

NetSuite Solution Training

October 23, 2011

443 North Clark St, Suite 350
Chicago, IL 60610
Ph: (312) 229-1727

www.adeptia.com



ADEPTIA

Prerequisites



1. Setup the tables in backend database as mentioned in the Installation guide.
2. Update the DB Info object and test the connection to the NetSuite database.

The screenshot shows the NetSuite configuration interface for a Database Info object. The breadcrumb path is "Configure > Services > Connector > Database Info > DatabaseConnToNetSuiteSQLServer". The left sidebar shows a tree view with "Database Info" selected. The main area displays "Standard properties" for the object "DatabaseConnToNetSuiteSQLServer".

Standard properties:

- Name: DatabaseConnToNetSuiteSQLServer
- Description: For connect to SQL Server Database
- Select JDBC Driver: Use Existing (SQLServerJTDSDriver (JTDS Driver for SQL Serv...))
- Create New (Create Database Driver)
- Server URL: jdbc:sqlserver://localhost:1433/n (Help... ✓)
- User: sa
- Password: [masked]
- Confirm Password: [masked]
- Test Database Connection

Advanced properties:

Annotation: Edit the DB Info object specific to your database. If your backend is MySQL or Sql Server than edit that object only.

3. Go to Design > Process Flows and verify that all the NetSuite related flows are there.
4. Verify that your NetSuite login account (provided during Adeptia installation) is correct.

Goals of this training



1. Learn NetSuite solution framework
2. How to modify the search criteria to extract data from NetSuite
 - a. Get last modified this week
 - b. Execute the SearchContact_From_NetSuite
3. How to update/insert data from a database to NetSuite
 - a. Change a record in the database
 - b. Execute the ContactUpdateList_to_NetSuite flow

Understanding the Solution Framework

The image illustrates the solution framework for integrating Adeptia with NetSuite. It consists of three main components:

- Adeptia Integration Suite:** Shows a process flow with several steps. Step 1 is 'PA_SearchFromNetSuiteFromNetSuite', Step 2 is 'PA_SearchCustomerFromNetSuite', and Step 3 is 'T_1Y20ObjectFromNetSuite'. A red arrow points from Step 3 to the database query.
- NetSuite:** Shows a contact record for Allister Sullivan, with fields like Name, Address, and Phone.
- Database:** Shows a SQL query: `SELECT TOP 1000 [NetSuiteInternalID], [CustomForm], [CustomFormInternalID], [EntityID]`. The results table shows a record with `modifyBy` set to `SynchronizedByAdeptia`.

Solution Framework:

Executing NetSuite solution process flows will connect Adeptia to NetSuite and depending upon the search criteria Adeptia will extract the data records from NetSuite and load it into the database. The solution allows end-users to synchronize NetSuite data with backend database.

1. After executing the process check the logs and verify that the process flow has executed successfully. Check the status of each activity by going to the Details section of the process logs. In the Repository there will be several intermediate files that are archived and one of them will contain confirmation (SOAP XML) from NetSuite in terms of the data record that is picked up from the application.
2. Now go to the database of the object table (such as Lists_Contact, Lists_Customers etc) and you will see the new data that is picked up from NetSuite.
3. When the data is picked up from NetSuite the modifyBy field is set to "SynchronizedByAdeptia". This means that the data is successfully picked up by Adeptia and is loaded into the Database.
4. The NetSuite solution flows are of two types:
 - a. Search From NetSuite: get data from NetSuite based on a search criteria and load the data into the database tables.
 - b. Insert/Update To NetSuite: Take data from database and load it into NetSuite.

Also refer to NetSuite SuiteTalk (<http://www.NetSuite.com/portal/developers/suitetalk.shtml>) guidelines to understand how the Web Service communication works and what type of methods are supported.

Understanding the Solution Framework (cont.)

