Oracle B2C Service

Integration and Extensibility Frameworks

ORACLE TECHNICAL BRIEF/ APRIL, 2019

ORACLE

TABLE OF CONTENTS

INTRODUCTION

This white paper provides a detailed overview of the integration and extension capabilities available within Oracle B2C Service that provide a connected application experience to our customers.

INTENDED AUDIENCE

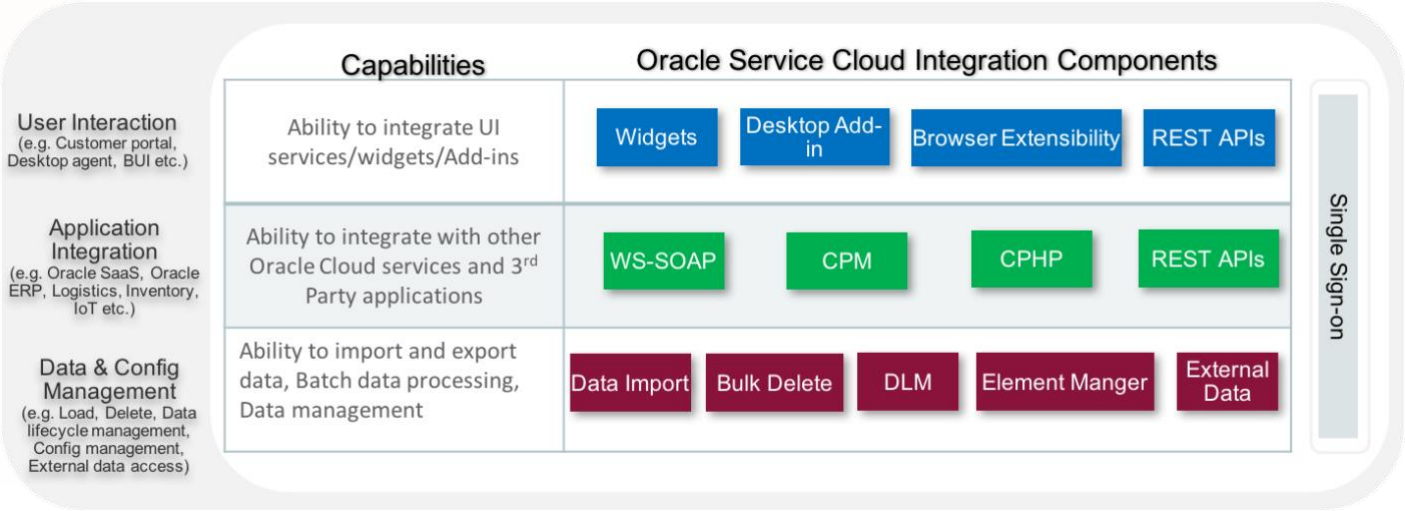
This technical brief is intended to provide best practices for the integration approaches and extensibility design patterns for Oracle B2C Service to developers, systems integrators, technical consultants and analysts.

ORACLE B2C SERVICE PLATFORM - INTEGRATION APPROACH

Most customer experience centers operate in a tightly managed environment where calls are routed to agents in real time. Most agent activity involves taking orders, troubleshooting, handling returns and exchanges, or fielding account inquiries. To provide a better customer experience, they need access to information and tools across disparate enterprise applications to meet a customer's need.

Oracle B2C Service provides frameworks, accelerators, connectors to help contact centers achieve this unified connected experience. The Oracle B2C Service platform provides a robust set of backwards-compatible open-standards-based public APIs to build both server-side and client-side integrations. The ability to integrate with other applications is one of the important benefits of the Oracle B2C Service Platform.

The image below provides an overview of the various integration capabilities available in the Oracle B2C Service platform across the UI, Application and Data layer along with security considerations.



Overview of OSvC Integration Capabilities

The integration capabilities are generally based on Connect Common Object Model (CCOM) and Managed Frameworks that provide common and standard means of building APIs across the B2C Service functionalities.

Oracle B2C Service customers can leverage these capabilities to integrate various CTI providers into the UI or leverage the APIs to integrate the Oracle CX suite of SaaS applications, on-premises ERP or legacy applications, and third-party business applications.

The integration capabilities are broken down into the following topics which are covered in detail in subsequent sections.

- Application Integration
- Desktop Integration
- Data Management
- Configuration Management
- Packaged Integrations

APPLICATION INTEGRATION

The enterprise need of integrating with Oracle B2C Service can be accomplished through API services that are provided with every B2C Service instance. The Oracle B2C Service APIs are built on a managed framework and leverage the Connect Common Object Model (CCOM) to access data within a Oracle B2C Service site.

The API bindings offered on all Oracle B2C Service instances are Connect REST (CREST), Connect Web Service for SOAP (CWSS), and Connect for PHP (CPHP). These APIs are all backward compatible public APIs.

