An Oracle Best Practice Guide  
October 2013

Setting up the integration between Oracle Social Engagement & Monitoring Cloud Service and Oracle RightNow Cloud Service

Introduction 2

Creation of the custom object required for the integration 3

Creation of the custom process required for the integration 7

Creation of a Profile that Social Engagement & Monitoring Cloud Service would use to call Oracle RightNow APIs 15

Creation of a Staff Account that Social Engagement & Monitoring Cloud Service would use to call Oracle RightNow APIs 17

Setting up Oracle RightNow Cloud Service plug-in parameters in the Social Engagement & Monitoring Cloud Service admin screen 19

Setting up Twitter channel Accounts in Oracle RightNow Social Monitor 19

Conclusion 20

# Introduction

This white paper provides the step-by-step instructions to set up the integration between Oracle Social Engagement & Monitoring Cloud Service and Oracle RightNow Cloud Service.

**Note**: The step “Creation of the custom process” in this white paper contains sample code that should be considered as examples and is not part of the RightNow product or the APIs that are supported by Oracle RightNow Customer Support.

Before we get into the integration, here are the Oracle RightNow Cloud Service version requirements for this integration:

1. The February 2012 release can receive social posts from Social Engagement & Monitoring Cloud Service but will not update the status of the post in Social Engagement & Monitoring Cloud Service when an incident is closed.
2. On the May 2012 release and after, the Custom Process Models (CPM) feature can be used in order to better leverage all the features of the integration.
3. On releases prior to February 2012, it is recommended to first upgrade to the November 2012 release or later.

After implementing this white paper you will be able to:

1. Manually escalate a social post from Social Engagement & Monitoring Cloud Service to Oracle RightNow Cloud Service Incident.
2. Respond to twitter posts using Oracle RightNow Social Monitor.
3. See the updates in the Social Engagement & Monitoring Cloud Service post’s activity history when an incident is closed.

Described below is the overview of the flow of data between Social Engagement & Monitoring Cloud Service and Oracle RightNow Cloud Service. When a Social Engagement & Monitoring Cloud Service user clicks on the menu item Send to RightNow:

1. The system checks if the post author already exists as a contact in Oracle RightNow Cloud Service. This is done by checking if there is any contact with the same social network handle as the post author.
2. If a contact is found, the system would create an incident for that contact.
3. If a contact is not found in Oracle RightNow Cloud Service, a contact is created in the following situations:
   1. If the channel corresponding to the post is Twitter, Social Engagement & Monitoring Cloud Service creates a contact with the first and last name as the post author’s twitter handle. This is because Twitter does not expose the first and last name publicly.
   2. If the channel corresponding to the post is a Facebook fan page, a contact is created with the appropriate first and last name from Facebook
4. For all other scenarios (Social Engagement & Monitoring Cloud Service Listen Topic feed, feeds from other channels like RSS), a sparse contact with the name OSEM Contact is used.
5. After the contact identification and creation process, an incident is created in Oracle RightNow Cloud Service for that contact. When creating the incident, the incident subject is populated with the post content. Other than the subject, incident thread(s) are also created with the post content and URL.
6. After the creation of the incident, an agent picks up the incident and works on it. Once the incident is resolved, when the agent sets the status of the incident to Solved, the system will update the corresponding post’s activity history in Social Engagement & Monitoring Cloud Service, to indicate that the incident was closed.

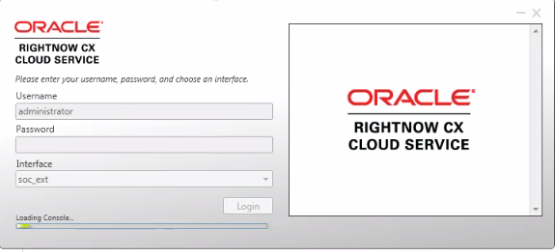
This setup process covers the following topics:

1. Creation of the custom object required for the integration
2. Creation of the custom process to update Social Engagement & Monitoring Cloud Service once the incident is closed
3. Setting up of Oracle RightNow Cloud Service plug-in parameters in the Social Engagement & Monitoring Cloud Service admin screen
4. Setting up Twitter channel Accounts in Oracle RightNow Social Monitor

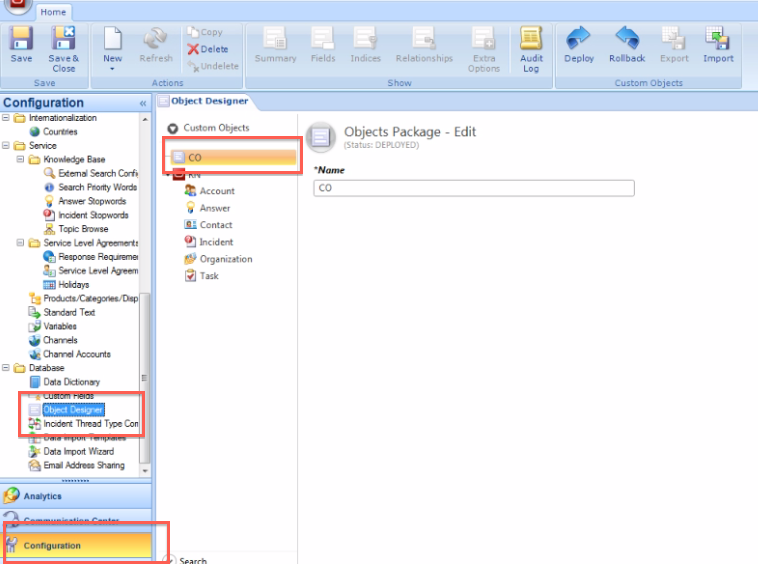
# Creation of the custom object required for the integration

This section will guide you through the process of creating the custom object required for this integration.

