

Leading the **Integration** Revolution

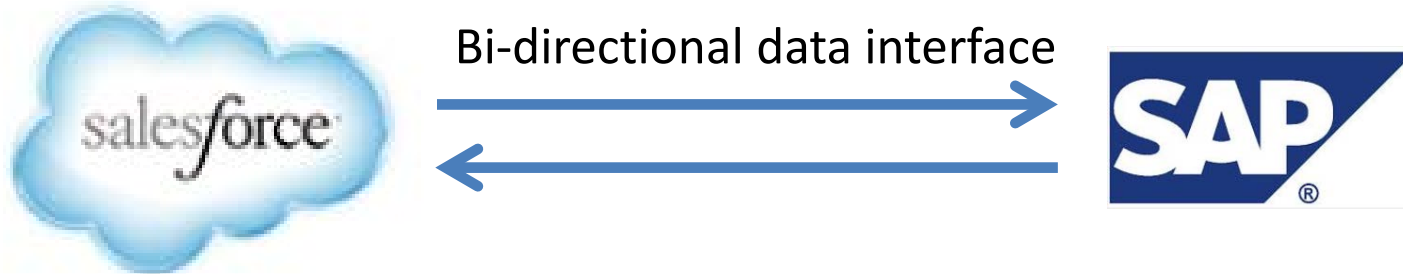
Your business problems have changed.
Why hasn't your integration solution?



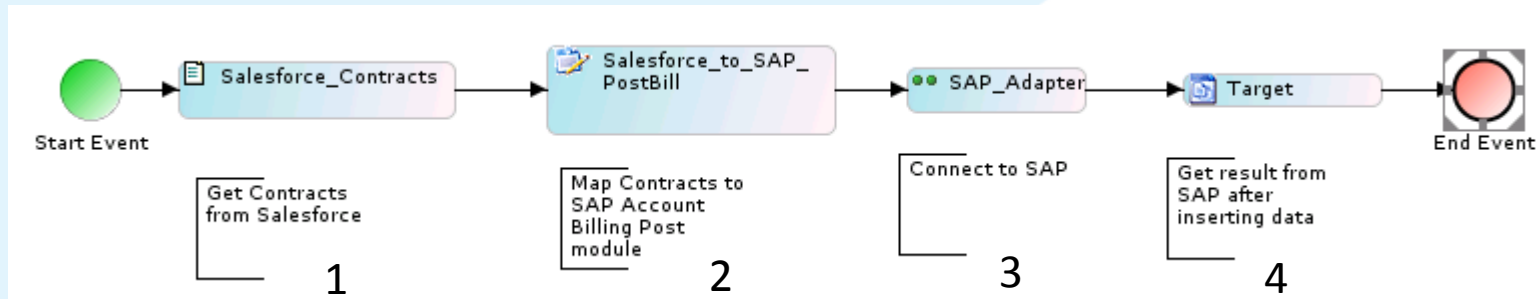
ADEPTIA

Use Case: Integrating Salesforce with SAP

High-level scenario



Technical design



For this orchestration we are going to get Contracts from Salesforce and post them into SAP's Accounting module. Specifically Billing Post record in SAP Accounting. Similarly using this process as a template you can design a reverse process to get data from SAP and push the updates into Salesforce.

The orchestration gets Contracts from Salesforce which are obtained by executing a separate orchestration that logs into Salesforce and gets the contracts. These are stored into a file which is then used as a source for the process flow shown above. Also refer to the attached deployable solution that can be loaded into your Adeptia instance using the Migration Utility.

Description of the orchestration:

1. "Salesforce_Contracts" is a file that is created after running "Salesforce Get Contracts" process flow
2. "Salesfroce_to_SAP_PostBill" is a mapping that maps the Contracts to the SAP's Billing Post BAPI schema
3. "SAP_Adapter" is a pre-built connector that is used to connect to SAP via BAPI
4. "Target" is a context variable that stores the results of the SAP BAPI method execution

Note: In order to execute the process you would also need to install the SAP Integration Accelerator in your Adeptia instance . This accelerator has BAPI and IDoc connectors. You can download this accelerator by going to www.adeptia.com
> Solutions > SAP Integration

Let's run the orchestration

Run-time: Execute the Salesforce to SAP orchestration

Execute the process flow by going to Develop > Process Flow

Click on the Execute icon next to the process name highlighted below. This action will execute the process and then click on the “Go to Logs” to see the results. Results would show the response messages with status for each of the records that are pushed into SAP.