The screenshot displays the Adeptia Integration Suite interface. The top section shows a 'Design > Process Flow' view with a table of data flows. A red circle highlights a specific flow, and red arrows point from it to a 'Web Services' configuration window. The 'Web Services' window shows a list of configured consumers, including NS_AddWS, NS_SearchMore, NS_UpdateWS, NS_Search, NS_LogoutWS, and NS_LoginWS.

| # | Name | Description | Style | Owner |
|---|---------------|--|----------|--------------|
| 1 | NS_AddWS | Web service to add leads on Netsuite | document | NetSuiteUser |
| 2 | NS_SearchMore | Makes the Search More Web service Call to NetSuite | document | NetSuiteUser |
| 3 | NS_UpdateWS | To Update Existing Leads in Netsuite | document | NetSuiteUser |
| 4 | NS_Search | Makes the Search Web service Call to NetSuite | document | NetSuiteUser |
| 5 | NS_LogoutWS | Logout request to netsuite | document | NetSuiteUser |
| 6 | NS_LoginWS | Web service to login in NetSuite | document | NetSuiteUser |

Solution Framework:

Communication between Adeptia and NetSuite occurs through Web Services. Go to Configure > Services > Web Service > Consumer to view the different Web Service consumer calls Adeptia makes to NetSuite ranging from Adding/Updating data to NetSuite and Searching data from NetSuite. These are pre-defined Web Service Consumer activities which are included within Adeptia's NetSuite solution and you do not need to make any changes here.

Exercise 1: Searching new Modified Contacts

The screenshot displays the Adeptia Integration Suite interface. The left sidebar shows a navigation tree with 'Services' expanded to 'Extensions', where 'Custom Plugin' is selected. The main area shows the configuration for the 'PA_Contacts_GenerateSearchMessage' plugin. A table lists various plugins, with the selected one highlighted. Below the table, the 'Standard properties' section shows the plugin's name and description. The 'Script' section contains a Java code snippet for generating a SOAP search request.

Standard properties

| # | Name | Description | Owner | Perm. | Modified |
|----|-----------------------------------|---|--------------|-------|----------------|
| 1 | PA_Contacts_GenerateSearchMessage | Generates input to the search Contact from NetSuite | NetSuiteUser | RWX | 10/27/11 13:10 |
| 2 | PA_SalesOrder_C | | | | |
| 3 | PA_kitItem_Gene | | | | |
| 4 | PA_Opportunity_I | | | | |
| 5 | PA_InventoryIter | | | | |
| 6 | PA_ServiceSaleIt | | | | |
| 7 | PA_discountItem | | | | |
| 8 | PA_CreditMemos | | | | |
| 9 | PA_CashSales_G | | | | |
| 10 | PA_CashRefund_I | | | | |
| 11 | PA_AssemblyIter | | | | |

Script

```
StringBuffer buffer = new StringBuffer();
buffer.append("<?xml version='1.0' encoding='UTF-8'?>");
buffer.append("<platformMsgs:search
xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:s0='urn:relationships_2010_2.lists.webservices.netsuite.com'
xmlns:s1='urn:common_2010_2.platform.webservices.netsuite.com'
xmlns:platformMsgs='urn:messages_2010_2.platform.webservices.netsuite.com'>");
buffer.append("<platformMsgs:searchRecord xsi:type='s0:ContactSearch'>");
buffer.append("<s0:basic>");
buffer.append("<s1:lastModifiedDate operator='within'>");
buffer.append
("<platformCore:predefinedSearchValue>thisWeek</platformCore:predefinedSearchValue>");
buffer.append("</s1:lastModifiedDate>");
buffer.append("</s0:basic>");
buffer.append("</platformMsgs:searchRecord >");
buffer.append("</platformMsgs:search>");
service.write(buffer.toString().getBytes(), "default");
```

Execute the Search_Contact_From_NetSuite flow:

1. The search criteria for getting data from NetSuite of a particular data object such as Contacts, Customers etc are stored in a custom plugin.
2. Go to Configure > Services > Extensions and select the Contacts_GenerateSearchMessage plugin as shown in the image above.
3. In this plugin there are three parameters that we can edit to modify our search request:
 - Search basis and the operator: Example 'lastModifiedDate' and 'within'. For instance if we need to get all the records that were modified within a certain time period.
 - Condition value: Example 'thisWeek'. For instance get all modified Contacts that were modified within this week.
4. To learn more about the different criteria supported by NetSuite please check the next page.