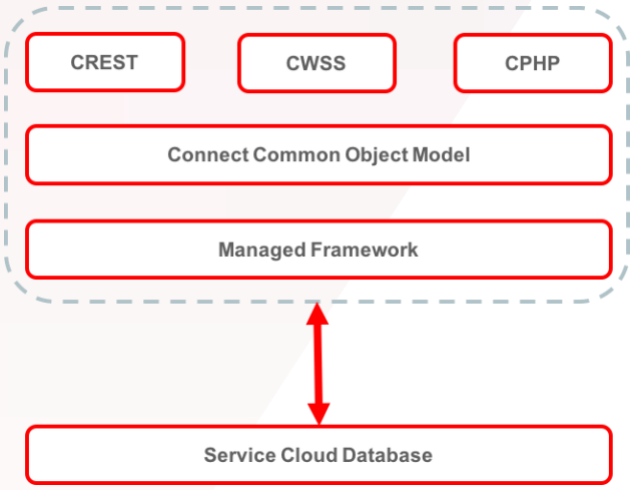
This section describes the following key concepts and describes the various uses of the APIs.

- Managed Framework
- Connect Common Object Model
- Connect REST (CREST)
- Connect Web Service for SOAP (CWSS)
- Connect for PHP (CPHP)

Managed Framework

Oracle B2C Service's managed frameworks are developed and subsequently versioned to allow integrations to be written once. Breaking changes are not introduced into versions of the APIs, so integrations can be expected to run without impact even as newer API versions are brought online. New versions only need to be used when an organization has a need for new functionality published in a later version.

Managed frameworks are available to use through the Connect APIs. Connect APIs are product release agnostic, which ensures backward compatibility so that your integrations will continue to work on new product releases. So, zero maintenance! (you can read more about Connect APIs here.)



Managed Framework

The Connect APIs are all versioned. We periodically release new Connect versions that introduce new features and capabilities to the framework. When integrations are written against a specific Connect version, they will continue to work between upgrades as long as the integration stays on the same connect version.

Connect Common Object Model

The Connect Common Object Model (CCOM), provides a unified developer experience and access to customer's data stored in Oracle B2C Service's extensible schema. CCOM provides a well-defined approach to accessing data within Oracle B2C Service tables such as incidents, contacts, organizations, tasks, and more.

CCOM defines a set of primary objects, sub-objects, delta lists, and helper objects that are used when invoking operations on the service. Primary objects are objects with a unique ID (primary key) which can be directly created, read, updated, and deleted. Sub-objects are subordinate to primary objects. That is, primary objects contain sub-objects and the only way to manage the sub-objects is through CRUD operations on the primary objects.

In addition to standard objects, Oracle B2C Service supports creating custom objects. Custom objects are available within the Agent Desktop workspace, and automatically available as CCOM objects that can be used with the B2C Service APIs. The advantage of custom objects is that Oracle B2C Service can be extended to meet specific business needs and use cases including Desktop and Application integrations.

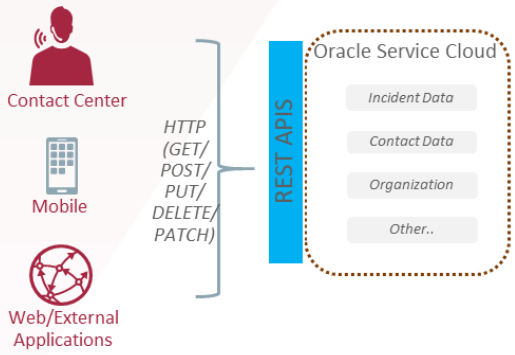
A common use case in the retail industry is to process returns using business-specific RMA procedures. There are a couple of ways to accomplish this use case with Oracle B2C Service. The first is to utilize the existing incident object to capture RMA specific information from the end users. This information could be details like the product being returned, serial number and reason for return. All of this information can be collected and stored within the incident fields.

For businesses that require collection of more detailed RMA information, an alternate approach is to define an RMA custom object within Oracle B2C Service schema. The RMA custom object can have an association with the incident object, so that the end-user can initiate the RMA process by creating an incident and filling out the related RMA fields that are stored within the custom object. Since custom objects are available via the Oracle B2C Service APIs, the RMA data can be integrated with an external ERP system for fulfillment.

Additional details on CCOM can be found in the API documentation: Introduction to CCOM

Connect REST API

Connect REST (CREST) API is a public API that leverages the Connect Common Object Model versions 1.3 and above. It supports CRUD operations and allows customers and partners to integrate with the Oracle B2C Service Platform (formerly Service Cloud Platform). Web and mobile developers looking to implement RESTful pattern while integrating with Oracle B2C Service should leverage the Connect REST API.



CREST API Overview

Oracle B2C Service ensures proper security mechanisms are in place while using Connect REST API. Connect REST API supports basic, session, and OAuth authentication mechanisms to ensure secure network data access.

Integrating with Oracle B2C Service REST APIs is easy and accessible through custom client applications developed using languages such as Java, JavaScript, Ruby, and others. The CREST API support CRUD (Create, Read, Update and Delete) operations using POST, PATCH, GET, and DELETE requests on the resources for standard and custom objects.

The syntax of a REST request to perform CRUD actions on Oracle B2C Service Objects is formed like:

```
https://<sitename>/services/rest/connect/version/resource
```

In the above format, version represents the API version (1.3 or 1.4) and the resource represents the object to invoke. Both standard and custom objects are available through the CREST API along with several other useful features like executing analytics reports, manage file attachments, and more.