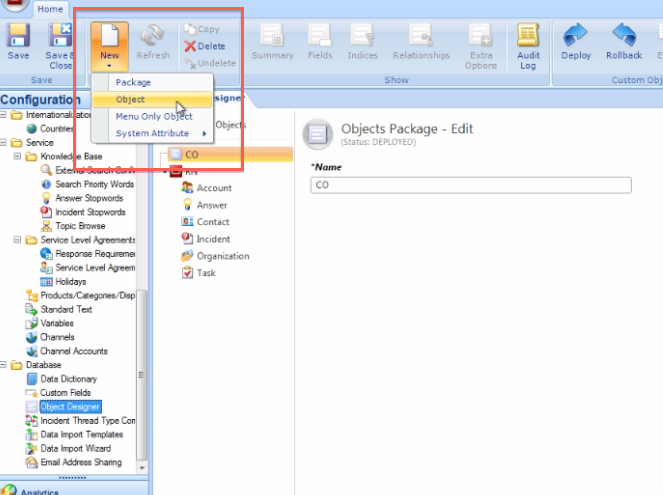
1. Login to Oracle RightNow Cloud Service using an account with administrator privileges.



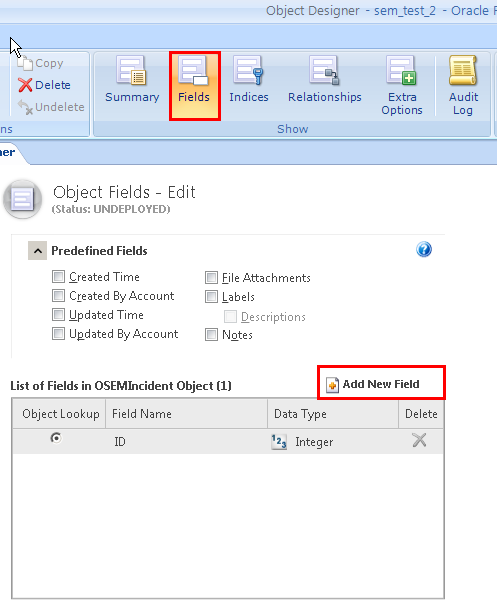
1. To create the custom object, navigate to Configuration> Database > Object Designer and double click to open the list of objects. In the Object Designer tab, select the node for CO (referring to *custom objects*).
2. Select the CO Package. If the CO Package does not exist, create a new CO package.



1. Create a new object in the CO Package.



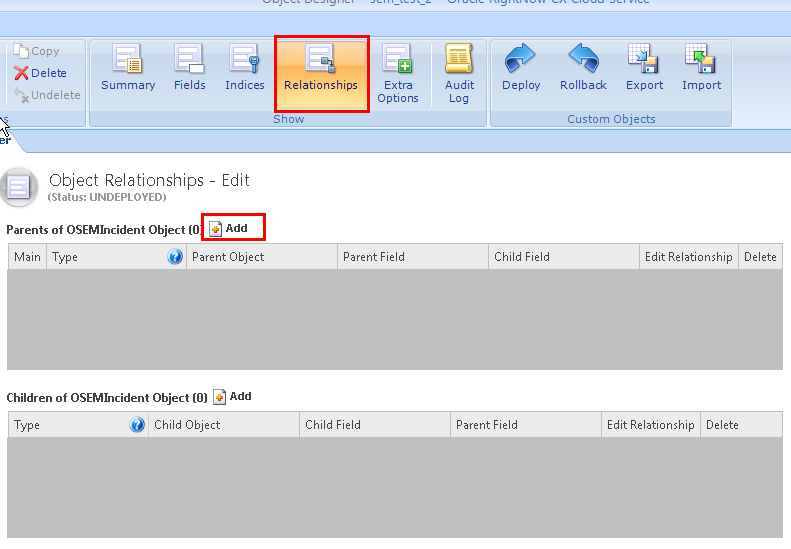
1. Set the Object name and other attributes as follows:
   1. Name = OSEMIncident
   2. Object is Available in Workspaces, Scripting and Workflow = Checked
   3. Object is Enabled for Transactions = Checked
   4. Label = 'OSEMIncident'
2. Add fields to the OSEMIncident Object.



1. Make sure that the fields have the following attributes when creating them:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Options** | **Integer/Text Field Settings** |
| ID | System created field. Leave as default | | |
| IncidentID | Integer | Is In single Field Index = checked  Other Options= Default | All Settings = Default |
| OSEMEndpoint | Text | All Options = Default | Field Usage = Plain Text  Length of Field = 255  All Other Settings = default |
| OSEMAuthToken | Text | All Options = Default | Field Usage = Plain Text  Length of Field = 255  All Other Settings = default |
| OSEMServiceRequestID | Text | All Options = Default | Field Usage = Plain Text  Length of Field = 255  All Other Settings = default |
| OSEMMessageID | Text | All Options = Default | Field Usage = Plain Text  Length of Field = 255  All Other Settings = default |

1. Save your changes and click on the Relationship ribbon button to add a new relationship with the Incident Object so that Incident is the parent of OSEMIncident.