Exercise 1: Searching new Modified Contacts (cont.)

Options that can be set in the Search custom plugins.

Example:

```
lastModifiedDate
within
thisWeek
```

Refer to notes

7

© Adeptia, Inc. All rights reserved.
Confidential & Proprietary Document.

| S.No. | SearchDateFieldOperator | SearchDateCriteria |
|-------|-------------------------|----------------------------------|
| 1 | after | lastBusinessWeek |
| 2 | before | lastFiscalQuarter |
| 3 | empty | lastFiscalQuarterToDate |
| 4 | notAfter | lastFiscalYear |
| 5 | notBefore | lastFiscalYearToDate |
| 6 | notEmpty | lastMonth |
| 7 | notOn | lastMonthToDate |
| 8 | notOnOrAfter | lastRollingQuarter |
| 9 | notOnOrBefore | lastRollingYear |
| 10 | notWithin | lastWeek |
| 11 | on | lastWeekToDate |
| 12 | onOrAfter | nextBusinessWeek |
| 13 | onOrBefore | nextFiscalQuarter |
| 14 | within (till 49 below) | nextFiscalYear |
| 15 | | nextFourWeeks |
| 16 | | nextMonth |
| 17 | | nextOneMonth |
| 18 | | nextOneQuarter |
| 19 | | nextOneWeek |
| 20 | | nextOneYear |
| 21 | | nextWeek |
| 22 | | previousOneDay |
| 23 | | previousOneMonth |
| 24 | | previousOneQuarter |
| 25 | | previousOneWeek |
| 26 | | previousOneYear |
| 27 | | previousRollingQuarter |
| 28 | | previousRollingYear |
| 29 | | sameMonthLastFiscalQuarter |
| 30 | | sameMonthLastFiscalQuarterToDate |
| 31 | | sameMonthLastFiscalYear |
| 32 | | sameMonthLastFiscalYearToDate |
| 33 | | sameQuarterLastFiscalYear |
| 34 | | sameQuarterLastFiscalYearToDate |
| 35 | | thisBusinessWeek |
| 36 | | thisFiscalQuarter |
| 37 | | thisFiscalQuarterToDate |
| 38 | | thisFiscalYear |
| 39 | | thisFiscalYearToDate |
| 40 | | thisMonth |
| 41 | | thisMonthToDate |
| 42 | | thisRollingQuarter |
| 43 | | thisRollingYear |
| 44 | | thisWeek |
| 45 | | thisWeekToDate |
| 46 | | thisYear |
| 47 | | today |
| 48 | | tomorrow |
| 49 | | yesterday |

EPTIA

Search criteria consists of three parts:

1. Search basis: This forms the basis of your search, for example I want all the last modified Contacts
2. Operator: This forms the operational attribute of the condition such as 'within' >> I want all the last modified Contacts within certain date/time.
3. Criteria value: This forms the time constraint of your search condition >> I want all the last modified Contacts within this week.

Note that all the search criteria values are case sensitive and must follow the format as listed in the table above. Adeptia can provide more

details on the search logic if further clarification is needed. To find out how to get the search base criteria other than lastModified, just click on the field in the NetSuite

GUI and you will see the technical field name that you can use as part of the search criteria. For example if you click on Date Created the Field ID is datecreated.

Now

you can use this field name as part of your search.

Exercise 1: Searching new Modified Contacts (cont.)

Running the process flow to get all the Contacts that were modified this week. See notes.