Using the above syntax, the following example request brings back all the fields for the incident with id = 1 using a GET request.

```
REQUEST: GET

https://mysite.example.com/services/rest/connect/v1.4/incidents/1
```

```
Sample RESPONSE Body

{

    "items": [

        {

            "id": 1,

            "lookupName": "100909-000000",

            "createdTime": "2013-08-21T21:13:23Z",

            "updatedTime": "2013-11-11T21:00:25Z",

            "links": [

                {

                 ….

                 ….

                }

….

….

}
```

Custom objects are also available via the REST API, and therefore all CREST API operations are supported on those objects. Once a custom object has been deployed, the following syntax is used to interact with it:

```
https://<sitename>/services/rest/connect/version/Package.ObjectName


ex:

https://<sitename>/services/rest/connect/1.4/Product.Returns
```

Single Sign On Using OAuth

An external identity provider (IdP) can be configured with Oracle B2C Service to provide credentials for making requests to the Connect REST API. The OAuth token, supplied by the external identity provider (IdP), can be passed in the authorization header in the REST request.

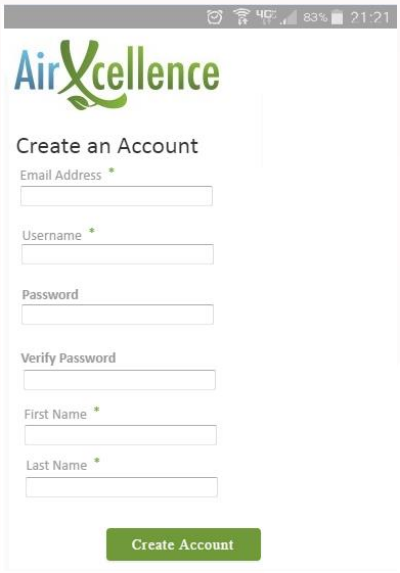
For more information, see Using OAuth authorization to access the Connect REST API.

The REST API is very powerful and makes integrating with Oracle B2C Service easy. Let's look at a simple use case to build product registration integration from a mobile device using the CREST API.

Step 1: Create a contact

The application will need to first create an account for the end user in B2C Service before the user can register their product. This can be done by creating a contact with details such as login, password and email address via the CREST API.

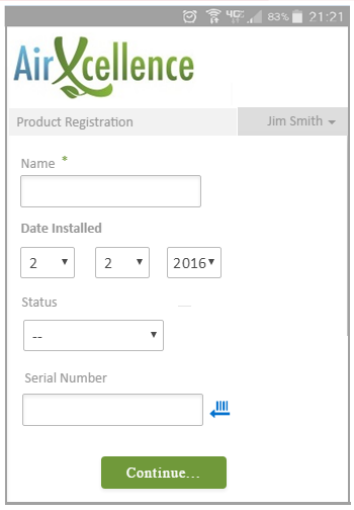
The image below shows an example of the contact fields that the user will need to provide to create their account in the system. Once the required fields are populated and submitted, the API request below shows the payload for the contact resource. This request will create the contact in B2C Service that will allow the user to go to the next step, which is to register their product.




```
REQUEST: POST

https://mysite.example.com/services/rest/connect/v1
.4/contacts

{

  "name": {

    "first": "Jim",

    "last": "Smith"

  },

  "login": "AirX.Jim",

  "emails": {

        . . . .

  }

}
```

Steps 2: Register the Product

The example image below shows the required product registration fields. These fields correspond to the asset object within B2C Service Thus, when the user fills the required fields and submits the data, the REST request to create an asset object within B2C Service is shown in the sample payload below. Once the request is submitted a new product registration record is created in the asset object.



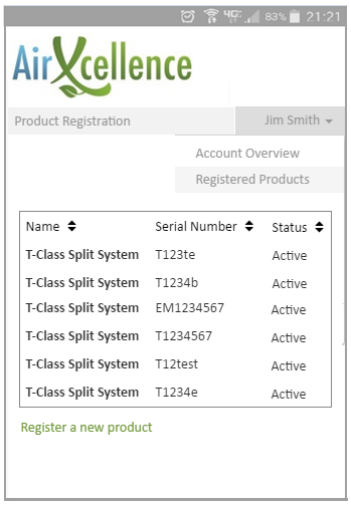

```
REQUEST: POST

https://mysite.example.com/services/rest/connect/v1.4/assets/

{

   "name": "MyProduct",

   "product": [ { "lookupName": "MyProduct" } ],

   "serialNumber": "T1234c",

   "InstalledDate" : "02-02-2016"

}
```

Step 3: View All Registered Products

The CREST API has function for executing analytics reports via the analyticsReportResults resource, that includes setting filter values in the request payload. In the product registration scenario, this function can be utilized to display all of the products that a user has registered.

To gather the list of registered products for a contact, invoke the API to create a "registered product for contact" report (not shown) that filters based on contact id. Then from the mobile application using the analyticsReportResults function an application can pass the report id (100010 in the example request) and the contact id as a filter to retrieve the list of all products for that contact.