1. Make sure that the relationship has the following attributes set in the Parents of OSEMIncident Object section:

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Parent Object | Parent Field | Child Field |
| Association | RN.Incident | i\_id | IncidentID |

1. Deploy your new custom Object.

# Creation of the custom process required for the integration

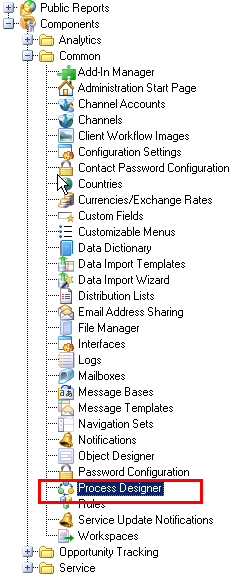
This section will guide you through the process of creating the custom process required for this integration. This custom process is used to call the Social Engagement & Monitoring API when the incident is closed. When this API is executed, the internal activity for the Social Engagement & Monitoring post will be updated to indicate that the related Incident is closed.

**Note**: It is not required to deploy this custom script if you only want to escalate a post as an incident and do not want to report back to Social Engagement & Monitoring Cloud Service if the incident was closed.

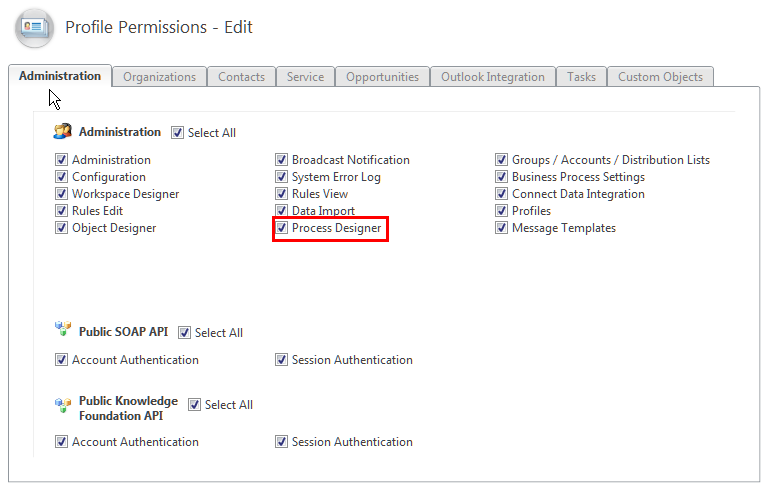
**Note**: Please be aware of the following limitations before deploying the custom script:

* 1. If for some reason the API call made to Social Engagement & Monitoring Cloud Service from this script fails, it will not be retried. The post which created the incident will not be updated to reflect the incident status change to Closed. Moreover, at present there is no manual process to update the post Social Engagement & Monitoring Cloud Service Engage post internal activity.
  2. Oracle RightNow Cloud Service expects all synchronous custom process scripts to be completed within 5 seconds to prevent run-away processes. If this script does not complete in 5 seconds, it will be timed out.
  3. If this script times out multiple times within a short duration, Oracle RightNow Cloud Service will disable the script. You will need to manually re-enable it by going to Process Designer.

1. Navigate to Process Designer. You may need to add Process designer to your Navigation Set. You should find Process Designer under Components -> Common as shown below.



1. If you do not find Process Designer in your Navigation set, check the following:
   1. Process Designer is enabled for your Site. Please contact your Oracle Customer Success Manager to confirm that Process designer has been enabled for your site.
   2. Confirm that your profile has the Process Designer permission checked. This permission is under the Administrator Section.



1. Click inside the box with PHP code below. Press the keys Ctrl + A to select all the code in the box.

<?

/\*

\* CPMObjectEventHandler: osem\_callback

\* Package: RN

\* Objects: Incident

\* Actions: Update

\* Version: 1.2

\*/

use \RightNow\Connect\v1\_2 as RNCPHP;

use \RightNow\CPM\v1 as RNCPM;

class osem\_callback implements RNCPM\ObjectEventHandler {

// { Just for debugging

private static $osem\_obj = NULL;

// } Just for debugging \*/

private static function my\_post\_request($osem\_obj) {

$optional\_headers = null;

$params = array('http' => array(

'method' => 'PUT',

'content' => http\_build\_query(array('service\_request' => array('auth\_token' => $osem\_obj->OSEMAuthToken)))));

if ($optional\_headers !== null) {

$params['http']['header'] = $optional\_headers;

}

$ctx = stream\_context\_create($params);

$final\_url = $osem\_obj->OSEMEndpoint . $osem\_obj->OSEMMessageID . '/service\_requests/' . $osem\_obj->OSEMServiceRequestID;

$fp = @fopen(($final\_url), 'rb', false, $ctx);

$response = @stream\_get\_contents($fp);

@fclose(fp);

return;

}

public static function apply($run\_mode, $action, $obj, $n\_cycles) {

if ($action==RNCPM\ActionUpdate) {

// Has the ClosedTime changed?

if ($obj->ClosedTime != $obj->prev->ClosedTime) {

if (is\_null($obj->ClosedTime)) {

// The Incident was re-opened

// Is there anything to do when the incident is re-opened?