The screenshot shows the SAP Process Flow console. The top navigation bar includes Home, Develop, Monitor, and Administer. Below this, there are tabs for Projects, Solutions, Process, Services, Events, Web Forms, and Reports and Dashboards. The main area is titled 'Process > Process Flow' and contains a table of process flows. The 'Salesforce_to_SAP_Contracts' process is highlighted in blue. A red checkmark is next to its name. A blue arrow points from the 'Go to Logs' button in the success message to the 'Go to Logs' button in the process flow log details.

Process Flow List:

Name	Description	Owner	Project Name	Modified	Action
Salesforce_to_SAP_Contracts	Post Bill record into SAP	admin	UseCase_Salesforce_to_...	11/19/13 22:55	[Menu]
Salesforce_Get_Contracts	get contracts from SF	admin	UseCase_Salesforce_to_...	11/19/13 20:31	[Menu]

Success Message:

A job has been submitted to Queue Processor for executing a process flow Salesforce_to_SAP_Contracts with process flow id 010064050089137925932453. See the Monitor > History > Process Flow Log for execution details.

Process Flow Log Details:

Process Flow > Salesforce_to_SAP_Contracts (010064050089137925932453200020)

PID: 010064050089138490609524600007 Status: **Executed** Triggered By: admin (User Name: admin)

Start Time: 11/20/2013 00:08:15 Execution Time: 3 second (s) 473 ms Triggered Event Type: Manual

End Time: 11/20/2013 00:08:18 Node Name: localhost

Activity Log:

Activity Name	Activity Type	Start Time	End Time	Execution Time	Status	Processed Data	Error Reco
ContextTarget (NA)	ContextTarget	11/20/2013 00:08:16	11/20/2013 00:08:16	13 ms	Executed	0	0
SAP_Adapter_CustomPlugin (192168001203131668495098800009)	CustomPlugin	11/20/2013 00:08:15	11/20/2013 00:08:16	818 ms	Executed	0	0
Salesforce_to_SAP_PostBill (010064050089137925886516600017)	DataMapping	11/20/2013 00:08:15	11/20/2013 00:08:15	121 ms	Executed	1077	0
Salesforce_Contracts (010064050089137925907702400019)	FileSource	11/20/2013 00:08:15	11/20/2013 00:08:15	60 ms	Executed	1077	0