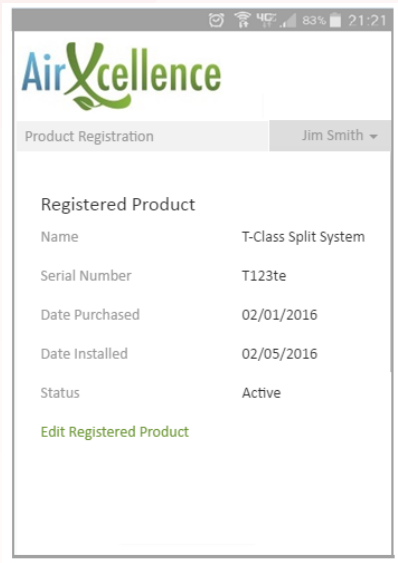
The sample REST request and the payload for invoking the report is show below.




```
REQUEST: POST

https://mysite.example.com/services/rest/connect

/v1.4/analyticsReportResults

{

   "id":100010,

   "limit":10

    "filters": [{

         "name": "ContactID",

          "values": <ID>

    }]

}
```

Step 4: View a Registered Product

Once all of the registered products for the user are displayed in the mobile application, we can make a GET request to perform a read operation against the asset object to display all the data for the desired registration. The REST request for retrieving the asset data is shown below along with the response displayed in the example image.




```
REQUEST: GET

https://mysite.example.com/services/rest/connect/v1.4/assets
    /12
```

The scenario above highlights a few capabilities of the CREST API in building mobile integrations very easily. The full list of API capabilities and additional details on CREST can be found in the API documentation: REST API for Oracle B2C Service

Connect Web Services for SOAP

Connect Web Services for SOAP (CWSS) provides application developers with a set of SOAP-based services used to securely access and modify data contained in the Oracle B2C Service platform. It can also be used in conjunction with the Desktop Add-Ins API (.NET API), or from any other application that requires access to data through a SOAP API.

Oracle B2C Service publishes three versioned WSDL files that provide access to Connect Web Services.

Standard WSDL: This WSDL should be used by application developers building integrations specific to their Oracle B2C Service instance. The typed WSDL is a strongly typed representation of the Connect Common Object Model. The typed WSDL can be accessed from the following URL with a standard HTTP GET request.

```
https://<sitename>/services/soap/connect/soap?wsdl=typed
```

Partner WSDL: This WSDL should be used by application developers building integrations that function across multiple instances of Oracle B2C Service. It provides the ability to work with Generic Objects that can be used across multiple instances of Oracle B2C Service. The generic WSDL can be accessed from the following URL with a standard HTTP GET request.

```
https://<your_site>/services/soap/connect/soap?wsdl=generic
```

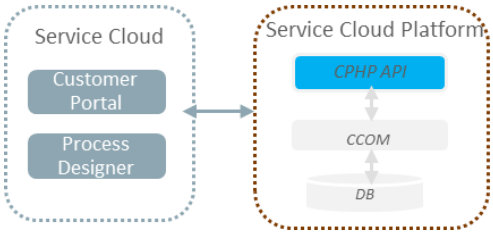
Event WSDL: This WSDL should be used by application developers who want to build an event notification service that allows external applications to discover subscriber objects in Oracle B2C Service. The event WSDL can be accessed from the following URL with a standard HTTP GET request.

```
https://<your_site>/services/soap/connect/soap?wsdl=event
```

The full list of API capabilities and CWSS sample code can be found in the API documentation: Connect Web Services for SOAP

Connect for PHP

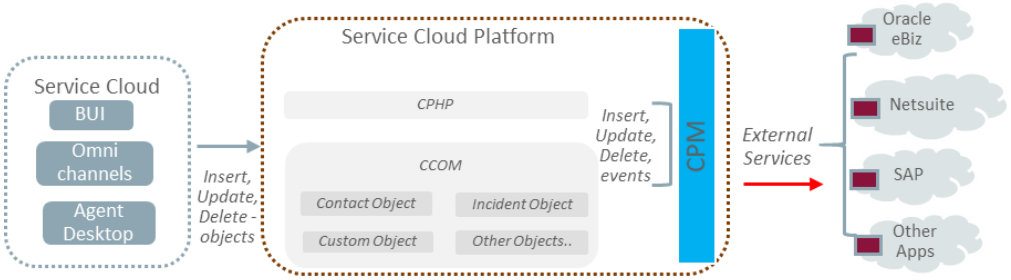
Connect PHP API is used to work with CCOM with processes that run on the Oracle B2C Service infrastructure. The Connect PHP API is generally used via Oracle B2C Service Customer Portal or Oracle B2C Service Process Designer modules, allowing developers to customize the customer experience using Customer Portal Framework and develop business logic in Custom Process Models.



Overview of Connect for PHP

Custom Processes

Custom processes are predefined object event handlers (PHP scripts) that run as the result of an event occurring on standard or custom objects. This is extremely useful for external integrations and advanced business logic implementations.



Overview of Custom Process Models

Object event handlers can help your organization automate tasks based on the events that trigger them. For instance, when an incident is updated, an object event handler can be triggered to update a record in an external system. Event handlers can also be used in rules, surveys, and campaigns.

A common use case for using Custom processes is for data integration with an external web service. The object handler PHP script link below adds a Note to an Incident when it is created in B2C Service.

An object event handler PHP script consists of two classes: the *ObjectEventHandler* and the *ObjectEventHandler_TestHarness*. The *ObjectEventHandler* class contains the logic for the actions performed when the event handler is invoked. In this example, the *ObjectEventHandler* class has logic to retrieve the external id by invoking the webservice using cUrl and creates a note with the external id in the incident record.

The *ObjectEventHandler_TestHarness* class contains test cases for all of the expected conditions and logic of the object event handler script. The tests ensure proper behavior of the corresponding ObjectEventHandler prior to deployment.