The screenshot displays the NetSuite interface for editing a contact. The contact details for Allister Sullivan at Sullivan Distributors, Inc. are shown, including job title (Sales Rep), main phone (800-555-2830), and address (1500 Central Pkwy, Chicago, IL 60610). Below this, the Adeptia integration interface is visible, showing a list of process flows. The 'Design > Process Flow' section is active, and the 'PA_SearchContact_From_NetSuite' process flow is selected and highlighted. A text box on the right side of the screenshot reads: 'Integration Make changes to NetSuite record and then run the flow in Adeptia.' A 'Refer to notes' button is also present in the Adeptia interface.

| # | Name | Description | Owner | Perms | M... |
|----|---|--|--------------|-------|----------------|
| 13 | PA_SearchInventoryItem_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 09/20/11 14:43 |
| 14 | PA_SearchItemsFrom_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/20/11 14:18 |
| 15 | PA_SearchOpportunity_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/20/11 12:49 |
| 16 | PA_SearchServiceSalesItem_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/20/11 12:23 |
| 17 | PA_SearchInvoices_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/20/11 11:30 |
| 18 | PA_SearchAssemblyItem_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/19/11 18:51 |
| 19 | PA_SearchCashRefund_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/19/11 18:27 |
| 20 | PA_SearchSalesOrder_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/19/11 17:26 |
| 21 | PA_SearchCustomer_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/18/11 13:13 |
| 22 | PA_SearchContact_From_NetSuite | Pull Data from Netsuite and Migrate to Database Tables | NetSuiteUser | RWX | 07/14/11 15:25 |
| 23 | PA_PtoSyncDatabase_with_NS | ProcessFlow to Synchronized database with Netsuite | NetSuiteUser | RWX | 01/21/09 13:12 |
| 24 | T_XYZObject_From_NetSuite | Pull Data from Netsuite to Internal System | NetSuiteUser | RWX | 04/16/08 10:00 |

Login to your NetSuite account and go to Lists > Relationships > Contacts > Search > Edit any Contact (such as Address, Title etc.).

We will then retrieve this modified record when we execute the below process flow. Go to Design > Process Flow > Select the SearchContact_From_NetSuite and click on execute. And then click on view process flow logs.

Exercise 1: Searching new Modified Contacts (cont.)

Checking the results of the transaction execution. See notes.

Refresh | View Process flow log

| Activity Name | Activity Type | Status | UserID | Start Time | End Time | Action |
|-----------------------------------|------------------------|----------|--------------|---------------------|---------------------|---|
| NS_LogoutWS | WsMessageCall | Executed | NetSuiteUser | 10/18/2011 14:19:00 | 10/18/2011 14:19:00 | Details Diagnostics |
| Scripted-Guard | Condition | Executed | NetSuiteUser | 10/18/2011 14:19:00 | 10/18/2011 14:19:00 | Details Diagnostics |
| PA_ContactDataTarget | AdvancedDatabaseTarget | Executed | NetSuiteUser | 10/18/2011 14:18:59 | 10/18/2011 14:19:00 | Details Diagnostics |
| PA_ContactSearch_Resp | DataMapping | Executed | NetSuiteUser | 10/18/2011 14:18:57 | 10/18/2011 14:18:59 | Details Diagnostics |
| NS_Search | WsMessageCall | Executed | NetSuiteUser | 10/18/2011 14:18:56 | 10/18/2011 14:18:57 | Details Diagnostics |
| PA_Contacts_GenerateSearchMessage | CustomPlugin | Executed | NetSuiteUser | 10/18/2011 14:18:55 | 10/18/2011 14:18:56 | Details Diagnostics |
| Login | Call Tag | Executed | NetSuiteUser | 10/18/2011 14:18:50 | 10/18/2011 14:18:55 | Details Diagnostics |
| PA_SearchContact_From_NetSuite | Transaction | Executed | NetSuiteUser | 10/18/2011 14:18:49 | 10/18/2011 14:19:02 | Details Diagnostics |