} else {

// The Incident was closed

$query = "SELECT CO.OSEMIncident FROM CO.OSEMIncident WHERE CO.OSEMIncident.IncidentID = {$obj->ID}";

$osem\_obj = RNCPHP\ROQL::queryObject($query)->next()->next();

if (!is\_null($osem\_obj)) {

if (RNCPM\RunModeLive == $run\_mode) {

osem\_callback::my\_post\_request($osem\_obj);

} else {

echo("In Test Mode ID='{$obj->ID}', IncidentID: {$osem\_obj->IncidentID->ID}\n");

$inv\_id = "OSEM:{$obj->ID}";

if ($inv\_id === $osem\_obj->OSEMMessageID && $osem\_obj->OSEMAuthToken === "secret\_key" && $osem\_obj->OSEMServiceRequestID === "111" && $osem\_obj->OSEMEndpoint === "testurl.com") {

$obj->Subject = "TestSuccess";

}

}

}

}

}

// { Add a private note to the Incident with some debugging info

// when the subject begins with "Involver test"

if (0 === strpos($obj->Subject, 'Involver test')) {

if (is\_null($obj->Threads)) {

$obj->Threads = new RNCPHP\ThreadArray;

}

$note = new RNCPHP\Thread;

$note->EntryType->ID = 1;

$note->Text = static::$debug\_text;

$obj->Threads[] = $note;

$obj->save(RNCPHP\RNObject::SuppressAll);

// Clean up after the debugging/invented OSEMIncident object

if (!is\_null(static::$osem\_obj)) {

static::$osem\_obj->destroy();

static::$osem\_obj = NULL;

}

}

// } Add a private note to the Incident with some debugging info \*/

}

return;

}

}

class osem\_callback\_TestHarness implements RNCPM\ObjectEventHandler\_TestHarness {

static $incident = array();

public static function setup()

{

// Create Primary Contact

$contact = new RNCPHP\Contact();

$contact->Emails = new RNCPHP\EmailArray();

$contact->Emails[0] = new RNCPHP\Email();

$contact->Emails[0]->AddressType=new RNCPHP\NamedIDOptList();

$contact->Emails[0]->AddressType->LookupName = "Email - Primary";

$contact->Emails[0]->Address = "bsmith@test.net";

$contact->Login = "bsmith";

$contact->Name = new RNCPHP\PersonName();

$contact->Name->First = "Bill";

$contact->Name->Last = "Smith";

$contact->save();

// Update Two Incidents

$inc1 = new RNCPHP\Incident();

$inc1->Subject = 'This is a post from twitter';

$inc1->PrimaryContact = $contact;

$inc1->StatusWithType = new RNCPHP\StatusWithType();

$inc1->StatusWithType->Status = new RNCPHP\NamedIDOptList();

$inc1->StatusWithType->Status->ID = 2;// ID 2 is for lookup name "Solved"

$inc1->save();

$osem\_inc1 = new RNCPHP\CO\OSEMIncident();

$osem\_inc1->OSEMMessageID = "OSEM:{$inc1->ID}";

$osem\_inc1->IncidentID = $inc1;

$osem\_inc1->OSEMAuthToken = 'secret\_key';

$osem\_inc1->OSEMServiceRequestID = '111';

$osem\_inc1->OSEMEndpoint = 'testurl.com';

$osem\_inc1->save();

$inc2 = new RNCPHP\Incident();

$inc2->Subject = 'This is a post from facebook';

$inc2->PrimaryContact = $contact;

$inc2->save();

static::$incident[0] = $inc1;

static::$incident[1] = $inc2;

return;

}

public static function fetchObject($action, $object\_type ) {

return(static::$incident);

}

public static function validate($action, $object ) {

$retval = false;

if ($action==RNCPM\ActionUpdate) {

if ($object->StatusWithType->Status->LookupName == "Solved") {

if ($object->Subject == "TestSuccess") {

$retval = true;

}

} else {

if ($object->Subject != "TestSuccess") {

$retval = true;

}

}

}

if ($retval) {

echo(" Success\n");

} else {

echo(" Failure\n");

}

return $retval;

}

public static function cleanup() {

foreach (static::$incident as $inc)

{

$inc->destroy;