Example code for the Object Event Handler script can be found in the product documentation: Example Event Handler Script

Additional details on configuring and deploying CPMs on B2C Service can be found in the product documentation: Custom Processes

Customer Portal

Customer Portal is a web application framework and out-of-the-box web page set for managing and creating web experiences for customer service. The self-service pages use the Customer Portal (CP) framework, providing a rich set of widgets for building dynamic customizations for specific business needs. The Customer Portal framework utilizes the Model-View-Controller (MVC) design pattern which allows the separation of logic and presentation so that each can vary independently.

The CP framework provides standard widgets for various B2C Service features such as knowledge base search, chat, policy automation and other functions. There are more than one hundred widgets in the CP standard widgets collection.

In addition to the standard widgets, customers can also create custom widgets by extending the functionality of a standard widget or creating custom widgets from scratch. These widgets can be embedded within Customer Portal pages and leverage the Connect PHP API.

Additional details on the Customer Portal Framework are found in the product documentation: Customer Portal

DESKTOP INTEGRATION

Oracle B2C Service offers a .NET-based Agent and Administration Desktop, as well as an alternative Browser User Interface. The desktop integration APIs provide a robust set of capabilities to extend, automate and integrate custom business functions directly in the Agent Desktop.

Very often businesses deploy several applications along with B2C Service such as a Telephone Interface system, Inventory Management application or an internal tool used by agents. Using B2C Service's Desktop Integration Framework, these applications can be integrated and displayed within the B2C Service Agent Desktop, providing agents with a unified application experience.

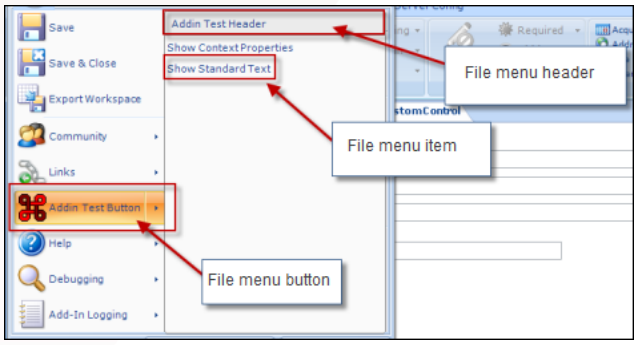
The next few sections will cover the desktop integration APIs available for the B2C Service Agent Desktop:

- Desktop Add-In Framework (.NET API)
- Agent Browser UI Extensibility Framework
- Desktop Integration JavaScript API

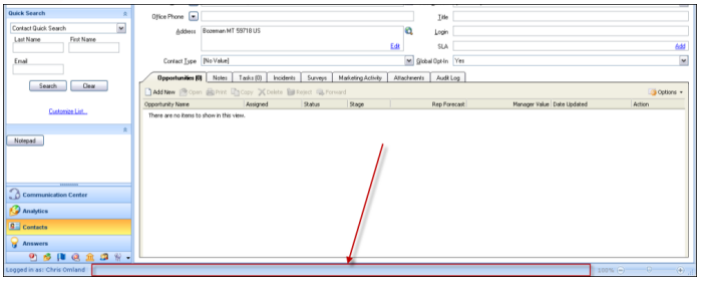
Desktop Add-In Framework

The desktop Add-In framework is a .NET based API that provides components to extend, automate and integrate the B2C Service Agent Desktop. This API allows customers to create various UI components and custom controls and deploy them as .NET Add-Ins within the Agent Desktop.

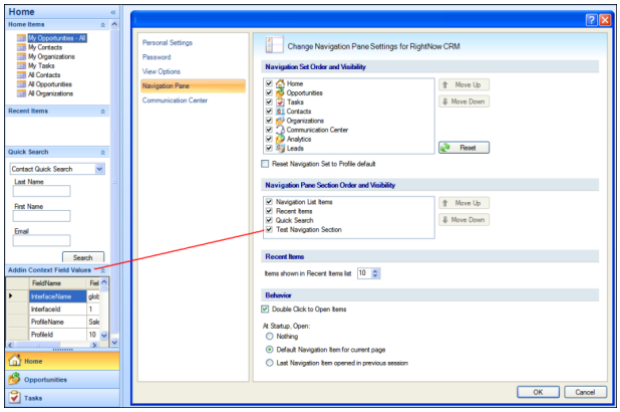
.NET Add-Ins can be added to the different visual areas of the agent desktop such as the application menu, status bar, workspaces or navigation pane to achieve objectives such as automate steps within the application, integrate with external systems, or create entirely new and real-time extensions within the agent desktop.



Application Menu Add-In



Status Bar Add-In



Navigation Item

In addition to UI components, the Add-In framework provides several events such as login, logout, workspace and report refresh and several other events to support automation.

Computer Telephony Integration (CTI) is a common use case used in the Agent Desktop that utilizes the Add-In framework for UI control and the CWSS for data access. The CTI toolbar provides a unified agent experience to manage phone requests using from within the B2C Service desktop.

CTI Toolbar Add-In

The CTI toolbar reference implementation is published as an accelerator along with documentation, sample code and other resources available for download here: CTI Accelerator

The full list of API capabilities and sample code for the various Add-In's can be found in the API documentation: Connect AddIn Framework

Agent Browser UI Extensibility Framework

The Agent Browser User Interface (BUI) Extensibility Framework is the API framework that allows to write JavaScript code to create custom controls and UI components for the Agent Browser User Interface. Use these APIs to build components (we call them "Extensions") to extend, automate and integrate the Agent Browser User Interface and provide a more unified agent experience.

