



An Oracle White Paper  
Sep 2014

## Commerce (ATG)-RightNow Integration

## Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Executive Overview .....	2
Feature Objective.....	3
Integration Overview .....	3
Module Composition .....	4
Prerequisites.....	4
Installation.....	4
Configuration.....	14
Operations.....	18
Initial Profile Load.....	18
Incremental Profile Updates .....	19
Puller Service .....	20
CSC Integration.....	20
REST API.....	22
Customization.....	30
Conclusion .....	34

## Executive Overview

Oracle provides a range of industry leading products to help organizations sell to and service their customers, while providing exceptional customer experience. Oracle views customer experience in terms of a never-ending cycle of assisting customers to research and purchase items or services, and then providing support to those customers. As shown in the following diagram, the cycle of attracting, selling, servicing, and retaining customers is an ongoing process and requires consistent interaction with the customer base.



Two key elements of the Oracle solution are Oracle Commerce and Oracle RightNow. Commerce is an industry leading commerce solution that orchestrates the user experience during the research, selection, and purchase process. In the support and service area, RightNow provides best of breed customer support capabilities.

The integration discussed in this white paper brings together the capabilities of Oracle Commerce and Oracle RightNow to provide a unified solution for businesses to sell and support their customers. The integration provides for synchronization of customer data between Oracle commerce and RightNow to ensure that there is a consistent, 360-degree view of the customer across both sales and support. In addition, the RightNow call center Agent application has been integrated with Oracle commerce's Commerce Service Center call center application, allowing efficient use of the tools by an agent without the need to copy and paste data or coordinate activities between two systems.

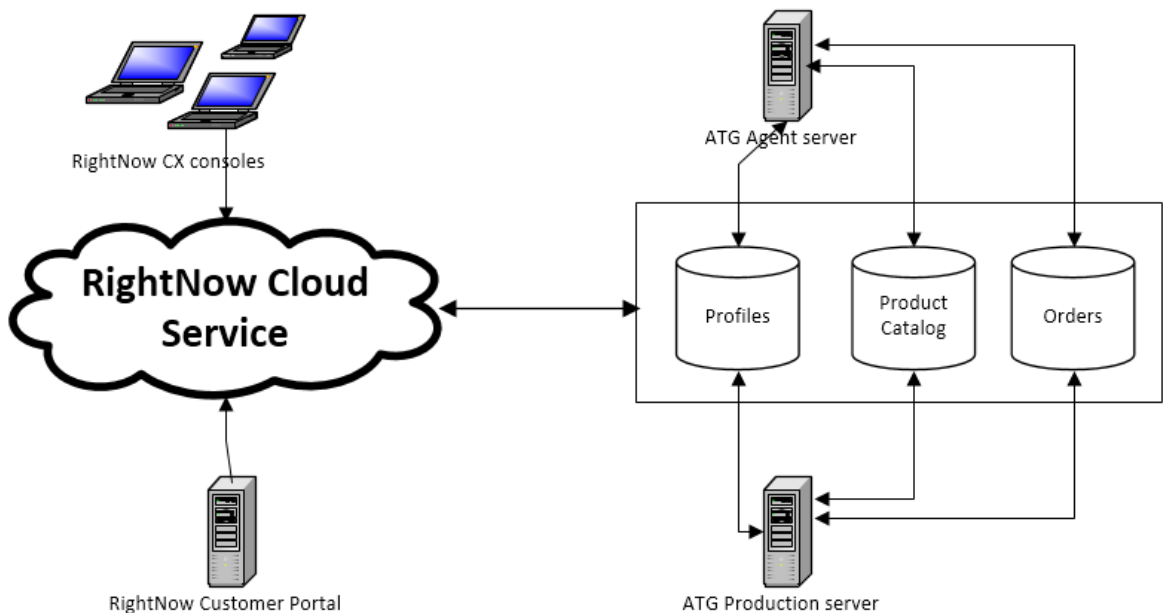
Combined RightNow and Oracle Commerce provide an effective solution for managing customer data end to end, while greatly improving call center efficiency and customer experience for commerce providers.

## Feature Objective

The Integration module is responsible for synchronization of profiles from Oracle Commerce to RightNow and provides the SSO mechanism for the Oracle Commerce CSC application. In addition, RightNow can access order history information, profile details, etc., through the new REST API.

## Integration Overview

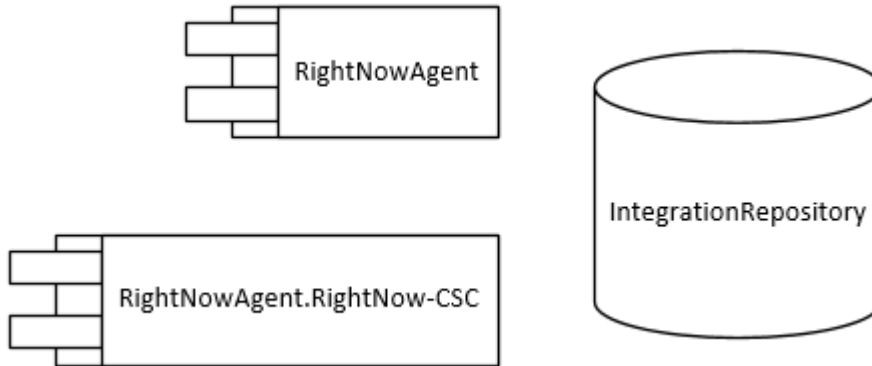
The following figure depicts the high level integration overview. On the Oracle Commerce side, it offers full synchronization of profiles and read-only access to product catalogs and orders.



- **RightNow CX console:** .NET desktop CRM application. With the integration enabled, agents have access to contact order history and can start CSC applications in *window* mode.
- **RightNow Customer Portal:** self-service site, customers can register new Contacts here.
- **RightNow Cloud Services:** all consoles, Customer Portal, and other RightNow apps are shared data in the cloud.

- **Oracle Commerce Agent server:** runs CSC applications.
- **Oracle Commerce Production server:** runs storefront applications (CRS OOTB).

## Module Composition



- **RightNowAgent:** a core module, consisting of basic functionality like IntegrationRepository definition, puller service, and bulk and incremental loaders.
- **RightNowAgent.RightNow-CSC module:** used on the agent server and responsible for REST API and CSC SSO integration.
- **IntegrationRepository:** a new repository which holds rightnowItem items among others. These items link ATG profiles to RightNow contacts.

## Prerequisites

The RightNowAgent module has been tested with ATG 9.4, ATG 10. 2, ATG 11.0 and Oracle Commerce 11.1 and refer the below pre-requisites for installation of Oracle Commerce 11.1

- Java JDK 1.7.0\_11 (or later). Ensure that the JAVA\_HOME and JAVA\_HOME/bin variables are set in your path correctly.
- A supported application server, such as JBoss or WebLogic. Consult the ATG Installation and Configuration Guide for your application server. In our guide we will use a WLS 12 application server.
- WebLogic server version 12 should be installed and running.
- Contact Right Now team and ask to create a separate site, which will be mapped for server with installed ATG builds. Example of the site link:[https://<some\\_url>/cgi-bin/myinterface.cfg/php/admin/launch.php](https://<some_url>/cgi-bin/myinterface.cfg/php/admin/launch.php)
- Name of the server where ATG builds are installed (eg. hostname.company.com), CSC URL (e.g. https:// hostname.company.com:7003/), IP of the server where ATG builds are installed (eg. 10.152.97.23).

- After the RN team create a site and provide a link, install this site locally: Open the link using IE browser. Click Install Oracle Right Now CX Cloud Service. Wait until the installation is complete. (It could take some time. About 10 min for Downloading (~4,27 MB)).  
Oracle RN will be started after install. Login to the application using user name and password, provided by RN team.

## Installation

The RightNowAgent integration module is distributed as a JAR package and can be downloaded from the TBD. Please refer below for the complete installation.

