### Related Questions



#### [Proxy out of date error. Please help!](#)

By Gopalkrishna Baliga Oct 12, 2011

#### [Abap Proxy Generation \(Outbound\)](#)

By Former Member Sep 01, 2006

#### [Error in ABAP Proxy SERIALIZE\\_APPLICATION\\_DATA Conversion error ABAP => XML](#)

By Satish Karemore Oct 23, 2017

## 5 Comments

You must be [Logged on](#) to comment or reply to a post.



Former Member

December 8, 2015 at 5:12 pm

Great work Sathwik! Really appreciate your understanding in Outbound Proxy.

Like 0 | Share



**Sathwik Reddy G K** | Blog Post Author

December 8, 2015 at 5:45 pm

Thanks Gilbert 😊

Like 0 | Share



**Former Member**

December 11, 2015 at 9:59 am

I am new to proxy concept in abap. Your work helped me quite well in understanding the concept

Like 0 | Share



**AKIL SYED**

May 22, 2019 at 8:41 am

I am from LTI. Nice document....

Like 0 | Share



**rajeswari CH**

August 12, 2020 at 1:20 am

Simple and clear understanding blog.Thank you.

Like 0 | Share

**Find us on**

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support