If you are using the Agent Browser UI and plan to extend the functionality of the Agent Desktop then you should use the Extensibility Framework. This API allows you to create custom components and functionality on the BUI desktop client to achieve the following:

1. Extend the functionality of the BUI desktop to build a new application such a CTI, RMA, Order Management, etc.

2. Automate an action such as automatically set a field value, trigger workspace rules and/or remove any manual steps to improve agent effectiveness.

3. Integrate with external systems to read and write data from and into Oracle B2C Service objects (such as contact, incident, custom objects, etc.) to provide real time data to agents.

The BUI Extensibility Framework also allows you to build custom UI controls, components and applications using a rich set of events and functions that allow you to access the visual components and console events available in the Agent Browser UI.

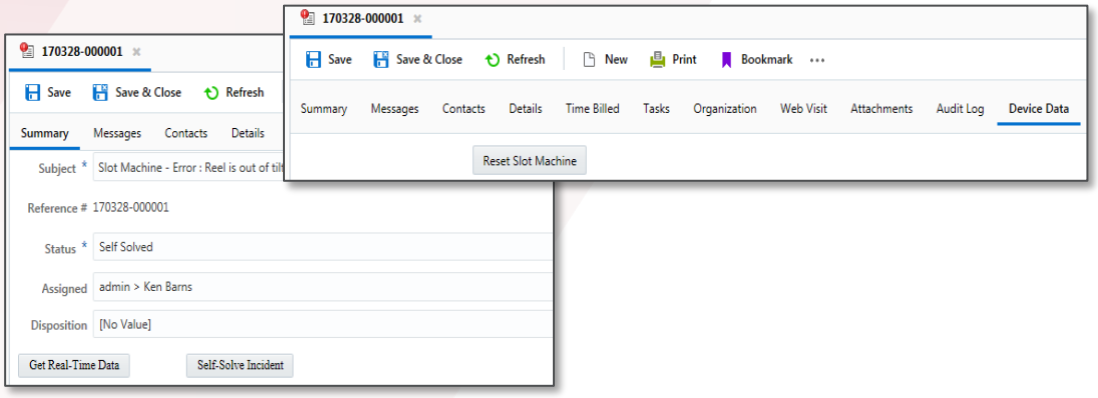
There are four types of extensions that can be created using the Extensibility Framework:

Library Extensions

Library extensions can be used to create a shared libraries or common functions that can be used by various extensions deployed on the site.

Workspace Extensions

Workspace extensions allow you to add functionality to Agent Browser UI workspaces such as incidents, contacts, custom objects and others.



Workspace Extensions

Console Extensions

Console extensions allow you to build UI components for the Agent Browser UI desktop. Examples of these extensions include the following:

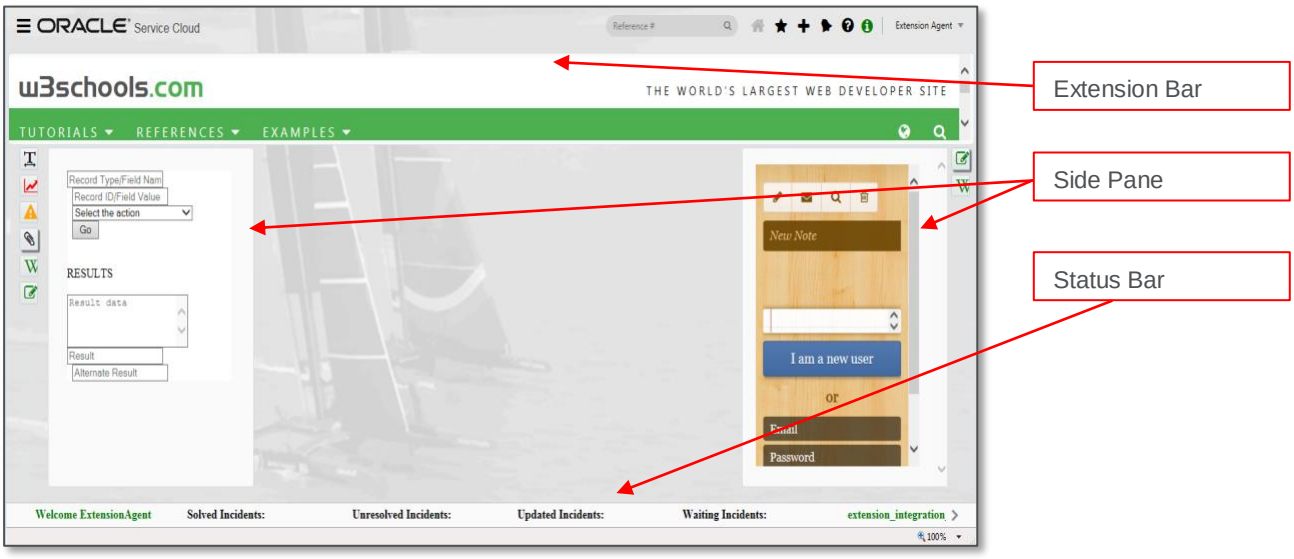
Content Pane: Load an external application or create a custom workspace inside a tab.

Navigation items: Add items to the Navigation Bar.

Status Bar: Build a UI component at the bottom of the screen.

Side Pane: Create a collapsible UI on the left or right within the agent desktop.

Extension Bar: This is a UI component that can be docked to left, right or across the top of the desktop. The extension bar can be used to house a CTI media bar or an external lookup tool.