### Install Oracle Commerce products:

#### 1) Install ATG 11.1

- Run the respective REL (OCPlatform\_FlexInstr11.1\_249REL.bin)build for Oracle Commerce 11.1
- Choose Locale
- Click “Next”
- Accept the License Agreement and click “Next”
- Choose the folder to install your ATG Platform (e.g. “/work/service/ATG / Oracle Commerce 11.1”) and click “Next”
- Click “Select All” then click “Next”
- Choose “Oracle WebLogic” and click “Next”
- Choose the folders where your WLS and JDK are installed and click “Next”
- Click “Install”

#### 2) Install ENDECA (As from ATG 11.0 onwards Endeca is being replacing ATG Search)

Please refer the below steps for its installation

##### a) MDEX Installation

Execute the installer file OCmdex6.5.1-Linux64\_829811.sh by giving the below command

```
./ OCmdex6.5.1-Linux64_829811.sh --target <location of Endeca install>
```

It will uncompress the installer file & later run the source command to set your environment variables followed by verifying the path in set environments i.e.

```
ENDECA_MDEX_ROOT
```

##### b) Platform Service Installation

Execute the installer file OCplatformservices11.1.0-Linux64\_842407REL.sh by giving the below command

```
./ OCplatformservices11.1.0-Linux64_842407REL.sh --target <location of Endeca install>
```

It will Uncompress Endeca Platform Services and configure the ports by giving the default ones.

Please enter the port number of the Endeca HTTP service. The typical default is 8888.

: 8888

Please enter the shutdown port number of the Endeca HTTP service. The typical default is 8090.

: 8090

Please enter the port number of the Endeca Control System JCD, or nothing if you do not plan to use a Control System environment. The typical default is 8088.

: 8088

Would you like this installation configured to run the Endeca Application Controller (EAC)? (Y/N)? (Please note: this will also install the EAC Agent)

: Y

Please enter the root of your Endeca MDEX Engine installation. The root directory path typically includes the version number.

: Give the default path

Would you like to install the reference implementations? (Y/N)?

: Y

Run the following command to set your environment variables:

csh or tcsh: source

```
/work/ProductSetup/install/endeca/PlatformServices/workspace/setup/installer_csh.  
ini
```

### c) Install Tools and Frameworks

Go to installed Endeca directory & from there we need to install Tools and Frameworks by uncompressing the OCxmgr11.1.0-Linux64\_89335900182REL.zip file and also location of this file should be given in the below command

```
jar - xvf <location of zip file>
```



Unzipping of the file will take place & Open the .cshrc file and please make sure to verify the path of below environment parameters

```
ENDECA_TOOLS_ROOT=/work/platform/endeca/ToolsAndFrameworks/11.1.0
```

```
ENDECA_TOOLS_CONF=/work/platform/endeca/ToolsAndFrameworks/11.1.0/  
server/workspace
```

Check whether user has executable permissions on all files in ToolsAndFrameworks directory and if not please give the below command

```
chmod -R 777 ToolsAndFrameworks
```

Go to Endeca directory and refer the below commands to start Tools and Frameworks server

```
cd ToolsAndFrameworks/3.1.1/server/bin/
```

```
./startup.sh
```

#### d) **Install CAS**

Go to your previous location where you have stored your executable file and for CAS installation please give the below command

```
./cas-3.1.1-x86_64pc-linux.sh --target <location of Endeca install>
```

It will initiate the Uncompressing Content Acquisition System & configure the ports by giving the default ones.

Please enter the port for the CAS service. The typical default is 8500.

```
: 8500
```

Please enter the shutdown port for the CAS service. The typical default is 8506.

```
: 8506
```

Please enter the fully qualified name, including domain information, of the CAS server.

Please give your hostname here

```
:hostname.company.com
```

Installing casconsole.xml into

```
/work/ProductSetup/install/endeca/ToolsAndFrameworks/11.1.0/server/workspace  
/conf/Standalone/localhost.
```

CAS installation is complete.

### e) Starting CAS

Once CAS installation is complete, you will need to re-start the ToolsAndFrameworks and Platform services.

Shutdown the Endeca Tools Service by

```
cd /work/ProductSetup/install/endeca/ToolsAndFrameworks/11.1.0/server/bin/  
./shutdown.sh
```

Shutdown the Platform Services by

```
cd $ENDECA_ROOT/tools/server/bin  
./shutdown.sh
```

Start Platform Services by

```
./startup.sh
```

Start the Endeca Tools Service again

```
cd /work/ProductSetup/install/endeca/ToolsAndFrameworks/11.1.0/server/bin/  
./startup.sh
```

Start CAS Service by giving the below command

```
cd into the CAS_INST_DIR/version/bin directory,  
e.g.: work/platform/endeca/CAS/11.1.0/bin  
invoke cas-service.sh script as follows
```

```
./cas-service.sh > casLog.txt &
```

Here by Endeca installation is completed and in order to verify whether all its services are running properly, please refer the below command

```
ps -eaf | grep Endeca
```

It should show the processing ids of MDEX, Platform Services, Tools and Frameworks and CAS which are running without any issues.

### 3) Install CRS

- Run OReferenceStore11.1\_137REL.bin
- Choose Locale
- Click “Next”
- Accept the License Agreement and click “Next”

- Choose the folder where you have installed the ATG Platform (e.g. “/work/service/ATG/ATG11.1”) and click “Next”
- Click “Install”

#### 4) Install Service

- Run OCServiceCenter11.1\_99REL.bin
- Choose Locale
- Click “Next”
- Accept the License Agreement and click “Next”
- Select all for “Product Features” and click “Next”
- Choose the folder where you have installed the ATG Platform (e.g. “/work/service/ATG/ATG11.1”) and click “Next”
- Leave Host as DEFAULT: localhost
- Click “Install”

#### 5) Right now JAR

Unjar latest OCRightNowAgent11.1\_8RCN.jar build from Road to DYNAMO\_ROOT, for example, run next command:

```
jar -xvf OCRightNowAgent11.1_8RCN.jar
```

- 6) Check DYNAMO\_HOME variable is correct, for example: DYNAMO\_HOME=/work/service/ATG/ATG11.1/home

#### Configure database and server instances:

- 7) Run CIM:
- Navigate to your ATG home folder (e.g. “/work/service/ATG/ATG11.1/home”).
  - Navigate to the “/bin” subfolder and run “./cim.sh”
- 8) Set the Administrator Password (eg. Admin123)
- 9) Product selection:  
The following products should be selected (NOTE: selected items are marked with ‘ \* ’):
- \* RightNow CSC Integration module  
RightNow CSC Integration module
  - \* Oracle ATG Commerce Service Center:  
Agent-facing commerce application
  - \* Oracle ATG Commerce Reference Store:  
Includes the ATG platform, Content Administration, Site Administration, Oracle ATG Web Commerce and Oracle ATG Web Commerce Merchandising. Optional: data warehouse components and Preview
- 10) Do not Choose any AddOns :

- 
- [1] Reporting
  - [2] Staging Server
  - [3] Dedicated Lock Servers
  - [4] Abandoned Order Services
  - [5] Preview Server
  - \* [D] Done
- 11) Do not Include Demo Application:
- [1] Quincy Funds Demo
  - \* [D] Done
- 12) Choose datasource
- \*[1] Non-Switching Datasource
  - [2] Switching Datasource
- 13) Do not include Search QueryConsole
- [1] Include Search QueryConsole on Management
  - [2] Include Search QueryConsole on Production
  - [3] Include Search QueryConsole on Agent
  - \*[D] Done
- 14) Choose Commerce Service Center Addons:
- [1] ClickToConnect
  - \*[2] Multi Site
  - [3] Horizontal Partitioning (Sharding) of Search Index for Orders
- \* [D] Done
- 15) Choose ATG Commerce Reference Store AddOns :
- \* [1] Storefront Demo Application
  - \* [2] International : Enables multi-country and multi-language functionality
  - \* [3] Fulfillment
  - [4] Oracle Recommendations On Demand Integration
  - [5] RightNow KnowledgeBase
  - [6] ClickToCall Integration
  - [7] Mobile Reference Store
  - \*[D] Done
- 16) Do not Choose Storefront AddOns:
- [1] Inspect Application
  - \* [D] Done
- 17) Choose index by product
- \*[1] Index by Product
  - [2] Index by SKU
- 18) Choose Storefront Sample Data Configuration:
- \* [1] Full : Populated product catalog, users, orders, promotions, etc.
  - [2] Minimal : Empty product catalog and minimum data required for startup
- 19) Select application server

- Oracle Weblogic Online (Either Production or Developer Mode)
  - Enter path to your WLS folder, eg.  
“/work/service/Oracle/Middleware/Oracle\_Home/wlserver” and then domain path  
“/work/service/Oracle/Middleware/Oracle\_Home/user\_projects/domains/base\_domain”
  - Enter URL to admin server , eg. “t3://localhost:7001”
  - Enter username to admin server: weblogic
  - Enter password to admin server: weblogic1
  - Reenter password and Perform Validation. Validation should be successful, WLS is up and running.
- 20) In this guide we will use Oracle DB. We should create 3 schemas: publishing, production, agent. Schemas should be created before database configuration process starts.