Let's build the process flow

Design methodology



Design a high level process flow



Configure all the activities in the process flow

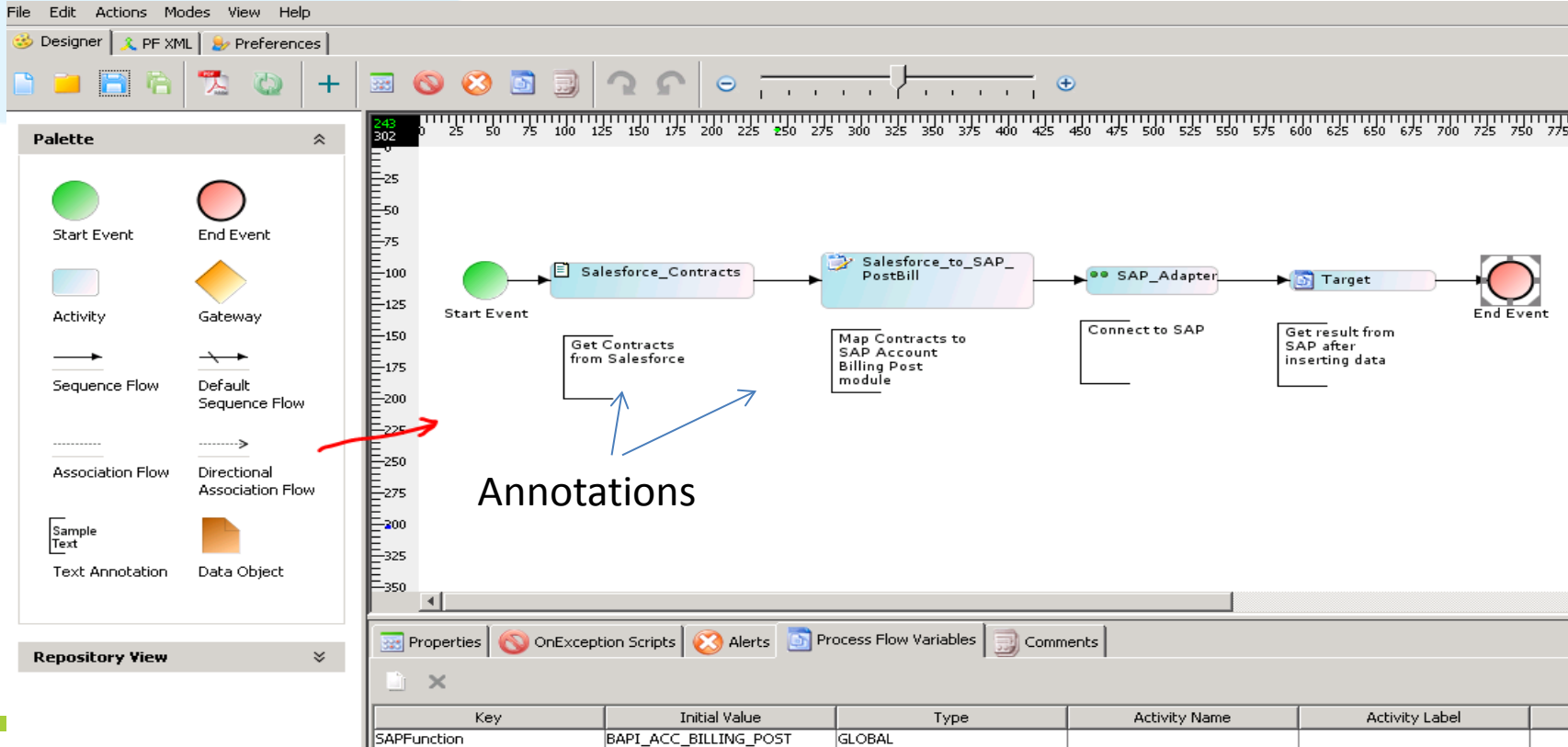


Attach the process to an Event or Execute on-demand

Design: First let's create a Top level design of the solution using the Process Designer