Process Flow Log Details

| Process Flow Log Details | | | | | | | |
|--|--------------------------------|---------------------|---------------------|---------------------|----------|--------------------------------|---------------|
| Process Flow Name | PA_SearchContact_From_NetSuite | | | | | | |
| Process Flow Id | 192168001137121517461268700003 | | | | | | |
| Process Flow PID | 192168005062131896552949501181 | | | | | | |
| Status | Executed | | | | | | |
| Start Time | 10/18/2011 14:18:49 | | | | | | |
| End Time | 10/18/2011 14:19:02 | | | | | | |
| Triggered By (User Name) | NetSuiteUser | | | | | | |
| Triggered By (User Id) | 192168001041120574962729600003 | | | | | | |
| Triggered Event Type | Manual | | | | | | |
| Process Flow Execution Time | 17 second (s) 781 ms | | | | | | |
| Activity Name (ID) | Activity Type | Start Time | End Time | Execution Time | Status | Processed Data (Bytes/Records) | Error Records |
| NS_LogoutWS (192168001012120047815709300006) | WsMessageCall | 10/18/2011 14:19:00 | 10/18/2011 14:19:00 | 0 ms | Executed | 0 | 0 |
| Scripted-Guard (NA) | Condition | 10/18/2011 14:19:00 | 10/18/2011 14:19:00 | 15 ms | Executed | 0 | 0 |
| PA_ContactDataTarget (192168001137121517461268700004) | AdvancedDatabaseTarget | 10/18/2011 14:18:59 | 10/18/2011 14:19:00 | 1 second (s) 61 ms | Executed | 2 | 0 |
| PA_ContactSearch_Resp (192168001137121517461268700005) | DataMapping | 10/18/2011 14:18:57 | 10/18/2011 14:18:59 | 1 second (s) 202 ms | Executed | 2321 | 0 |
| NS_Search (1921680012021212130) | WsMessageCall | 10/18/2011 14:18:56 | 10/18/2011 14:18:57 | 1 second (s) 14 ms | Executed | 0 | 0 |
| PA_Contacts_GenerateSearchMessage (NA) | CustomPlugin | 10/18/2011 14:18:55 | 10/18/2011 14:18:56 | 1 second (s) 30 ms | Executed | 0 | 0 |
| Login (NA) | Call Tag | 10/18/2011 14:18:50 | 10/18/2011 14:18:55 | 5 second (s) 519 ms | Executed | 0 | 0 |

[View details](#)

Refer to notes

9

© Adeptia, Inc. All rights reserved.
Confidential & Proprietary Document.



Login to your NetSuite account and go to Lists > Relationships > Contacts > Search > Edit any Contact (such as Address, Title etc.).

We will then retrieve this modified record when we execute the below process flow. Go to Design > Process Flow > Select the SearchContact_From_NetSuite and click on execute. And then click on view process flow logs.

Process Flow logs will show the status of every step in descending order (latest step on top). You can click on the Details link to see further details of the transaction. To view the actual data in XML, you can click on the Repository link in the Details page and view the SOAP XML files.

Exercise 1: Searching new Modified Contacts (cont.)

Checking the results in the backend database

The screenshot shows a SQL Server Enterprise Manager interface. On the left, a tree view displays a list of database tables, with 'dbo.Lists_Contact' selected and marked with a checkmark. The main window shows a SQL query window with the following text:

```
SELECT  
    , [companyInternalID]  
    , [salutation]  
    , [firstName]  
    , [middleName]  
    , [lastName]  
    , [title]  
    , [phone]
```

A yellow box labeled 'Result' points to the query text. Below the query window, a 'Results' grid displays the following data:

| entityID | company | companyInternalID | salutation | firstName | middleName | lastName | title |
|----------|-------------------|-----------------------------|------------|-----------|------------|----------|-----------|
| 1 | Alistair Sullivan | Sullivan Distributors, Inc. | 43 | Mr. | Alistair | Sullivan | Sales rep |

Below the 'Results' grid, a 'Messages' grid shows the following message:

| id | date | lastModifiedDate | modifyBy | ErrorMessage | msrepl_tran_version |
|----|-------------------|-------------------------|-----------------------|--------------|--------------------------------------|
| 1 | 7-25 02:00:00.000 | 2011-10-18 14:18:01.000 | SynchronizedByAdeptia | ✓ | 841C6376-57CF-4C40-96AD-EF8D6B875373 |

A yellow box labeled 'Refer to notes' is located at the bottom left of the screenshot area.