**Database configuration.**

- Publishing  
Connection details **(for example :)**  
Enter User Name > phanna01  
Enter Password > foophanna01  
Enter Host Name > busgt3202.company.com  
Enter Port Number [[1521]] >  
Enter Database Name > utf8112  
Enter Database URL [[jdbc:oracle:thin:@busgt3202.company.com  
: 1521:utf8112]] >  
Enter Driver Path > /work/service/db/ojdbc6.jar  
Enter JNDI Name [[ATGPublishingDS]]  
Perform next steps:  
Test connection  
Create schema  
Import initial data
  
- Production  
Connection details **(for example :)**  
Enter User Name > phanna02  
Enter Password > \*\*\*\*\*  
Enter Host Name > busgt3202.company.com  
Enter Port Number [[1521]] >  
Enter Database Name > utf8112  
Enter Database URL [[jdbc:oracle:thin:@busgt3202.company.com:1521:utf8112]]  
>

```

Enter Driver Path > /work/service/db/ojdbc6.jar
Enter JNDI Name [[ATGPublishingDS]]
Test connection
Create schema
Import initial data
• Agent datasource
  Connection details (for example:)
  Enter User Name > phanna04
  Enter Password > *****
  Enter Host Name > busgt3202.company.com
  Enter Port Number [[1521]] >
  Enter Database Name > utf8112
  Enter Database URL [[jdbc:oracle:thin:@busgt3202.company.com:1521:utf8112]]
>
Enter Driver Path > /work/service/db/ojdbc6.jar
Enter JNDI Name [[ATGPublishingDS]]
Test connection
Create schema
Import initial data

```

### Service Instance Configuration

21) Configure publishing server:

```

* Publishing Server General Configuration – REQUIRED (for example:)
  Enter Production Lock Server Hostname > busgt0515.company.com
  Enter Production Lock Server Port [[9012]] >
  Enter RightNow endpoint: > https://rnatg1.qa.lan/cgi-
bin/rnatg1.cfg/services/soap
  Enter RightNow Customer Id: > service
  Enter RightNow Customer Password: > *****
  Re-enter Password > *****

```

Instance Management - REQUIRED

```

Add Server Instance
Publishing with a Server Lock Manager: Minimum 1 Required
Enter Server Instance Name: atg_publishing_lockserver
Choose Custom Port Binding ports-01:
Enter HTTP Port [[7103]] >
Enter HTTPS Port [[7104]] >
Enter Site HTTP Port [[7103]] >
Enter RMI Port [[8861]] >
Enter DRP Port [[8851]] >
Enter File Deployment Port [[8811]] >

```

Enter Lock Server Port [[9010]] >

22) Configure production server:

\* Production Server General Configuration – REQUIRED (for example:)

Enter RightNow endpoint: [[https://rnatg1.qa.lan/cgi-bin/rnatg1.cfg/services/soap]] >

Enter RightNow Customer Id: [[service]] >

Enter RightNow Customer Password: [[\*\*\*\*\*]] >

SMTP Email Host Name > internal-mail-router.company.com

SMTP Email Port Number [[25]] >

Enter Search Engine directory: > /work/service/ATG/ATG11.1/Search11.1/SearchEngine

Enter directory where you want search engine indexes deployed: >

/work/service/ATG/ATG11.1/Search11.1/SearchEngine

\* Instance Management - REQUIRED

Add Server Instance

Production with a Server Lock Manager : Minimum 1 Required

Enter Server Instance Name : atg\_production\_lockserver

Use Custom Port Binding. Ports 02

Enter HTTP Port [[7203]] >

Enter HTTPS Port [[7204]] >

Enter Site HTTP Port [[7203]] >

Enter RMI Port [[8862]] >

Enter DRP Port [[8852]] >

Enter File Deployment Port [[8812]] >

Enter Lock Server Port [[9012]] >

23. Configure agent server:

\* Agent Server General Configuration – REQUIRED (for example:)

Enter RightNow endpoint: [[https://rnatg1.qa.lan/cgi-bin/rnatg1.cfg/services/soap]] >

Enter RightNow Customer Id: [[service]] >

Enter RightNow Customer Password: [[\*\*\*\*\*]] >

Enter RightNow-CSC request hash digest: [[SHA-256]] >

Enter RightNow-CSC request hash salt: [[Enter HTTP Port [[7203]] >]] > RightNowSalt

Enter Production Server Hostname > busgt0515.company.com

Enter Production Server Site HTTP Port [[7003]] > 7203

Enter Production Lock Server Address [[busgt0515.company.com]] >

Enter Production Server Client Lock Manager Port: [[9012]] >

SMTP Email Host Name [[internal-mail-router.company.com]] >

SMTP Email Port Number [[25]] >

Enter Search Engine directory: [[/work/service/ATG/ATG11.1/Search11.1/SearchEngine]]

>

Enter directory where you want search engine indexes deployed:

[[/work/service/ATG/ATG11.1/Search11.1/SearchEngine]] >

- Enter the default catalog ID: [[default\_catalog\_id]] > masterCatalog  
 Enter the default site ID > storeSiteUS  
 Enter Default MDEX Host Name [[localhost]] >  
 Enter Default MDEX Port Number [[15000]] >
- \* Instance Management - REQUIRED
- Add Server Instance
    - Service Agent Server
      - Enter Server Instance Name : [[atg\_svcagent]] >
      - Choose Default Port Binding
        - Enter HTTP Port [[7003]] >
        - Enter HTTPS Port [[7004]] >
        - Enter RMI Port [[8860]] >
        - Enter DRP Port [[8850]] >
        - Enter File Deployment Port [[8810]] >
        - Enter File Synchronization Deploy Server Port [[8815]] >
        - Enter Agent Lock Server Port > 9011
- 24) Application Assembly & Deployment
- a. Accept default values on all steps.
- 25) Quit CIM
- 26) Verify server instances created:**
- a. Navigate to your ATG home folder (e.g. “/work/service/ATG/ATG11.1/home”)
  - b. Navigate to the “/servers” subfolder. There are 3 servers we’ve just created:
    - i. atg\_production\_lockserver
    - ii. atg\_publishing\_lockserver
    - iii. atg\_svcagent\_lockserver

## Configuration

**Note:** Install the trusted SSL certificate, and enable SSL in the application server. For the procedure, refer the below steps:

### 1) Configure SSL for Weblogic

1. Run the below Java keytool -genkey command to generate required files:

Note: Below busgf1410 is the host name of the installed machine, referred as an example



In the below location of security:

```
/scratch/service/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain/security
```

```
keytool -genkey -v -alias busgf1410_key -keyalg RSA -keysize 1024 -dname "CN=busgf1410.company.com, OU=CSR Web, O=CSR\, Inc., C=US" -keypass atgpass -validity 365 -keystore busgf1410_keystore.jks -storepass atgpass
```

2. Copy the newly generated file to BEA\_HOME\user\_projects\domains\<>your domain> directory
3. Run the below Java keytool command -certreq utility:

In base domain directory of below path

```
/scratch/service/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain
```

```
keytool -certreq -v -alias busgf1410_key -file busgf1410_keystore.pem -keypass atgpass -storepass atgpass -keystore busgf1410_keystore.jks
```

4. Restart all servers, Start the BEA admin server and all of the managed servers
5. Open a web browser and navigate to BEA console. Navigate to Environment → Servers → your managed server → Configuration → Key stores.  
Svc agent is the Managed Server here
6. Lock the console.