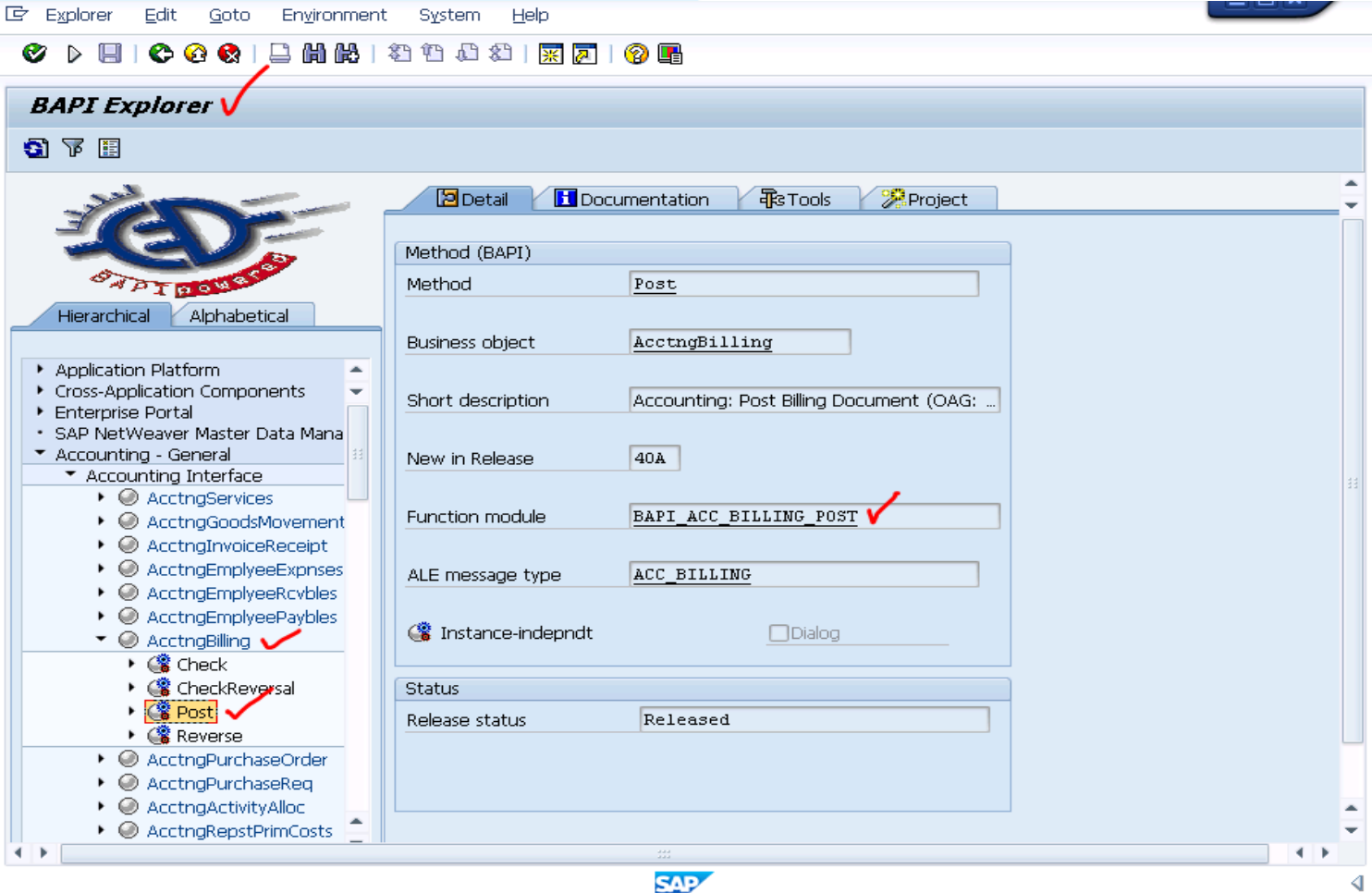
Go to Develop > Process > Process Flow

Create new and in the designer window design a top-level flow by pulling the icons from the palette area. Refer to the reference video and the annotations for each activity to see how to design a process flow. Each activity performs a discreet function such as getting source data, mapping, connecting to SAP and storing response. The process of connecting to Salesforce and getting Contracts is executed separately prior to running this orchestration. For this process design the source is the Contracts data from Salesforce.



Configure: (Step 1) Find the SAP BAPI Function module name using the BAPI Explorer

Using the SAP's BAPI Explorer interface get the name of the function module that will be used to integrate source data into SAP. For example we have chosen Accounting > Account Billing > Post



Configure: (Step 2) Define the schema definition of that BAPI Function in Adeptia

Using Adeptia's BAPI Schema builder screen, create the schema definition of the function by Go to Monitor > My Solutions > SAP > Create Schema

Provide the name and click submit. This would create a XML Input and Output schema for the particular BAPI function. These schemas are used in mapping data to and from SAP.

Home Develop **Monitor** Administer

My Tasks ▾ | My Documents ▾ | Dashboard ▾ | Reports ▾ | History ▾ | My Solutions ▾

My Solutions > SAP > Create Schema

SAP ▶ Create Schema

New SAP Schema

SAP Interface* BAPI

To Email-Id(s) (comma separated)* user@my_company.com

BAPI Function Name* BAPI_ACC_BILLING_POST

Submit

* Mandatory fields.

Note: Please ensure that the function name is valid. (Example: BAPI_PO_GETITEMSREL)

Configure: (Step 3) See the BAPI function schema in XML Schema Manage page

Once the “Create BAPI schema form” is submitted, Adeptia will automatically create and place the Input/Output schemas under Schema > XML manage page. Here user can view all the BAPI schemas and can manage these schemas from this centralized repository.