}

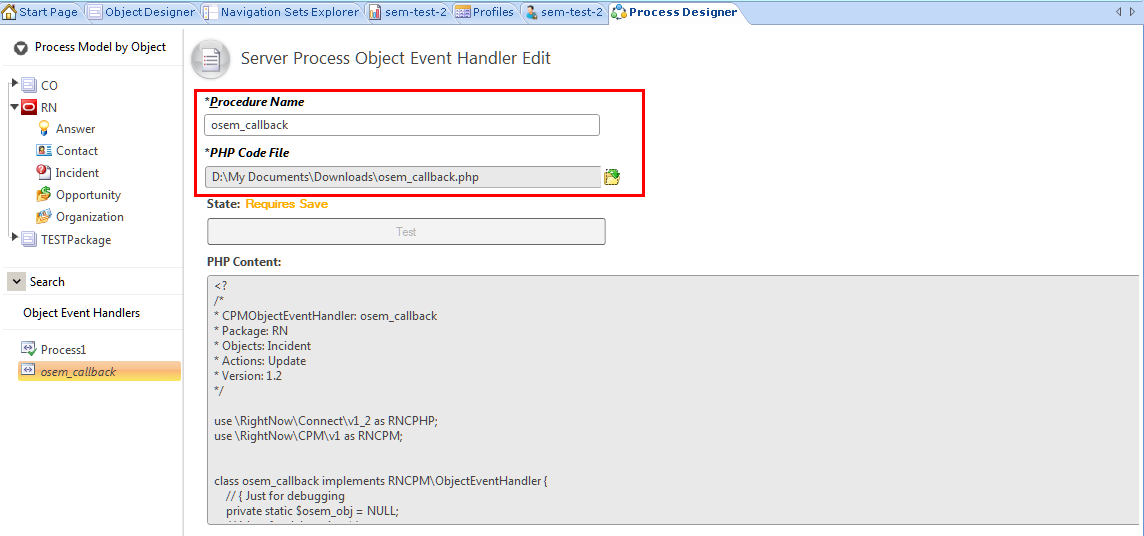
static::$incident = NULL;

return;

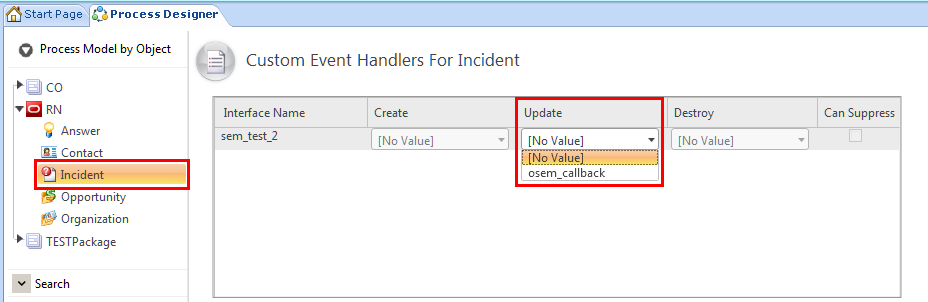
}

}

1. Now press the keys Ctrl + C to copy the code to the clipboard.
2. Open notepad and press the keys Ctrl + V to paste the code into notepad. Save the content to a file with name osem\_callback.php. Make sure you select the save as type option = All Files before you save.
3. Create a new Server Process Object Event handler with the procedure name: *osem\_callback* and upload the previously saved osem\_callback.php file.



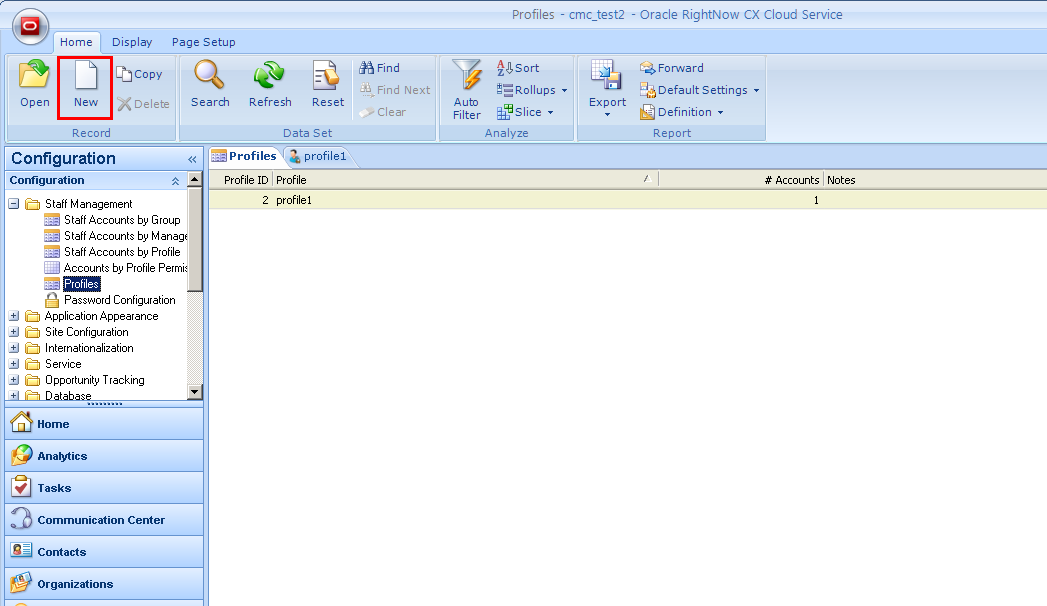
1. Save the procedure and then test the procedure by clicking on the Test button.
2. Confirm that the test result was a Success. A failed test is typically because there is an error in the script itself or an exception was not caught in the test harness of the event handler. Please verify your script and test again. More details on custom processes can be found in the Core Features > Custom Processes > Overview > Adding object event handlers section of the user guide. The latest user guide can be found at: https://cx.rightnow.com/app/answers/detail/a\_id/5168.
3. Save the procedure once again and then deploy the procedure by clicking on the deploy button.
4. Once deployed, click on RN->Incident to go to the Custom Event Handlers for the Incident Object.