In the Keystores page, specify the following properties:

Keystores: Custom Identity and Java Standard Trust

Custom Identity Keystore: busgt0515\_keystore.jks

Custom Identity Keystore Type: JKS

Custom Identity Keystore Passphrase: atgpass

Java Standard Trust Keystore Passphrase: changeit

7. Navigate to Environment → Servers → your managed server → Configuration → SSL.

In the SSL page, specify the following properties:

Identity and Trust Locations: Keystores

Private Key Alias: busgt0515\_key

Private Key Passphrase: atgpass

8. Save the changes.

9. Navigate to Environment → Servers → your managed server → Configuration → General. Verify that the check box next to SSL Listen Port Enabled is checked and define a SSL Listen Port other than its http port. Give SSL port it as 7004 if http is 7003 for Svc agent.
10. Activate the changes and restart the managed servers.

## 2) Add certificate to the Java keystore:

- a) Open the Right now site URL in the below format which will be received from Right now team.  
[https://<some\\_url>/cgi-bin/myinterface.cfg/services/soap?wsdl](https://<some_url>/cgi-bin/myinterface.cfg/services/soap?wsdl)
- b) Click on the Lock symbol available in Address bar & followed by View Certificates
- c) Click on Install Certificate followed by finish button.
- d) Now go to Internet Options → Content → Certificates → Trusted Root Certification Authorities → RNT Internal Certificate & then export it.
- e) Save the certificate file to your hard disk , copy to any location of ur linux box and give its path accordingly in the below command.
- f) Navigate to the JDK folder (e.g. “c:\Java\jdk1.7.0\_40\” & Navigate to the “\jre\lib\security\” subfolder
- g) Run the following command  
keytool -import -file <path\_to\_certificate\_file> -keystore cacerts -storepass changeit  
keytool -import -file /scratch/ptoleti/Downloads/ RTnow.cer -keystore cacerts -storepass changeit
- h) For deleting the existing certificate, refer the below command  
keytool -v -delete -alias mykey -keystore cacerts -storepass changeit

## 3) Use next commands to start servers:

```
Goto /work/service/bea/user_projects/domains/base_domain/bin
nohup ./startManagedWebLogic.sh atg_production_lockserver http://localhost:7001/
1>production.log
nohup ./startManagedWebLogic.sh atg_svcagent_lockserver http://localhost:7001/ 1>svcagent.log
nohup ./startManagedWebLogic.sh atg_publishing_lockserver http://localhost:7001/
1>publishing.log
```

## 4) After environment creation you should run initial bulkLoad in the following components:

<http://<server>:<port>/dyn/admin/nucleus/atg/commerce/search/OrderOutputConfig/>

<http://<server>:<port>/dyn/admin/nucleus/atg/userprofiling/search/ProfileOutputConfig/>

Here server is the hostname of machine and port belongs to agent server which is referred in below example

<http://busgf1410.comapny.com:7203/dyn/admin/nucleus/atg/integration/rightnow/userprofiling/ProfileOutputConfig/>

& hit Invoke Bulk method accordingly and later verify that all the contacts are now moved to Right now console from Service Center.

## 5) Errors & Remedies

**Error no 1** /atg/integration/rightnow/puller/ContactSchedulablePuller ---

javax.xml.ws.WebServiceException: javax.net.ssl.SSLKeyException: Hostname verification failed: HostnameVerifier=weblogic.security.utils.SSLWLSHostnameVerifier, hostname=atgqa2.qa.lan.

**Remedy :** Update the setDomainEnv.sh in bin folder with the below content :

```
/scratch/service/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain/bin
JAVA_OPTIONS="{JAVA_OPTIONS}" -
Dweblogic.security.SSL.ignoreHostnameVerification=true"
export JAVA_OPTIONS
```

**Error no 2** --- com.sun.xml.ws.client.ClientTransportException: The server sent HTTP status code 200: OK

**Remedy:** update the WebServiceClient.properties available in:

```
/scratch/service/ATG/ATG11.1/home/servers/atg_svcagent_lockserver/localconfig/atg/integration/rightnow
```

With the proper end point URL of Rightnow instance which is endpoint=https://atgqa2.qa.lan/cgi-bin/atgqa2.cfg/services/soap

**Error no 3:** SEVERE: (Error) null atg.repository.search.indexing.IndexingException: com.rightnow.ws.wsdl.RequestErrorFault: Cannot save/create: Contact: (null)

While performing create on Contact at

```
atg.integration.rightnow.BulkSubmitterSession.submitQueue(BulkSubmitterSession.java:81)
```

**Remedy:** While doing the bulk load, always delete the existing contacts in Right now before doing the bulk load

**Error no 4**

```
**** Error Thu Nov 28 07:02:27 EST 2013 1385640147766
```

```
/atg/integration/rightnow/puller/ContactSchedulablePuller ---
```

com.sun.xml.ws.client.ClientTransportException: The server sent HTTP status code 404: Not

**Remedy:** update the WebServiceClient.properties available in :

```
/scratch/service/ATG/ATG11.1/home/servers/atg_svcagent_lockserver/localconfig/atg/integration/rightnow
```

With the proper end point URL of Rightnow instance which is endpoint=https://atgqa2.qa.lan/cgi-bin/atgqa2.cfg/services/soap

**Error no 5**

**\*\*\*** Caused by: atg.repository.search.indexing.IndexingException:

atg.repository.search.indexing.IndexingException: com.rightnow.ws.wsdl.RequestErrorFault: Invalid ID: Cannot resolve ID; Contact.Emails[0].AddressType.ID While performing create on Contact

**Remedy:** Go to /atg/integration/rightnow/IntegrationRepository and check if values are present in namedIdValue Item-Descriptor.

If Values are not present

- 1) Go to /atg/integration/rightNow/NamedIdSyncService & Invoke executeScheduleMetod method
- 2) Check if the values got added in the IntegrationRepository & If Yes, restart ATG CSC instance

- 3) In case RightNow contains any contacts, delete them
- 4) Perform bulkLoad from /atg/integration/rightnow/userprofiling/ProfileOutputConfig

## Operations

### Initial Profile Load

IT admin manually starts the bulk profile load from the Dynamo Administration UI. To do this, the personnel responsible for deploying and configuring ATG solutions should open the <http://<host>:<port>/dyn/admin/nucleus/atg/integration/rightnow/userprofiling/ProfileOutputConfig/> component and invoke the **bulkLoad** method.

The config/atg/integration/rightnow/userprofiling/profile-output-config.xml component specifies OOTB mapping between ATG profiles and RightNow contacts:

```
<item item-descriptor-name="user" is-document="true">
  <!-- Use title as a name of RightNow object -->
  <title property-name="Contact" is-non-repository-property="true"/>
  <meta-properties>
    <property name="login" type="string" store-as-meta-index="true" output-name="login"/>

    <!-- Email object -->
    <property name="email" type="string" store-as-meta-index="true" output-name="emails.emailList.email"
property-accessor="/atg/integration/rightnow/EmailPropertyAccessor"/>

    <property name="firstName" type="string" store-as-meta-index="true" output-name="name.first"/>
    <property name="lastName" type="string" store-as-meta-index="true" output-name="name.last" />

    <!-- Mobile phone object -->
    <property name="daytimeTelephoneNumber" type="string" store-as-meta-index="true" output-
name="phones.phoneList.phone" property-
accessor="/atg/integration/rightnow/MobilePhonePropertyAccessor" />
  </meta-properties>

  <!-- Use home address as primary contact address -->
  <item property-name="homeAddress" is-multi="false">
    <meta-properties>
      <property name="city" type="string" store-as-meta-index="true" output-name="address.city" />
      <property name="country" type="string" store-as-meta-index="true" output-name="address.country" />
      <property name="postalCode" type="string" store-as-meta-index="true" output-
name="address.postalCode" />
    </meta-properties>
  </item>
</item>
```

```

    <property name="state" type="string" store-as-meta-index="true" output-name="address.stateOrProvince"
/>

    <!-- Combine address1 and address2. Listen to changes in both properties -->
    <property name="address1" type="string" store-as-meta-index="true" output-name="address.street"
property-accessor="/atg/integration/rightnow/AddressPropertyAccessor"/>
    <property name="address2" type="string" store-as-meta-index="true" output-name="address.street"
property-accessor="/atg/integration/rightnow/AddressPropertyAccessor"/>

    <!-- Home phone object -->
    <property name="phoneNumber" type="string" store-as-meta-index="true" output-
name="phones.phoneList.phone" property-
accessor="/atg/integration/rightnow/HomePhonePropertyAccessor" />
    </meta-properties>
</item>
</item>

