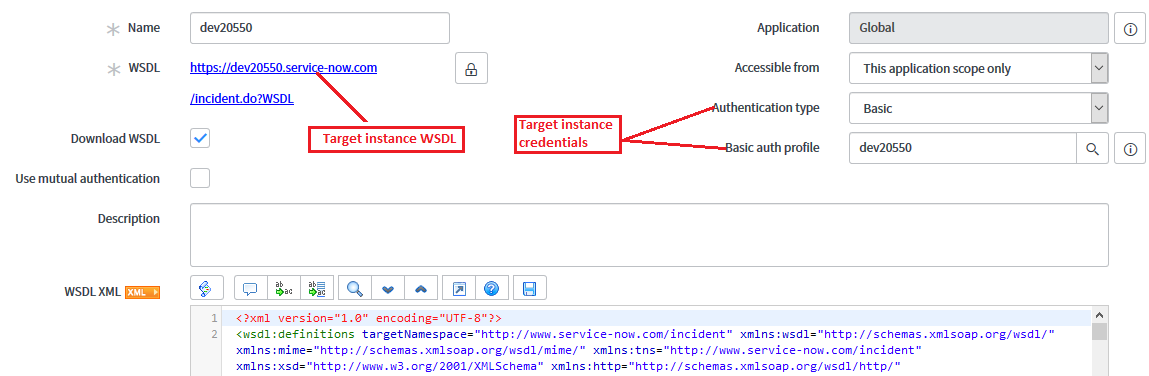
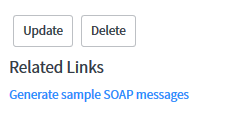
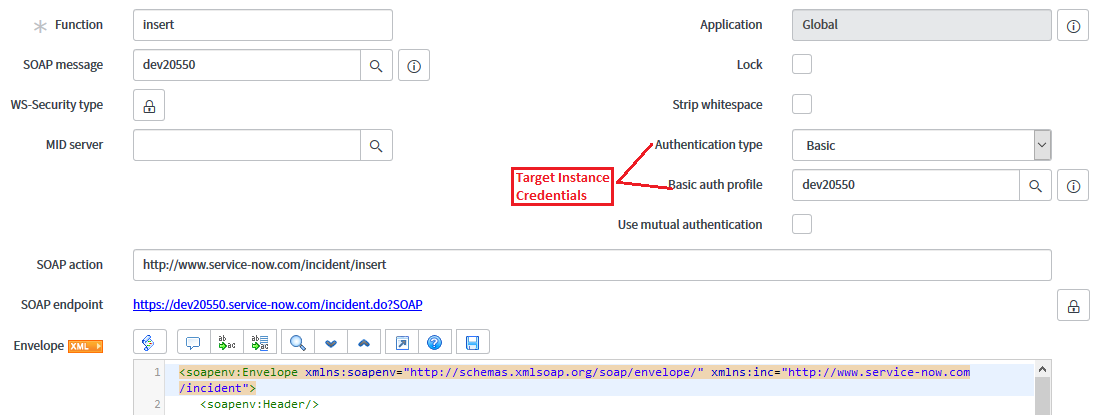
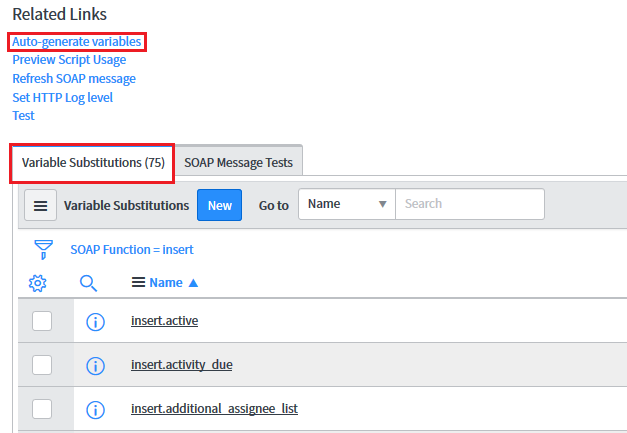
# ServiceNow to ServiceNow Bidirectional Integration using SOAP

1. Create outbound SOAP message in Source instance.
2. Click on ***Generate Sample SOAP Message*** related link to retrieve WSDL XML and SOAP Message functions.