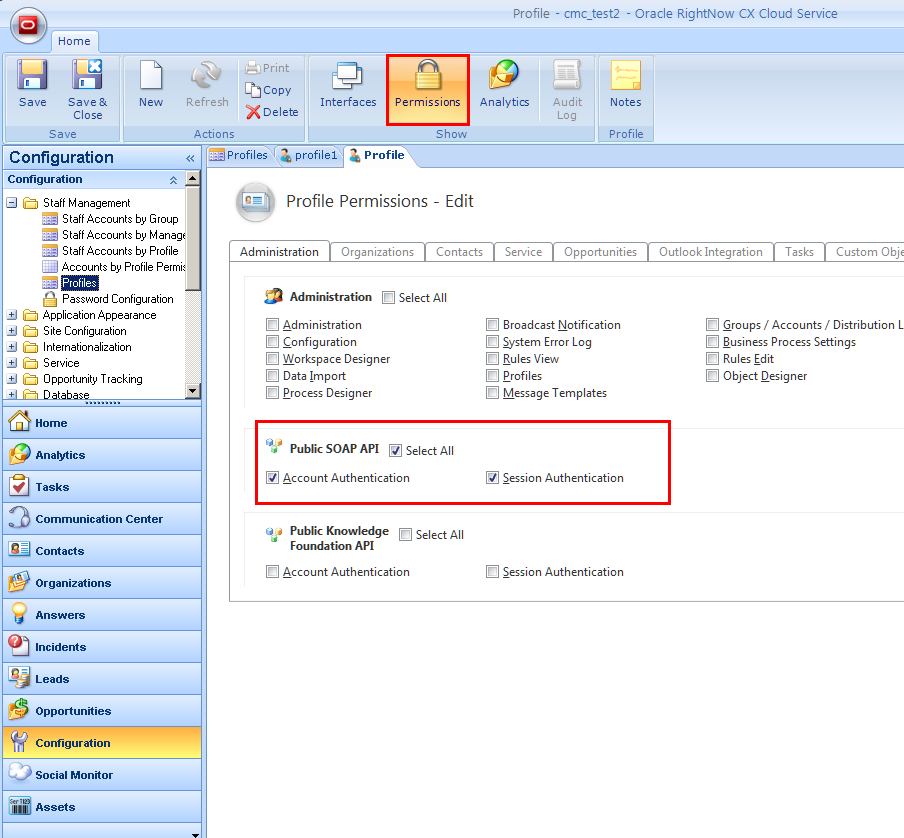
1. Select *osem\_callback* for the Update event. Save and Deploy the event handler.
2. This completes the deployment of the custom process for the integration.

# Creation of a Profile that Social Engagement & Monitoring Cloud Service would use to call Oracle RightNow APIs

1. In Oracle RightNow Cloud Service navigate to Configuration -> Staff Management -> Profiles
2. Click on ‘New’ on the ribbon bar



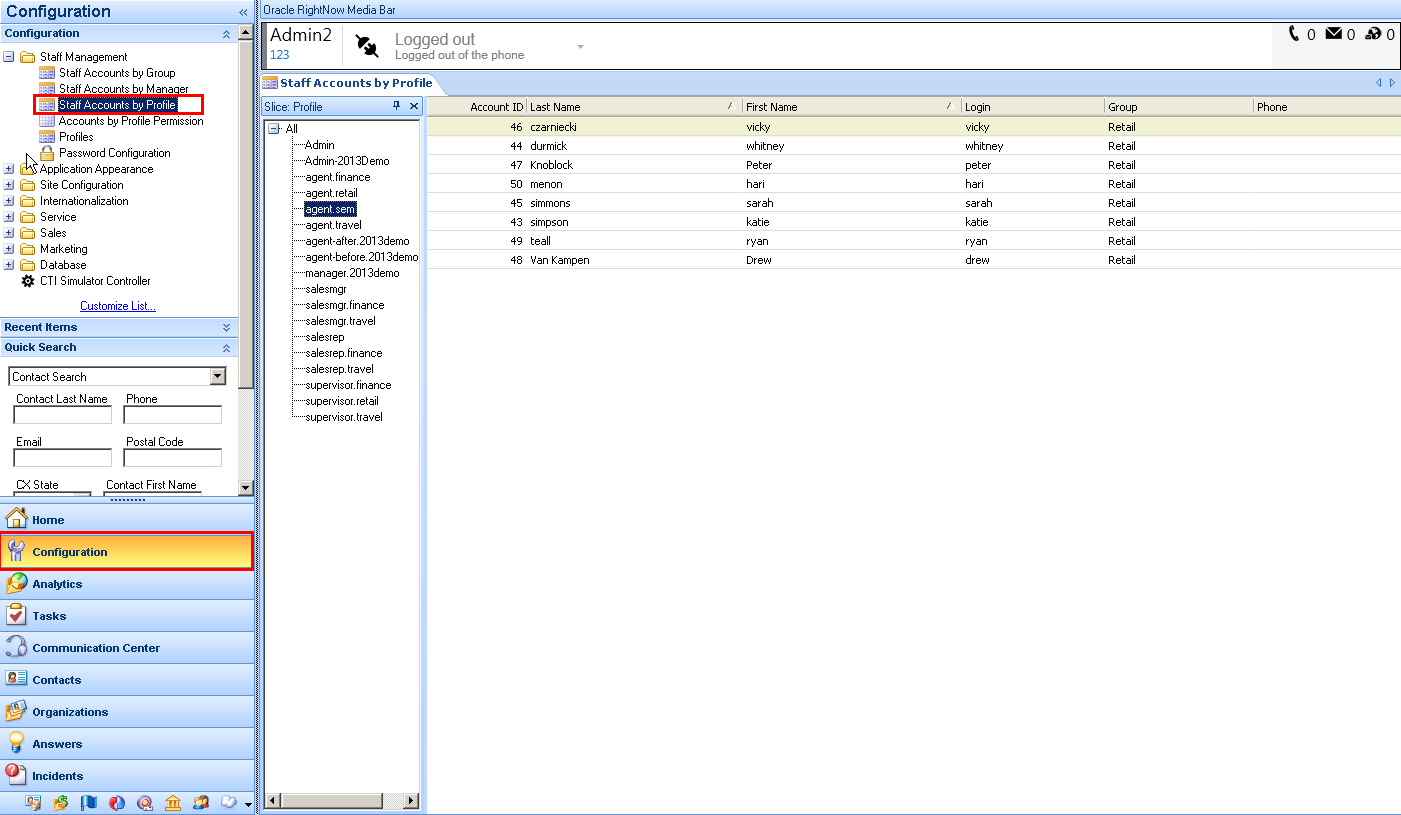
1. Create a new Profile in RightNow and fill in the appropriate fields.
2. Click on the “Permissions” button in the Ribbon bar