```

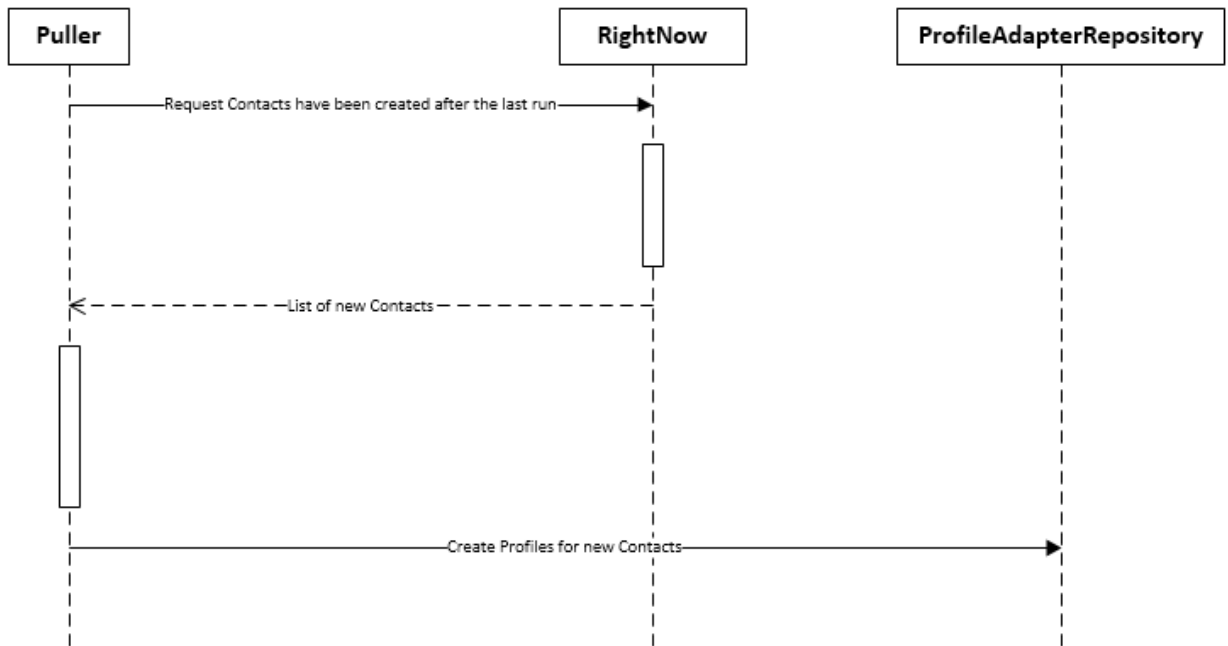
We are using title as a RightNow object name. The user repository item descriptor is mapped to the Contact RightNow object. Every profile property listed previously is mapped to the contact field. For instance, profile.firstName is mapped to Contact.name.first. If additional transformation is required, property-accessor is used.

### Incremental Profile Updates

Every two seconds (**Note:** this value is configurable), the atg/integration/rightnow/repository/IndexingPeriodicService component runs atg/integration/rightnow/repository/IncrementalLoader and performs near real-time live indexing of profile changes. IncrementalLoader also performs near real-time pushing of profile changes. Every change made on CSC or storefront application is uploaded into the RightNow site.

### Puller Service

The puller allows viewing of the contact created/updated in RightNow through CX console, Customer Portal, 3rd party system, etc. in ATG as a profile. It runs every two seconds and requests new contacts that have been created in RightNow site since the last synchronization. Once the puller gets the list of new contacts, it processes every contact and creates appropriate profiles on ATG side.



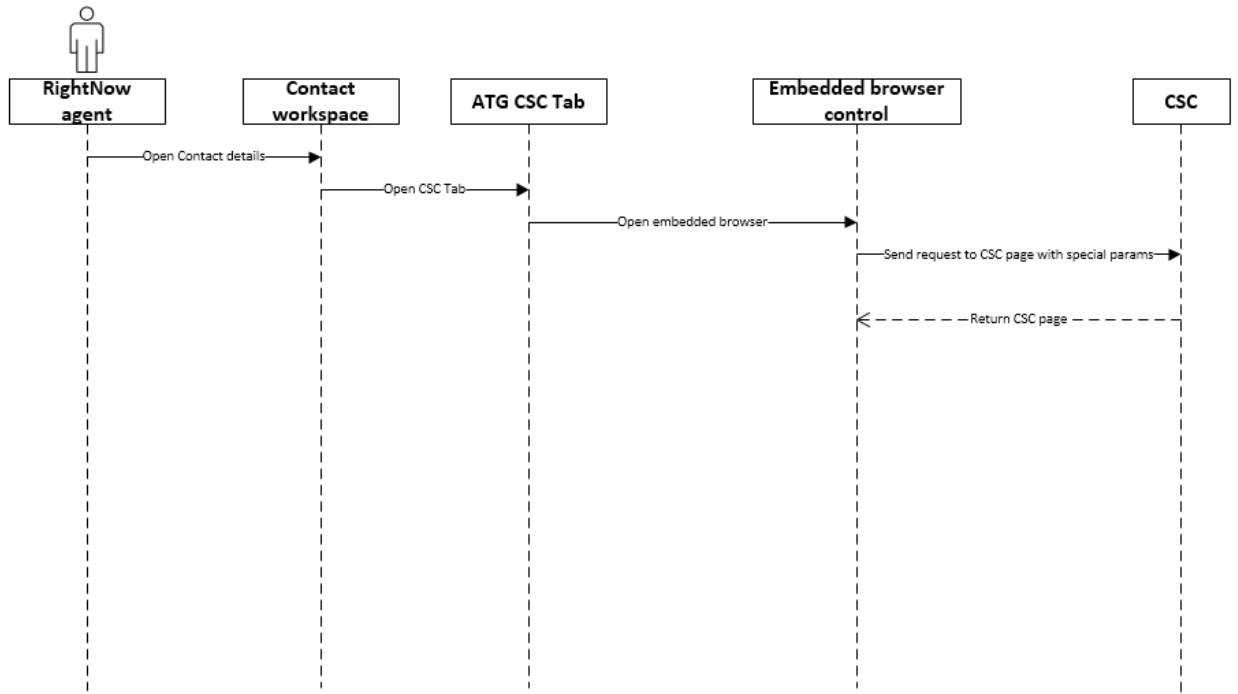
We are using a special condition to retrieve contacts from RightNow called the **ROQL statement**, which is configurable in `atg/integration/rightnow/puller/ContactSchedulablePuller`:

```
pullROQL=SELECT Contact FROM Contact WHERE UpdatedTime > :state
```

### CSC Integration

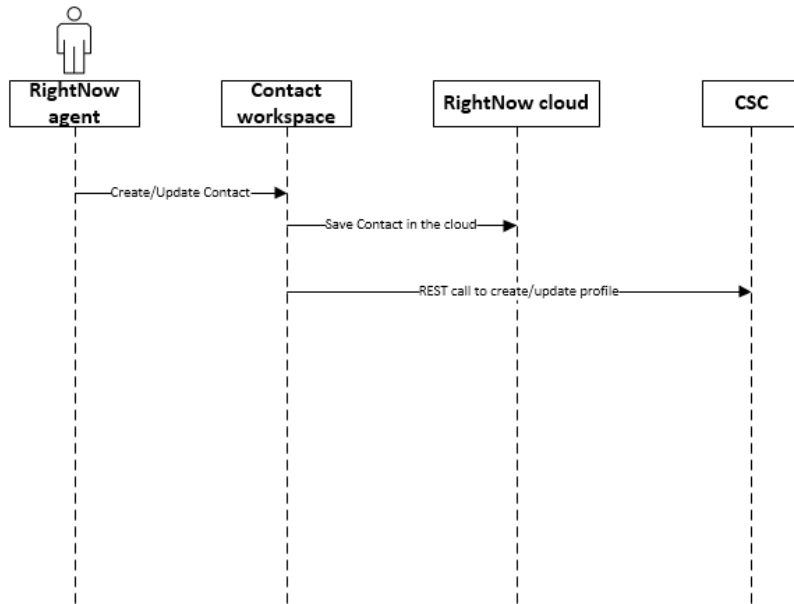
The RightNow agent opens Contact in the contact workspace in RightNow CX console. There is an **ATG CSC Contact Details** tab which has embedded browser controls. When the agent clicks this tab, the console initiates an HTTP request to the CSC application and opens the CSC page inside of the embedded browser control.