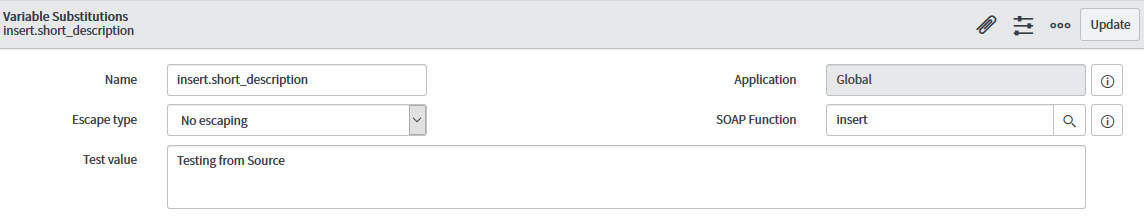


## Insert function:

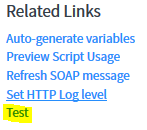
1. Select Insert SOAP Message function to create an Incident in the Target instance.
2. Click on the ***Auto-generate variables*** related link to generate ***Variable Substitutions.***

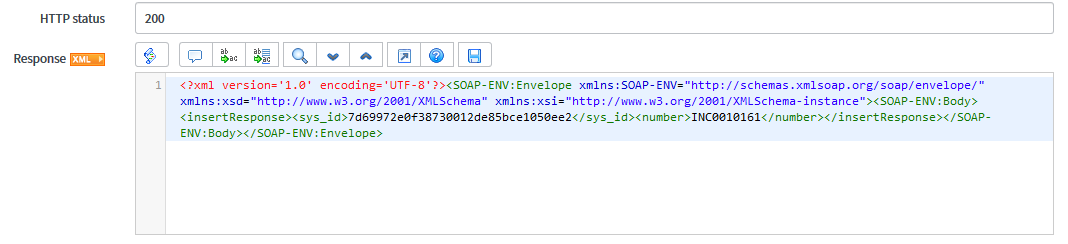


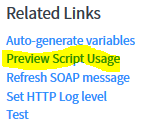
1. Pass the variable for testing.

Ex**: instert.short\_description = ‘Test’;**

1. Click on ***Test*** related Link.



1. Will have the response and the status as shown below.
2. Click on the ***Preview Script Usage*** related link to copy the insert function code.



1. Copy the complete insert function code which pops up.

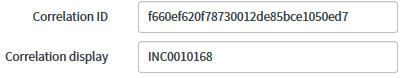


1. Create a after business rule on Incident table.

**When to Run** : After Insert

**Condition**: Correlation ID is EMPTY

1. (function executeRule(current, previous /\*null when async\*/) {
3. try {
4. var s = new sn\_ws.SOAPMessageV2('dev20550', 'insert');
6. //Pass all the required parameters
7. s.setStringParameterNoEscape('insert.assignment\_group', current.assignment\_group);
8. s.setStringParameterNoEscape('insert.caller\_id', current.caller\_id);
9. s.setStringParameterNoEscape('insert.assigned\_to', current.assigned\_to);
10. s.setStringParameterNoEscape('insert.description', current.description);
11. s.setStringParameterNoEscape('insert.opened\_by', current.opened\_by);
12. s.setStringParameterNoEscape('insert.category', current.category);
13. s.setStringParameterNoEscape('insert.short\_description', current.short\_description);
14. s.setStringParameterNoEscape('insert.subcategory', current.subcategory);
16. //Excute the SOAP insert function
17. var response = s.execute();
18. var responseBody = response.getBody();
19. var status = response.getStatusCode();
21. //Read XML response and get sys\_id & number of target instance.
22. var xmlutil = new GlideXMLUtil();
23. var payloadDoc = new XMLDocument(responseBody);
24. var numRef = payloadDoc.getElementValueByTagName("number");
25. var sysID = payloadDoc.getElementValueByTagName("sys\_id");
27. //Update current incident record with Target sys\_id.
28. current.correlation\_id = sysID;
29. current.correlation\_display = numRef;
30. current.update();
32. }
33. catch(ex) {
34. var message = ex.getMessage();
35. }
37. })(current, previous);
38. Create an Incident in source instance and see the Correlation ID (Target Instance sys\_id ) populated.