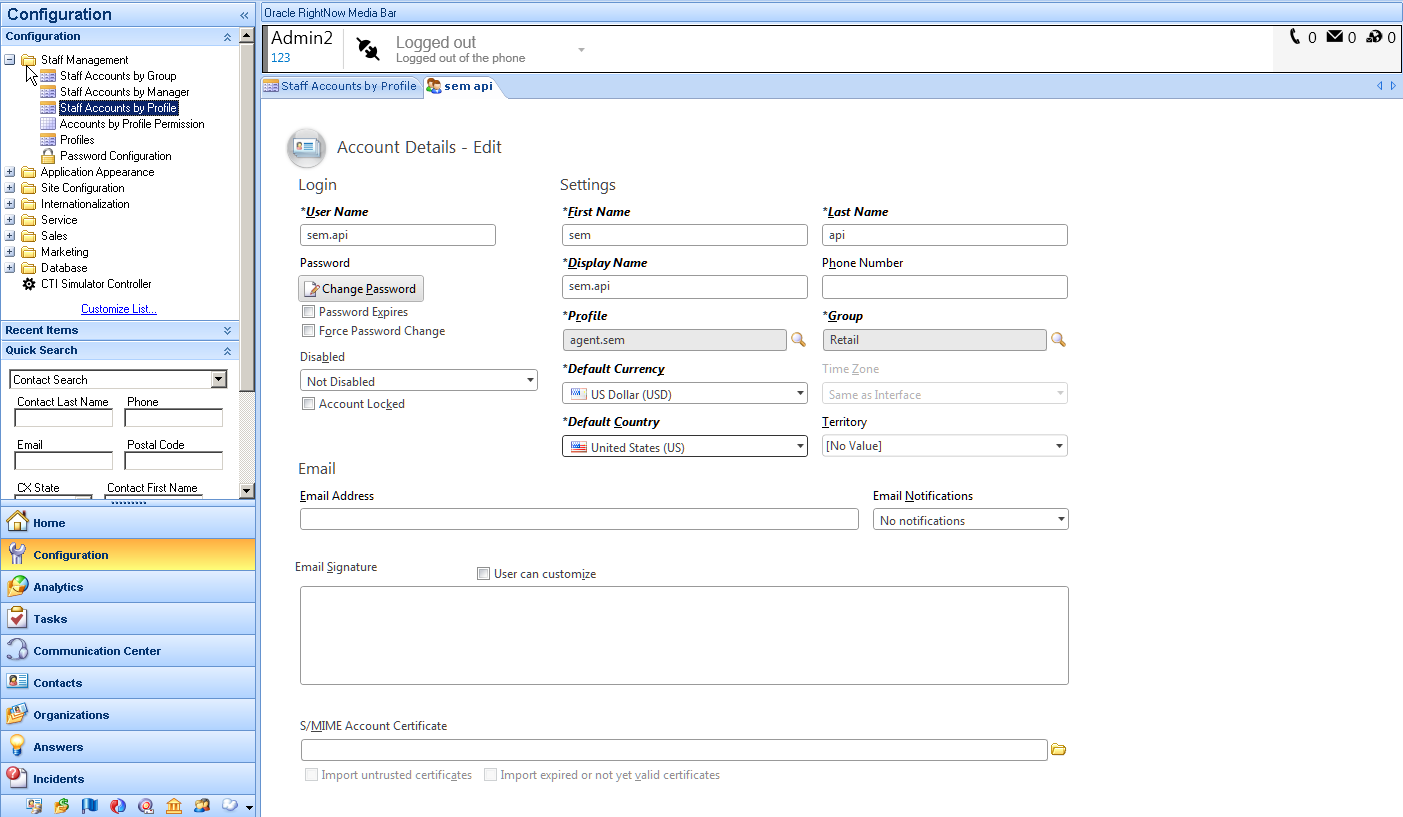
1. Make sure that the Public Soap APIs permissions are selected.
2. You can use an existing profile if required; as long as you make sure that the profile has the above Public SOAP API permissions selected.