CSC processes the HTTP request from the console and performs an auto login for the user provided in the HTTP header. For security reasons, the header also contains the hash, which is analyzed on ATG agent server to make sure the request is valid.



## REST API

An example of a typical REST call to create or update a profile is shown in the following diagram. The RightNow agent opens the contact workspace, and creates a new contact or modifies an existing one. This action is then saved. The operation initiates a REST call to the ATG server and a new profile is created or an existing profile is updated.



RightNowAgent provides REST API for RightNow to interact with ATG internal and external profiles, orders, perform search, etc. Full list of REST services:

- Login: `/rest/bean/atg/userprofiling/InternalProfileFormHandler/login`
- Logout: `/rest/bean/atg/userprofiling/ProfileServices/logoutUser`
- Create internal profile: `/rest/bean/atg/userprofiling/ProfileServices/createInternalUser`
- Create external profile: `/rest/bean/atg/userprofiling/ProfileServices/createContact`
- Update external profile: `/rest/bean/atg/userprofiling/ProfileServices/updateContact`
- Delete internal profile: `/rest/bean/atg/userprofiling/ProfileServices/deleteInternalUser`
- Get list of order statuses: `/rest/bean/atg/commerce/OrderServices/getOrderStatuses`
- Get list of order originators: `/rest/bean/atg/commerce/OrderServices/getOrderOriginators`
- Get list of commerce item statuses:  
`rest/bean/atg/commerce/OrderServices/getCommerceItemStatuses`
- Get profile by RightNow Contact ID:  
`/rest/bean/atg/userprofiling/ProfileServices/getProfileByRightNowId`



- Get orders by RightNow Contact ID:  
rest/bean/atg/userprofiling/ProfileServices/getOrdersByRightNowId
- Search for profile: /rest/bean/atg/integration/rightnow/order/RestOrderSearchFormHandler/search

Following are examples of REST services usage given with the curl utility: <http://curl.haxx.se/>

#### Login

```
curl -v -c cookies.txt -X POST -H "Content-Type: application/json" -d "{atg-rest-return-form-handler-exceptions:true,value.login:'service',value.password:'Service123'}" https://pt-asery-olx6-2.company.com:8282/rest/bean/atg/userprofiling/InternalProfileFormHandler/login
```

Response successful: {"atgResponse": null}

Response unsuccessful: {

```
"class": "class atg.rest.processor.BeanProcessor$FormHandlerExceptions",
"formExceptions": ["The supplied user name <i>service</i> is incorrect"],
"result": false
}
```

#### Create Contact

```
curl -v -b cookies.txt -X POST -H "Content-Type: application/json" -d "{atg-rest-depth:3,arg1:'ID=righnowId,login=loginName,firstName=fName,lastName=lastName,email=email,address.city=City'}" https://pt-asery-olx6-2.company.com:8282/rest/bean/atg/userprofiling/ProfileServices/createContact
```

Successful response:

```
{
  "ID": "righnowId",
  "address": {
    "city": "City",
    "country": null,
    "phoneNumber": null,
    "postalCode": null,
    "repositoryId": "130035",
    "stateOrProvince": null,
    "street": null
  },
  "daytimeTelephoneNumber": null,
  "email": "email",
  "firstName": "fName",
  "lastName": "lastName",
  "login": "loginName",
  "middleName": null,
  "repositoryId": "120035"
}
```

Unsuccessful response:

```
{"atgResponse": "Error: Contact with given RightNowId: righnowId already exists"}
```

### Update Contact

```
curl -v -b cookies.txt -X POST -H "Content-Type: application/json" -d "{atg-rest-depth:3,arg1:'ID=righrnowId,login=loginName,firstName=fName,lastName=lastName,email=email,address.city=City'}" https://pt-asery-olx6-2.company.com:8282/rest/bean/atg/userprofiling/ProfileServices/updateContact
```

### Successful response:

```
"ID": "righrnowId",
"address": {
  "city": "City",
  "country": null,
  "phoneNumber": null,
  "postalCode": null,
  "repositoryId": "130035",
  "stateOrProvince": null,
  "street": null
},
"daytimeTelephoneNumber": null,
"email": "email",
"firstName": "fName",
"lastName": "lastName",
"login": "loginName",
"middleName": null,
"repositoryId": "120035"
}
```

### Unsuccessful response:

Similar response to Create Contact call.

### Get Orders by RightNow ID

```
curl -v -b cookies.txt -X POST -H "Content-Type: application/json" -d "{atg-rest-depth:3,arg1:'righnowId'}" https://pt-asery-olx6-2.company.com:8282/rest/bean/atg/userprofiling/ProfileServices/getOrdersByRightNowId
```

### Response:

```
{"atgResponse": [
  {
    "contactFirstName": "Stuart",
    "contactId": "StuartRightnow",
    "contactLastName": "Schmidt",
    "itemSummary": "Striped Tie",
    "numberOfItems": 0,
    "numberOfItemsReturned": 1,
    "orderId": "xco320011",
    "orderItems": [{
      "description": "Striped Tie",
      "discountedTotalPrice": 39.99,
      "price": 39.99,
      "quantity": 0,
      "repositoryId": "xci2100006",
      "status": 0,
      "totalPrice": 39.99
    }],
    "orderTotal": 46.16,
  }
]}
```

```
"orderTotalCurrencyCode": "USD",
"originator": 2,
"repositoryId": "xco320011",
"status": 5,
"submittedDate": 1340879761
},
{
  "contactFirstName": "Stuart",
  "contactId": "StuartRightnow",
  "contactLastName": "Schmidt",
  "itemSummary": "Dotted Repp Tie; Varsity Trainer",
  "numberOfItems": 2,
  "numberOfItemsReturned": 1,
  "orderId": "xco30012",
  "orderItems": [
    {
      "description": "Dotted Repp Tie",
      "discountedTotalPrice": 38.5,
      "price": 38.5,
      "quantity": 0,
      "repositoryId": "xci1000001",
      "status": 0,
      "totalPrice": 38.5
    },
    {
      "description": "Varsity Trainer",
      "discountedTotalPrice": 80,
      "price": 40,
      "quantity": 2,
      "repositoryId": "xci1000002",
      "status": 0,
      "totalPrice": 80
    }
  ],
  "orderTotal": 118,
  "orderTotalCurrencyCode": "USD",
  "originator": 0,
  "repositoryId": "xco30012",
  "status": 5,
  "submittedDate": 1340624231
},
{
  "contactFirstName": "Stuart",
  "contactId": "StuartRightnow",
  "contactLastName": "Schmidt",
  "itemSummary": "Hepplewhite Side Table; Distressed Jeans; Distressed Brown Belt; Cashmere Scarf",
  "numberOfItems": 0,
  "numberOfItemsReturned": 5,
  "orderId": "xco20040",
  "orderItems": [
    {
      "description": "Hepplewhite Side Table",
      "discountedTotalPrice": 119.2,
      "price": 149,
      "quantity": 0,
      "repositoryId": "xci1000070",
      "status": 0,
      "totalPrice": 149
    }
  ],
}
```

```

    {
      "description": "Distressed Jeans",
      "discountedTotalPrice": 178,
      "price": 89,
      "quantity": 0,
      "repositoryId": "xci1000071",
      "status": 0,
      "totalPrice": 178
    },
    {
      "description": "Distressed Brown Belt",
      "discountedTotalPrice": 29.5,
      "price": 29.5,
      "quantity": 0,
      "repositoryId": "xci1000072",
      "status": 0,
      "totalPrice": 29.5
    },
    {
      "description": "Cashmere Scarf",
      "discountedTotalPrice": 0,
      "price": 10,
      "quantity": 0,
      "repositoryId": "xci1000073",
      "status": 0,
      "totalPrice": 10
    }
  ],
  "orderTotal": 332.7,
  "orderTotalCurrencyCode": "USD",
  "originator": 0,
  "repositoryId": "xco20040",
  "status": 5,
  "submittedDate": 1340629855
},
{
  "contactFirstName": "Stuart",
  "contactId": "StuartRightnow",
  "contactLastName": "Schmidt",
  "itemSummary": "Braided Leather Belt; Paris Clock; Vermont Table Lamp",
  "numberOfItems": 1,
  "numberOfItemsReturned": 2,
  "orderId": "xco30106",
  "orderItems": [
    {
      "description": "Braided Leather Belt",
      "discountedTotalPrice": 45,
      "price": 45,
      "quantity": 0,
      "repositoryId": "xci1000011",
      "status": 0,
      "totalPrice": 45
    },
    {
      "description": "Paris Clock",
      "discountedTotalPrice": 41.65,
      "price": 49,
      "quantity": 0,
      "repositoryId": "xci1000012",
    }
  ]
}