The database records in the table in Lists_Contact will now contain all the records from NetSuite modified this week. In our example, we edited

Alistair Sullivan's Title to Sales Rep and now after running the process flow the database contains that record with all the fields.

In addition to the record fields, Adeptia has also added fields such as 'modifyBy', 'LastModifiedDate' and 'ErrorMessage' to help keep track of the status.

The status after loading the data into the table shows 'SynchronizedByAdeptia' meaning that database is now in-synch with NetSuite in regards to that record.

When loading data into NetSuite, these fields help identify if any record fails to get uploaded into NetSuite due to any errors. Users can correct those errors and re-run the flow.

Exercise 2: Loading new/updated data records

Update/insert Contact records into database and change their status to 'ModifiedByWeb' in modifyBy field.

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the server tree is expanded to show the 'dbo.Lists_Contact' table. The main window displays a SQL query window with the following query:

```
SELECT TOP 1000 (NetSuiteInternalID),
(customForm),
(customFormInternalID),
(entityID),
(company),
(companyInternalID),
(salutation),
(firstName),
(middleName),
(lastName),
(title)
FROM Lists_Contact
```

The Results window shows the following data:

| entityID | company | companyInternalID | salutation | firstName | middleName | lastName | title |
|----------|-------------------|-----------------------------|------------|-----------|------------|----------|--------------|
| 1 | Allister Sullivan | Sullivan Distributors, Inc. | 43 | Mr. | Allister | Sullivan | BusinessUser |

Below the Results window, a table shows the record details:

| dateCreated | lastModifiedDate | modifyBy | ErrorMessage | msrepl_tran_version |
|-------------|-------------------------|-------------------------|---------------|--------------------------------------|
| 1 | 2006-07-25 02:00:00.000 | 2011-10-18 14:18:01.000 | ModifiedByWeb | 841C6376-57CF-4C40-96AD-EF8D6B875373 |

Refer to notes

Update the modifyBy field to 'ModifiedByWeb'. This flag enables Adeptia's process to pick up the changed data and load it into NetSuite.

11

© Adeptia, Inc. All rights reserved.
Confidential & Proprietary Document.



Update or Insert data into the database table. For now you can update the existing record in the Lists_Contact table and make changes to some of the values.

Adeptia identifies the records that need to be picked up by flag set in the modifyBy field. If it contains 'ModifiedByWeb' then that record will be picked up as part of the database extraction and will be loaded into NetSuite.

Such as update the data record by running these two SQL queries:

```
UPDATE NetSuite.dbo.Lists_Contact
SET Title = 'BusinessUser'
WHERE NetSuiteInternalID = 42
GO
```

The above query will change the Title of this record to BusinessUser.

```
UPDATE NetSuite.dbo.Lists_Contact
SET modifyBy = 'ModifiedByWeb'
WHERE NetSuiteInternalID = 42
GO
```

The above query will change the modifyBy to 'ModifiedByWeb'.

Exercise 2: Loading new/updated data records (cont.)

Now run the process PA_ContactUpdateList_To_NetSuite flow which will pick data from database and load it into NetSuite.

| # | Name | Description | Owner | Perm |
|---|--------------------------------------|---|--------------|------|
| 1 | PA_SalesOrderAddList_To_Netsuite | Push new records from Database to Netsuite | NetSuiteUser | RWX |
| 2 | PA_CustomerAddList_To_Netsuite | Push new records from Database to Netsuite | NetSuiteUser | RWX |
| 3 | PA_ContactAddList_To_Netsuite | Push new records from Database to Netsuite | NetSuiteUser | RWX |
| 4 | PA_UpsertRecords_To_NetSuite | ProcessFlow to Update and add records to Netsuite | NetSuiteUser | RWX |
| 5 | PA_ContactUpdateList_To_Netsuite | Push Updated data from Database to Netsuite | NetSuiteUser | RWX |
| 6 | PA_OpportunityUpdateList_To_Netsuite | Push Updated data from Database to Netsuite | NetSuiteUser | RWX |
| 7 | PA_OpportunityAddList_To_Netsuite | Push new records from Database to Netsuite | NetSuiteUser | RWX |