# Creation of a Staff Account that Social Engagement & Monitoring Cloud Service would use to call Oracle RightNow APIs

1. In Oracle RightNow Cloud Service navigate to Configuration -> Staff Management -> Staff Account by Profile



1. Click on the New button in the ribbon bar to create a new staff account.

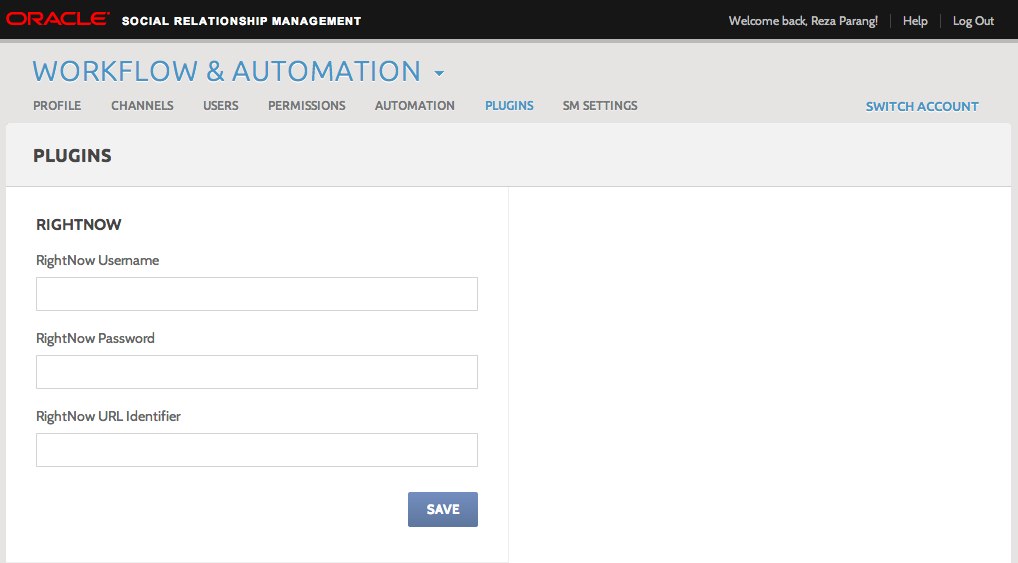


1. Enter the required fields to create a staff account and save the record
2. Make sure that for the staff account you are creating, you select the profile that you configured in the previous section.

# Setting up Oracle RightNow Cloud Service plug-in parameters in the Social Engagement & Monitoring Cloud Service admin screen

In this step you will input the end-point URL of the Oracle RightNow SOAP APIs and username/password that Social Engagement & Monitoring Cloud Service will use to call the Oracle RightNow APIs.

1. Login to Oracle Social Engagement & Monitoring Cloud Service. Navigate to Plugins sub-section of the Workflow & Automation section.



1. Populate the RightNow username, password & URL in the RightNow section of the plugins page.
2. The RightNow WSDL URL should be of the format: http://<host\_name>/cgi-bin/<interface>.cfg/services/soap?wsdl. Contact your Oracle RightNow Customer Success Manager if you have any questions about the WSDL.

# Setting up Twitter channel Accounts in Oracle RightNow Social Monitor

In order to respond to twitter posts sent from Oracle Social Engagement & Monitoring Cloud Service to Oracle RightNow Cloud Service, you need to add twitter channel accounts to Oracle RightNow Cloud Service. The details to setup the twitter channel account can be found in the Oracle RightNow User guide in the section: Social Experience > Channels > Social channels overview > Adding Twitter channel accounts

You can find the latest Oracle RightNow User guide at the location: https://cx.rightnow.com/app/answers/detail/a\_id/5168

# Conclusion

This concludes the required configuration for you to send posts from Oracle Social Engagement & Monitoring Cloud Service to Oracle RightNow Cloud Service as incidents and for Oracle Social Engagement & Monitoring Cloud Service to get updates on the post’s internal activity once the incident is closed.

|  |  |
| --- | --- |
| O_signature_clr_rgb | A special Oracle logo highlighting Oracle's commitment to developing practices and products that protect the environment. |
| Integration between Oracle RightNow Cloud Service & Social Engagement & Monitoring Cloud Service  September 2013  Oracle Corporation World Headquarters 500 Oracle Parkway Redwood Shores, CA 94065 U.S.A.  Worldwide Inquiries: Phone: +1.650.506.7000 Fax: +1.650.506.7200  oracle.com | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.  Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.  Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0312  HSET_1line_clr |