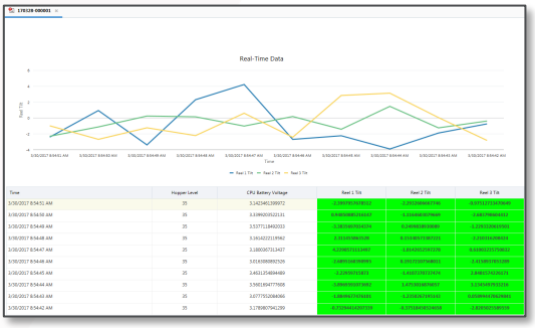


Console Extensions

In addition to the above there are other Console UI components available such as the Header Menu's, Popup and Modal windows that allow you to further enrich the UI for improved agent experience.

Analytics Extensions

Analytics extensions provide the ability to interact with reports in the Browser UI Desktop. Capabilities include the ability to create a report definition, get and modify filter lists, and fetch report row/column data.



Analytics Extensions

Analytics for External Data

The extensibility framework also supports creating virtual tables to retrieve and display data at real time using reports within the Agent Browser UI Desktop.

These virtual tables can be used to create reports that retrieve external data and populate the virtual tables with external data at runtime. These reports support the standard table properties including table name, column names, filters, and filter names.

Events and Functions

There are several events and functions available to interact and access data and events on the desktop. These events are available as global events and workspace events.

Global Extensions can use the GlobalContext object to fetch information such as interface name and url, agent id, agent session id, language name and id.

Workspace Extensions have access to events/operations and functions to interact and add functionality to workspaces. These events and operations allow extensions to:

- Listen to events such as loading, closing or saving workspaces.
- Get and Set field values on a workspace record.
- Perform workspace operations such as create or edit workspace records, focus on tabs and trigger named events.

Browser Control

The Browser Control provides a simple and easy way to load and integrate an external web application inside the Agent Browser UI and make use of the Extensibility Framework to interact with workspace records.

You can find examples and tutorials for the Extensibility Framework on the Developer Community: Agent Browser Extensibility Framework Quick Start Guide

Additional details about the Agent Browser UI Extensibility Framework can be found in the API documentation: Agent Browser UI Extensibility Framework

Desktop Integration: JavaScript API

The JavaScript API enables customers and partners to integrate data contained on a workspace with a web page that is contained on the workspace. This API provides an easy way to integrate data from an external web page to update or read workspace data from the web page.

For example, contact lookup from an external address validation page is a very common use case that can be accomplished using the JavaScript API. Typically, the external web page containing the contact lookup tool is embedded into a contact workspace via a browser control. The information presented about the contact record on the webpage can be used to update the workspace fields on the Contact record using the JavaScript API.

The JavaScript API is only available for integrating external web pages that are embedded in the workspace browser control. The JavaScript API supports access to standard objects (incidents, contacts, orgs and opportunity), custom objects and workspace events.

The example code below will read contact fields on the contact workspace and on a button click and display the contact data inside a text area on an external web page.

```
<script type="text/javascript">

function populate()

    {

        var contact = window.external.Contact;

        if (contact == null)

            alert("null contact");

        var contactString = "Name: " + contact.FullName + "\nEmail: " +
contact.EmailAddr + "\nc_id: " + contact.Id;

        data.ta.value = contactString;

    }

</script>


<form name='data'>

    <textarea name='ta' cols=40 rows=8></textarea>
```

```
</form>

<input type='button' value='Populate' onclick='populate()' />
```

The above code is an example to demonstrate how easy it is to integrate an external web page with a B2C Service workspace using the JavaScript API. Additional details about the JavaScript API can be found in the API documentation: JavaScript API

DATA MANAGEMENT

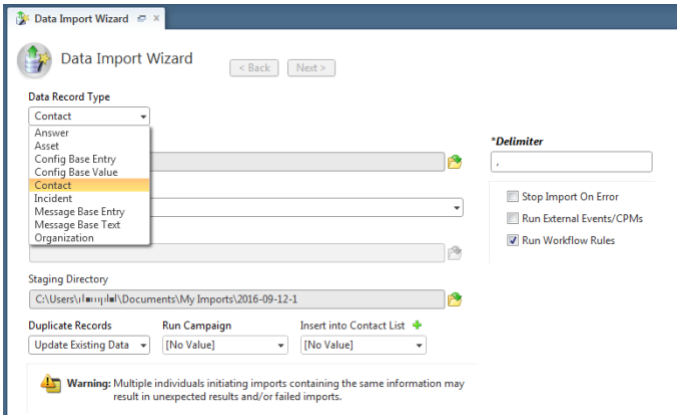
Oracle B2C Service provides tools and APIs to help customers manage their B2C Service data and artifacts. By leveraging these, customers can import data into B2C Service, or optimize their storage resources by exporting and deleting data efficiently. This section will cover the following topics in detail:

- Data Import
- Data Export
- Data Delete
- Element Manager

Data Import

Data Import Wizard enables you to add new contacts, answers, assets, incidents, organizations, and custom objects to Oracle B2C Service database from a data file. For example, using this tool one can import lists of prospects or recently acquired customers into your knowledge base as new contacts or update existing answers to keep your database synchronized with an external knowledge base.

For imports that you want to perform more than once, you can create column mapping templates that can be selected when importing data with the wizard. Customers can customize a data import template via the template designer.