```

```
"status": 0,
"totalPrice": 49
},
{
  "description": "Vermont Table Lamp",
  "discountedTotalPrice": 67.15,
  "price": 79,
  "quantity": 1,
  "repositoryId": "xci1000013",
  "status": 0,
  "totalPrice": 79
}
],
"orderTotal": 143.8,
"orderTotalCurrencyCode": "USD",
"originator": 0,
"repositoryId": "xco30106",
"status": 5,
"submittedDate": 1340624948
},
{
  "contactFirstName": "Stuart",
  "contactId": "StuartRightnow",
  "contactLastName": "Schmidt",
  "itemSummary": "Jersey Wood Table; Hubbard Chair; Moroccan Wood Block Candle",
  "numberOfItems": 8,
  "numberOfItemsReturned": 0,
  "orderId": "xco30076",
  "orderItems": [
    {
      "description": "Jersey Wood Table",
      "discountedTotalPrice": 223.2,
      "price": 279,
      "quantity": 1,
      "repositoryId": "xci1000003",
      "status": 0,
      "totalPrice": 279
    },
    {
      "description": "Hubbard Chair",
      "discountedTotalPrice": 715.2,
      "price": 149,
      "quantity": 6,
      "repositoryId": "xci1000004",
      "status": 0,
      "totalPrice": 894
    },
    {
      "description": "Moroccan Wood Block Candle",
      "discountedTotalPrice": 6.8,
      "price": 8,
      "quantity": 1,
      "repositoryId": "xci1000005",
      "status": 0,
      "totalPrice": 8
    }
  ],
  "orderTotal": 945.2,
  "orderTotalCurrencyCode": "USD",
```

```

"originator": 0,
"repositoryId": "xco30076",
"status": 5,
"submittedDate": 1340624372
},
{
"contactFirstName": "Stuart",
"contactId": "StuartRightnow",
"contactLastName": "Schmidt",
"itemSummary": "Executive Leather Chair with Footrest; Cushioned Ottoman; Swivel Desk Chair; Cashmere
Scarf; Gumdrop Lamp",
"numberOfItems": 5,
"numberOfItemsReturned": 0,
"orderId": "xco30098",
"orderItems": [
{
"description": "Executive Leather Chair with Footrest",
"discountedTotalPrice": 207.2,
"price": 259,
"quantity": 1,
"repositoryId": "xci1000006",
"status": 0,
"totalPrice": 259
},
{
"description": "Cushioned Ottoman",
"discountedTotalPrice": 119,
"price": 119,
"quantity": 1,
"repositoryId": "xci1000007",
"status": 0,
"totalPrice": 119
},
{
"description": "Swivel Desk Chair",
"discountedTotalPrice": 219,
"price": 219,
"quantity": 1,
"repositoryId": "xci1000008",
"status": 0,
"totalPrice": 219
},
{
"description": "Cashmere Scarf",
"discountedTotalPrice": 0,
"price": 10,
"quantity": 1,
"repositoryId": "xci1000009",
"status": 0,
"totalPrice": 10
},
{
"description": "Gumdrop Lamp",
"discountedTotalPrice": 24.65,
"price": 29,
"quantity": 1,
"repositoryId": "xci1000010",
"status": 0,
"totalPrice": 29
}
]
}

```

```

    }
  ],
  "orderTotal": 566.35,
  "orderTotalCurrencyCode": "USD",
  "originator": 0,
  "repositoryId": "xco30098",
  "status": 2,
  "submittedDate": 1340624552
}
}]

```

#### Get Orders Statuses

```
curl -v -b cookies.txt -X POST -H "Content-Type: application/json" -d "{atg-rest-depth:3}" https://pt-asery-olx6-2.company.com:8282/rest/bean/atg/commerce/OrderServices/getOrderStatuses
```

#### Response:

```

{"atgResponse": {
  "0": "Incomplete",
  "1": "Submitted to fulfillment",
  "10": "Pending fulfillment",
  "11": "Order acknowledged by SAP",
  "12": "Modify request pending for OMS notification",
  "13": "Remove request pending for OMS notification",
  "14": "Order acknowledged by OMS",
  "2": "Processing",
  "3": "Pending removal",
  "4": "Removed",
  "4000": "Pending return",
  "4001": "Scheduled order template",
  "4002": "Pending agent approval",
  "4003": "Cancelled by agent",
  "5": "Fulfilled",
  "5000": "The order is pending approval",
  "5001": "The order has been approved",
  "5002": "The order failed approval",
  "6": "Failed",
  "7": "Pending merchant action",
  "8": "Pending customer action",
  "9": "This order is a quote"
}}

```

#### Get Orders Originators

```
curl -v -b cookies.txt -X POST -H "Content-Type: application/json" -d "{atg-rest-depth:3}" https://pt-asery-olx6-2.company.com:8282/rest/bean/atg/commerce/OrderServices/getOrderOriginators
```

#### Response:

```

{"atgResponse": {
  "0": "default",
  "1": "scheduledOrder",
  "2": "contactCenter"
}}

```

## Customization

**Note:** For details on the data fields RightNow accepts, see the RightNow technical documentation on the Connect Web Services for SOAP.

The mapping of ATG profiles and RightNow contacts is based on three XML files, which are fully customizable.

For bulk and incremental asynchronous updates, the profile-output-config.xml file is used because the DAF.Routing API is used:

```
<item item-descriptor-name="user" is-document="true">
  <!-- Use title as a name of RightNow object -->
  <title property-name="Contact" is-non-repository-property="true"/>
  <meta-properties>
    <property name="login" type="string" store-as-meta-index="true" output-name="login"/>

    <!-- Email object -->
    <property name="email" type="string" store-as-meta-index="true" output-name="emails.emailList.email"
property-accessor="/atg/integration/rightnow/EmailPropertyAccessor"/>

    <property name="firstName" type="string" store-as-meta-index="true" output-name="name.first"/>
    <property name="lastName" type="string" store-as-meta-index="true" output-name="name.last" />

    <!-- Mobile phone object -->
    <property name="daytimeTelephoneNumber" type="string" store-as-meta-index="true" output-
name="phones.phoneList.phone" property-
accessor="/atg/integration/rightnow/MobilePhonePropertyAccessor" />
  </meta-properties>

  <!-- Use home address as primary contact address -->
  <item property-name="homeAddress" is-multi="false">
    <meta-properties>
      <property name="city" type="string" store-as-meta-index="true" output-name="address.city" />
      <property name="country" type="string" store-as-meta-index="true" output-name="address.country" />
      <property name="postalCode" type="string" store-as-meta-index="true" output-
name="address.postalCode" />
      <property name="state" type="string" store-as-meta-index="true" output-name="address.stateOrProvince"
/>

      <!-- Combine address1 and address2. Listen to changes in both properties -->
      <property name="address1" type="string" store-as-meta-index="true" output-name="address.street"
property-accessor="/atg/integration/rightnow/AddressPropertyAccessor"/>
      <property name="address2" type="string" store-as-meta-index="true" output-name="address.street"
property-accessor="/atg/integration/rightnow/AddressPropertyAccessor"/>

      <!-- Home phone object -->
      <property name="phoneNumber" type="string" store-as-meta-index="true" output-
name="phones.phoneList.phone" property-
accessor="/atg/integration/rightnow/HomePhonePropertyAccessor" />
    </meta-properties>
  </item>
</item>
```



Every ATG property maps directly to the RightNow property. If additional logic is required, the property-accessor component is used; for example, AddressPropertyAccessor creates a single string from address1 and address2 properties. The developer on behalf of the ATG customer can extend this customization by adding additional mapping entries and new accessor components.