Services > Schema > XML

Quick Search

<input type="checkbox"/>	Name	Description	Owner	Project Name	Modified
<input type="checkbox"/>	BAPI_ACC_BILLING_CHECK_Output_Schema	BAPI_ACC_BILLING_CHECK output schema	admin	Default	09/16/13 18:34
<input type="checkbox"/>	BAPI_ACC_BILLING_CHECK_Input_Schema	BAPI_ACC_BILLING_CHECK input schema	admin	Default	09/16/13 18:34
<input type="checkbox"/>	BAPI_COMPANY_GETDETAIL_Input_Schema	BAPI_COMPANY_GETDETAIL input schema	admin	UseCase_Salesforce_to_SAP	09/15/13 17:34
<input type="checkbox"/>	BAPI_COMPANY_GETDETAIL_Output_Schema	BAPI_COMPANY_GETDETAIL output schema	admin	UseCase_Salesforce_to_SAP	09/15/13 17:34
<input type="checkbox"/>	BAPI_ACC_BILLING_POST_Input_Schema ✓	BAPI_ACC_BILLING_POST input schema	admin	UseCase_Salesforce_to_SAP	09/15/13 17:33
<input type="checkbox"/>	BAPI_ACC_BILLING_POST_Output_Schema ✓	BAPI_ACC_BILLING_POST output schema	admin	UseCase_Salesforce_to_SAP	09/15/13 17:33
<input type="checkbox"/>	BAPI_PO_GETITEMSREL_Output_Schema	BAPI_PO_GETITEMSREL output schema	SAPDemoUser	Default	09/05/13 15:33
<input type="checkbox"/>	BAPI_PO_GETITEMSREL_Input_Schema	BAPI_PO_GETITEMSREL input schema	SAPDemoUser	Default	09/05/13 15:33
<input type="checkbox"/>	BAPI_ACC_INVOICE_RECEIPT_POST_Output_Schema	BAPI_ACC_INVOICE_RECEIPT_POST output schema	admin	Default	09/05/13 15:24
<input type="checkbox"/>	BAPI_ACC_INVOICE_RECEIPT_POST_Input_Schema	BAPI_ACC_INVOICE_RECEIPT_POST input schema	admin	Default	09/05/13 15:24
<input type="checkbox"/>	BAPI_ACC_EMPLOYEE_EXP_CHECK_Output_Schema	BAPI_ACC_EMPLOYEE_EXP_CHECK output schema	admin	Default	08/29/13 21:48
<input type="checkbox"/>	BAPI_ACC_EMPLOYEE_EXP_CHECK_Input_Schema	BAPI_ACC_EMPLOYEE_EXP_CHECK input schema	admin	Default	08/29/13 21:48

Configure: (Step 4) Map Salesforce Contracts to SAP's Billing Post Input schema

Go to Develop > Services > Data Transformation > Mapping
Map the Salesforce Contracts source schema to the SAP BAPI function target schema. Refer to the mapping created for this use case for more details.

The screenshot shows the ADEPTIA Data Mapper interface with the following components:

- Structure_of_source_XSD:** A tree view of the source schema. The `tns:records (Contract) (M, E - Contract)` element is highlighted with a red underline. It contains various sub-elements like `ens:Account`, `ens:BillingCity`, and `ens:CompanySigned`.
- Structure_of_destination_XSD:** A tree view of the target schema. The `BAPI_ACC_BILLING_POST_INPUT (BAPI_ACC_BILLING_POST_Input_Schema)` element is highlighted with a red underline. It contains an `INPUT` container with a `CUSTOMERCPD (M)` element, which lists various fields like `NAME`, `POSTL_CODE`, `CITY`, `COUNTRY`, `STREET`, `PO_BOX`, `POBK_CURAC`, `BANK_ACCT`, `BANK_NO`, `BANK_CTRY`, `TAX_NO`, and `REGION`.
- Mapping:** A central area with a vertical line and arrows showing the mapping from source elements to target elements. A red arrow points from `tns:records` to the `CUSTOMERCPD` container. Blue arrows point from various sub-elements of `tns:records` to their corresponding fields in the `CUSTOMERCPD` container.
- Mapping Rules:** A tab at the bottom showing a rule: `1 <S> tns:reco..` pointing to `1 <T> CUSTOMER..`
- Parameters:** A panel on the right showing a tree view of parameters: Global Variables, Local Variables, Custom Methods, Keys, Context Variables, and Templates.

Deploy: Execute on-demand or attach the process (orchestration) to a trigger

Attach the Process to an event by right-clicking on the Process name and choosing 'Bind Event' option. Here we are manually executing the process flow.