| Activity Name | Activity Type | Status | UserID | Start Time | End Time | Action |
|-----------------------------------|------------------------|----------|--------------|---------------------|---------------------|---------------------|
| PA_MailNotificationFor_PullPFs | MailNotification | Running | NetSuiteUser | 10/18/2011 15:34:29 | NA | Details Diagnostics |
| NS_LogoutWS | WsMessageCall | Executed | NetSuiteUser | 10/18/2011 15:34:29 | 10/18/2011 15:34:29 | Details Diagnostics |
| PA_DBTarget_UpdateListContact | DataMapping | Executed | NetSuiteUser | 10/18/2011 15:34:29 | 10/18/2011 15:34:29 | Details Diagnostics |
| PA_UpdateListResponse_Contact_Map | DataMapping | Executed | NetSuiteUser | 10/18/2011 15:34:28 | 10/18/2011 15:34:29 | Details Diagnostics |
| NS_UpdateWS | WsMessageCall | Executed | NetSuiteUser | 10/18/2011 15:34:26 | 10/18/2011 15:34:28 | Details Diagnostics |
| Login | Call Tag | Executed | NetSuiteUser | 10/18/2011 15:34:20 | 10/18/2011 15:34:26 | Details Diagnostics |
| PA_ContactUpdateList_Map | DataMapping | Executed | | | | |
| Context-Guard | Condition | Executed | | | | |
| RepeaterService | RepeaterService | Executed | | | | |
| PA_DBSource_UpdateListContact | AdvancedDatabaseSource | Executed | | | | |
| PA_ContactUpdateList_To_Netsuite | Transaction | Running | | | | |

Check the process logs and view the details. You can also view the XML intermediate files by going to Repository.

Refer to notes

12

© Adeptia, Inc. All rights reserved.
Confidential & Proprietary Document.



By executing this flow, Adeptia will pick the records from the Contact (and child tables) database and will load this data into NetSuite.

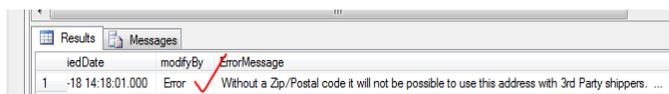
If there are any errors while loading data into NetSuite then check the Error Description field to see what the error is and how to resolve it. If for example the Error Description is saying that a particular mandatory fields value is missing than you can correct that field, change the modifyBy field to ModifiedByWeb and rerun the flow. Rest of the records that are successfully loaded into NetSuite will have their status changed to “SynchronizedByAdeptia”.

After process flow executes you can go to NetSuite and do a refresh to see the changes for that contact.

Error Handling

When inserting/updating data into NetSuite, Adeptia handles error conditions by inserting the error description into the ErrorMessage field of the database table. The message is captured from NetSuite and can be viewed by the end-user by checking the error message in the table. User can make the corrections and rerun the flow.

- dbo.KitItems_TranslationList
- dbo.Lists_Contact
- dbo.Lists_Contact_AddressBookList
- dbo.Lists_Contact_SubscriptionList



The screenshot shows a database table with columns: id, iedDate, modifyBy, and ErrorMessage. The ErrorMessage column contains the text: "Without a Zip/Postal code it will not be possible to use this address with 3rd Party shippers. ...".

| id | iedDate | modifyBy | ErrorMessage |
|----|------------------|----------|--|
| 1 | -18 14:18:01.000 | Error | Without a Zip/Postal code it will not be possible to use this address with 3rd Party shippers. ... |

In this example, the zip code is missing in the Contact's address information. Adeptia captures the error thrown by NetSuite as part of the Web Service communication and is displayed here.

[Refer to notes](#)

13

© Adeptia, Inc. All rights reserved.
Confidential & Proprietary Document.



After making corrections to your data depending upon the error message you can rerun the flow by first changing the modifyBy field to ModifiedByWeb and execute the flow. Records that are successfully loaded into NetSuite will have their status changed to "SynchronizedByAdeptia".