Filtering configuration is used for REST API purposes and required by the parent REST module:

```
<rest-filtering>

  <!-- External user profile and home address -->
  <component name="/atg/userprofiling/ProfileAdapterRepository" default-include="false">
    <item-descriptor name="user">
      <property name="ID" property-
customizer="atg.integration.rightnow.rest.filtering.RightNowIdPropertyCustomizerImpl" />
      <property name="newPassword" target="password" hidden="true" />
      <property name="address" target="homeAddress" />
      <property name="email" hidden="false" />
      <property name="firstName" hidden="false" />
      <property name="lastName" hidden="false" />
      <property name="login" hidden="false" />
      <property name="middleName" hidden="false" />
      <property name="daytimeTelephoneNumber" hidden="false" />
    </item-descriptor>

    <item-descriptor name="contactInfo">
      <property name="city" hidden="false" />
      <property name="country" hidden="false" />
      <property name="postalCode" hidden="false" />
      <property name="stateOrProvince" target="state" />
      <property name="street" target="address1" />
      <property name="phoneNumber" hidden="false" />
    </item-descriptor>
  </component>

  <!-- Order -->
  <component name="/atg/commerce/order/OrderRepository" default-include="false">
    <item-descriptor name="order" >
      <property name="orderId" target="repositoryId" />
      <property name="contactId" property-
customizer="atg.integration.rightnow.rest.filtering.OrderRightNowIdPropertyCustomizer" />
      <property name="contactFirstName" property-
customizer="atg.integration.rightnow.rest.filtering.OrderProfileFirstNameCustomizer" />
      <property name="contactLastName" property-
customizer="atg.integration.rightnow.rest.filtering.OrderProfileLastNameCustomizer" />
      <property name="repositoryId" hidden="false" />
      <property name="orderTotal" property-
customizer="atg.integration.rightnow.rest.filtering.PriceTotalPropertyCustomizerImpl" />
      <property name="orderTotalCurrencyCode" target="priceInfo.currencyCode" />
      <property name="numberOfItems" property-
customizer="atg.integration.rightnow.rest.filtering.NumberOfItemsPropertyCustomizerImpl" />
      <property name="numberOfItemsReturned" property-
customizer="atg.integration.rightnow.rest.filtering.NumberOfItemsReturnedPropertyCustomizerImpl" />
      <property name="itemSummary" property-
customizer="atg.integration.rightnow.rest.filtering.ItemSummaryPropertyCustomizerImpl" />
      <property name="submittedDate" property-
customizer="atg.integration.rightnow.rest.filtering.DatePropertyCustomizerImpl" />
    </item-descriptor>
  </component>
</rest-filtering>
```

```

    <property name="status" property-
customizer="atg.integration.rightnow.rest.filtering.OrderStatePropertyCustomizer" />
    <property name="originator" property-
customizer="atg.integration.rightnow.rest.filtering.OriginatorPropertyCustomizer" />
    <property name="orderItems" target="commerceItems" />
</item-descriptor>

<item-descriptor name="commerceItem">
    <property name="description" hidden="false" property-
customizer="atg.integration.rightnow.rest.filtering.ItemDescriptionPropertyCustomizer" />
    <property name="quantity" hidden="false" />
    <property name="status" property-
customizer="atg.integration.rightnow.rest.filtering.CommerceItemStatePropertyCustomizer" />
    <property name="price" property-
customizer="atg.integration.rightnow.rest.filtering.CommerceItemListPricePropertyCustomizer" />
    <property name="totalPrice" property-
customizer="atg.integration.rightnow.rest.filtering.CommerceItemTotalPricePropertyCustomizer" />
    <property name="discountedTotalPrice" property-
customizer="atg.integration.rightnow.rest.filtering.CommerceItemAmountPropertyCustomizer" />
</item-descriptor>
</component>

<!-- Internal user -->
<component name="/atg/userprofiling/InternalProfileRepository" default-include="false">
    <item-descriptor name="agent">
        <property name="ID" property-
customizer="atg.integration.rightnow.rest.filtering.RightNowIdPropertyCustomizerImpl" />
        <property name="newPassword" target="password" hidden="true"/>
        <property name="email" hidden="false" />
        <property name="firstName" hidden="false" />
        <property name="lastName" hidden="false" />
        <property name="login" hidden="false" />
    </item-descriptor>
</component>
</rest-filtering>

```

The developer can modify this file to change the list of properties to present to RightNow. If additional logic is required to access property value, the property-customizer component is used.

The puller service uses its own mapping XML file:

```

<mappings>

    <mapping left="com.rightnow.ws.objects.Contact" right="atg.repository.RepositoryItem" variant="rightnow-
integration">

        <route>
            <left name="login"/>
            <right name="login"/>
        </route>

        <route>
            <left name="" accessor="/atg/integration/rightnow/mapping/EMailAccessor">
                <attributes>
                    <entry key="atg.integration.rightnow.mapping.EMailAccessor.addressTypeID" value="0"/>
                </attributes>
            </left>

```

```

    <right name="email"/>
  </route>

  <route>
    <left name="name.first"/>
    <right name="firstName"/>
  </route>

  <route>
    <left name="name.last"/>
    <right name="lastName"/>
  </route>

  <route>
    <attributes>
      <entry key="atg.mapping.variant" value="rightnow-integration"/>
    </attributes>
    <left name="address"/>
    <right name="homeAddress"/>
  </route>

  <route>
    <left name="" accessor="/atg/integration/rightnow/mapping/PhoneAccessor">
      <attributes>
        <entry key="atg.integration.rightnow.mapping.PhoneAccessor.phoneTypeID" value="4"/>
      </attributes>
    </left>
    <right name="homeAddress.phoneNumber"/>
  </route>

  <route>
    <left name="" accessor="/atg/integration/rightnow/mapping/PhoneAccessor">
      <attributes>
        <entry key="atg.integration.rightnow.mapping.PhoneAccessor.phoneTypeID" value="1"/>
      </attributes>
    </left>
    <right name="daytimeTelephoneNumber"/>
  </route>

</mapping>

<mapping left="com.rightnow.ws.objects.Address" right="atg.repository.RepositoryItem" variant="rightnow-integration">

  <route>
    <left name="street"/>
    <right name="" accessor="/atg/integration/rightnow/mapping/StreetAccessor"/>
  </route>

  <route>
    <left name="city"/>
    <right name="city"/>
  </route>

  <route>
    <left name="country.name"/>
    <right name="country"/>
  </route>

```

```
<route>
  <left name="stateOrProvince.name"/>
  <right name="state"/>
</route>

<route>
  <left name="postalCode"/>
  <right name="postalCode"/>
</route>

</mapping>
</mappings>
```

Every ATG property is mapped to the relevant RightNow property. The developer can change mapping by modifying the XML file. No additional changes are required.

## Conclusion

The capabilities provided by this integration allow businesses to better serve their customers, provide better experiences, and increase call center agent efficiency. These benefits are enabled by capabilities of this integration, namely the synchronizing of customer information, and the ability to improve efficiency of the tool sets.

Together, Oracle Commerce and Oracle RightNow form the core of a superior customer experience solution to attract, sell, support, and service customers.





ATG-RightNow Integration  
August 2014  
Author: Aliaksei Sery  
Contributing Author: Steve Garriott

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2015, 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 of The Open Group. 0113

**Hardware and Software, Engineered to Work Together**