The screenshot displays the Adeptia web interface. At the top, there are navigation tabs: Home, Develop, Monitor, and Administer. Below these are menu items: Projects, Solutions, Process, Services, Events, Web Forms, and Reports and Dashboards. The main content area is titled "Process > Process Flow". It contains a table with columns: Name, Description, Owner, Project Name, Modified, and Action. Two processes are listed: "Salesforce_to_SAP_Contracts" and "Salesforce_Get_Contracts". A red checkmark is next to the first process name. A blue arrow points from the gear icon of the first process to a "Success" message box. The "Success" message states: "A job has been submitted to Queue Processor for executing a process flow Salesforce_to_SAP_Contracts with process flow id 010064050089137925932453. See the Monitor > History > Process Flow Log for execution details." Below the message is a "Go to Logs" button with a red checkmark and a blue arrow pointing to the "Process Flow Logs" section. The "Process Flow Logs" section shows a table with columns: Activity Name, Activity Type, Start Time, End Time, Execution Time, Status, Processed Data, and Error Reco. The "Status" column for the first row is circled in red and contains the word "Executed".

Name	Description	Owner	Project Name	Modified	Action
✓ Salesforce_to_SAP_Contracts	Post Bill record into SAP	admin	UseCase_Salesforce_to_...	11/19/13 22:55	⋮
⚙️ Salesforce_Get_Contracts	get contracts from SF	admin	UseCase_Salesforce_to_...	11/19/13 20:31	⋮

Success

A job has been submitted to Queue Processor for executing a process flow Salesforce_to_SAP_Contracts with process flow id 010064050089137925932453. See the Monitor > History > Process Flow Log for execution details.

[Go to Logs](#)

History > Process Flow Logs

Process Flow Logs: Salesforce_to_SAP_Contracts

Process Flow > Salesforce_to_SAP_Contracts (010064050089137925932453200020)

PID:	010064050089138490609524600007	Status:	Executed	Triggered By (User Name):	admin		
Start Time:	11/20/2013 00:08:15	Execution Time:	3 second (s) 473 ms	Triggered Event Type:	Manual		
End Time:	11/20/2013 00:08:18	Node Name:	localhost	Solution Detail			
Activity Name	Activity Type	Start Time	End Time	Execution Time	Status	Processed Data	Error Reco
ContextTarget (NA)	ContextTarget	11/20/2013 00:08:16	11/20/2013 00:08:16	13 ms	Executed	0	0
SAP_Adapter_CustomPlugin (192168001203131668495098800009)	CustomPlugin	11/20/2013 00:08:15	11/20/2013 00:08:16	818 ms	Executed	0	0
Salesforce_to_SAP_PostBill (010064050089137925886516600017)	DataMapping	11/20/2013 00:08:15	11/20/2013 00:08:15	121 ms	Executed	1077	0
Salesforce_Contracts (010064050089137925907702400019)	FileSource	11/20/2013 00:08:15	11/20/2013 00:08:15	60 ms	Executed	1077	0

Managing the activities created for this use case

You can manage and view all the activities of this use case by going to Develop > Projects
Click on the project named “UseCase_Salesforce_to_SAP” and it will show all the activities configured for this use case under its related Service categories. You can open any category to view its activities.

Projects

List of activities included in project "UseCase_Salesforce_to_SAP"

◀ Back Move ↺

<input type="checkbox"/>	Name	Description	Owner
[-] Custom Plugin (1)			
<input type="checkbox"/>	SAP_Adapter_CustomPlugin	Sends the request to execute and get back the ou...	SAPDemoUser
[-] Data Mapping (4)			
<input type="checkbox"/>	Salesforce_Get_Contracts	get contracts from Salesforce	admin
<input type="checkbox"/>	Salesforce_ServerURL	get header information	admin
<input type="checkbox"/>	Salesforce_to_SAP_PostBill	Post Bill from Salesforce to SAP	admin
<input type="checkbox"/>	Salesfore_Login	Login to Salesforce	admin
[-] File Source (1)			
<input type="checkbox"/>	Salesforce_Contracts	get Contracts data	admin
[-] File Target (1)			
<input type="checkbox"/>	Salesforce_Contracts	file target to store contracts file	admin
[-] Process Flow (2)			
<input type="checkbox"/>	Salesforce_Get_Contracts	get contracts from SF	admin
<input type="checkbox"/>	Salesforce_to_SAP_Contracts	Post Bill record into SAP	admin
[-] Web Service Consumer (1)			

Leading the **Integration** Revolution

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

Thank You!



ADEPTIA