## Update function:

1. Go to SOAP messages and open UPDATE function.
2. Perform same steps as we have done for Insert function and copy the update script from ***Preview Script Usage***.
3. Create a business rule and paste the code.

**When to Run:** After Update

**Condition:** Correlation ID is not Empty

1. (function executeRule(current, previous /\*null when async\*/) {
3. // Add your code here
4. try {
5. var s = new sn\_ws.SOAPMessageV2('dev20550', 'update');
7. //Pass parameters which needs to be updated in Target instance table
8. //Must pass the Target instance SYS ID which is store in CORRELATION ID
9. s.setStringParameterNoEscape('update.sys\_id', current.correlation\_id);
10. s.setStringParameterNoEscape('update.short\_description', current.short\_description);
12. //Passing worknotes whenever notes added to the Source instance table.
13. if (current.work\_notes.changes()){
14. var lastWN = current.work\_notes.getJournalEntry(1);
15. var index2 = lastWN.indexOf("\n");
16. s.setStringParameterNoEscape('update.work\_notes', lastWN.substring(index2));
17. }
18. // Excute update SOAP function
19. var response = s.execute();
20. var responseBody = response.getBody();
21. var status = response.getStatusCode();
22. }
23. catch(ex) {
24. var message = ex.getMessage();
25. }
26. })(current, previous);

## Copying Attachments:

1. In order to send attachments, we need to create a new web service for ***ECC Queue*** table.

WSDL: [ECC QUEUE WSDL](https://dev24491.service-now.com/ecc_queue.do?WSDL)

Reference Links: [ServiceNow Guru](https://www.servicenowguru.com/integration/sending-attachment-servicenow/)

[ServiceNow Docs](https://docs.servicenow.com/bundle/london-application-development/page/integrate/inbound-soap/reference/r_AttachmentCreatorSOAPWebService.html)

1. Perform same steps 1- 9 to generate SOAP messages, functions, Variable Substitutions and copy the script from Preview Usage Script.

## Insert function:

1. Create a business rule on sys\_attachment table.

**When to Run**: After Insert

**Condition**: Table name is incident

1. (function executeRule(current, previous /\*null when async\*/) {
3. // Add your code here
4. var inc = new GlideRecord(current.table\_name);
5. inc.get(current.table\_sys\_id);
6. var cID = inc.correlation\_id;
8. //Attachment Creator //https://www.servicenowguru.com/integration/sending-attachment-servicenow/
9. var sa = new GlideSysAttachment();
10. var binData = sa.getBytes(current);
11. var encData = GlideStringUtil.base64Encode(binData);
13. var sm = new sn\_ws.SOAPMessageV2('Dev24491 - Attach', 'insert');
15. //Pass mandatory parameters Source, Payload, Agent, Topic, Name
16. sm.setStringParameterNoEscape('insert.source', 'incident:'+cID);
17. sm.setStringParameterNoEscape('insert.payload', encData);
18. sm.setStringParameterNoEscape('insert.agent', 'AttachmentCreator');
19. sm.setStringParameterNoEscape('insert.topic', 'AttachmentCreator');
20. sm.setStringParameterNoEscape('insert.name', current.file\_name+':'+current.content\_type);
22. //Excute Insert SOAP function
23. var response1 = sm.execute();
24. var responseBody1 = response1.getBody();
25. var status1 = response1.getStatusCode();
27. })(current, previous);
28. The above script will send attachments whenever attachment is added to the Incident.
29. Perform same steps in target instance for ***Bidirectional*** integration.