Data Import Wizard

Additional details on Data Import is available in the product documentation: Data Import Wizard

Data Import using Connect APIs

For imports that require processing logic, such as data cross references, clean up or additional business logic, the Data Import Wizard will be unable to support such tasks. For such complex data import tasks, an import script can be written using the Connect APIs (CREST, CWSS and CPHP) to accommodate for the additional processing logic to be written into the import script and ensure a successful data import.

Best Practices for Data Imports

Most of the times data imported into OSvC does not necessarily need to be in the B2C Service database, especially if the imported data is used for foreign key references or historical context. Therefore, it is recommended that the data reside in an external system and be surfaced within B2C Service as needed using an integration.

The advantage with this approach is that B2C Service database is not loaded with data that is needed only for references. This allows the B2C Service database to be lean and helps to ensure optimal system performance.

Data Export

The RightNow Object Query Language (ROQL) provides a query subsystem that perform SQL-like queries against the Oracle B2C Service platform. The query language is based on a subset of Object Query Language (OQL), a text-based SQL-like query language that has been extended to handle object notation.

ROQL leverages the Connect Common Object Model (CCOM). The CCOM defines a set of primary objects, sub-objects, delta lists, and helper objects that are used when invoking operations on the service.

ROQL queries can be utilized to query and extract data out of the Oracle B2C Service database. The following sections provide details on various methods of extracting data using ROQL.

Queries using CREST

ROQL supports two different operations for querying data:

- Object Queries
- Tabular Queries

ROQL utilizes the “USE” keyword to specify which database the query executes against. Queries can be executed against the Operational (primary) database or the Reporting (Secondary) database.

Object queries

Object query retrieves a list of objects that are returned. Object queries are useful to extract all the fields on specific objects such as incident or contact.

```
https://<sitename>/services/rest/connect/v1.4/incidents?q=id>10 and id<15
```

Tabular Queries

Tabular query retrieves data in a tabular format similar to executing a normal SQL statement.

```
https://<sitename>/services/rest/connect/v1.4/queryResults? query=USE REPORT;SELECT
id, emails.address FROM contacts WHERE id = 10
```

Managed Tables

In addition to the objects available via the CCOM model, there are additional read-only data from Oracle B2C Service tables that can be queries using ROQL. These read only tables are exposed as Managed Tables. Some of the data available via the Managed Tables are transactions, deleted records, chat queue and several others.

Full list of Managed Tables supported in ROQL: Managed Tables List

The sample REST request below will query deleted incident data from the database

```
https://<sitename>/services/rest/connect/v1.4/queryResults?query=SELECT
incidentDeletedRecords.* FROM deletedRecordInformation WHERE
incidentDeletedRecords.id > 123;
```

Analytics Reports

While most of the OSvC data is available through the CCOM model, there may be additional table data that may not be directly available through ROQL.

For such cases where data is unavailable as CCOM data, the OSvC APIs provide the ability to execute reports to access such tables.

The sample REST request below will execute an report by passing a report filter.

Note: The report will need to be created using the Agent Desktop and the ID must be passed into the request body.

```
https://<sitename>/services/rest/connect/v1.3/analyticsReportResults


Body:

{

  "id": <Report ID>,

  "filters": [{

     "name": "contacts.any_phone",

     "values": "123-123-1234"

   }]

}
```

Batch operation Using CWSS API

The CWSS API includes a Batch operation that provides flexibility while extracting data. The batch operation can be used to send multiple heterogeneous operation requests to the server in a single SOAP request.

Additional information can be found in the API documentation: Batch Operation

API Limits

The OSvC APIs have limits on how much data can be extracted using ROQL.

A maximum of 20,000 rows can be returned in a single request. If the request generates more than 20,000 rows in the result, the first 20,000 rows will be returned. By default, when using the replication or reporting database, a maximum of 100,000 rows can be returned.

Additional API Limits can be found here: API Limits

Bulk Delete

Bulk delete APIs allow for deletion of large datasets efficiently with minimal impact on the resources and on operations of the site. Customers can leverage bulk delete API to purge incidents, opportunities, contacts, accounts, orgs and custom objects.

For example, B2C Service customer can use bulk delete API to purge old contacts who haven't contacted the service center in last 2 years.

The syntax for the bulk delete API call on the queryResults resource is as follows:

```
https://your_site_interface/services/rest/connect/version/queryResults/?query=semico
lon-separated ROQL DELETE queries
```

The syntax of ROQL DELETE query is as follows:

```
DELETE FROM primary object WHERE where clause expression ORDER BY fields LIMIT
number of rows
```

The following are examples of Delete Queries:

```
DELETE FROM Incidents LIMIT 1000; DELETE FROM Incidents LIMIT 1000; DELETE FROM Incidents
LIMIT 1000; DELETE FROM Incidents LIMIT 1000;


DELETE FROM Incidents WHERE ID >= 1 AND ID < 1000; DELETE FROM Incidents WHERE ID >= 1000
AND ID < 2000; DELETE FROM Incidents WHERE ID >= 2000 AND ID < 3000; DELETE FROM Incidents
WHERE ID >= 3000 AND ID < 4000; DELETE FROM Incidents WHERE ID >= 99000 and ID <100001;
```

Additional information on Bulk Delete can be found in the API documentation: Bulk Delete

CONFIGURATION MANAGEMENT

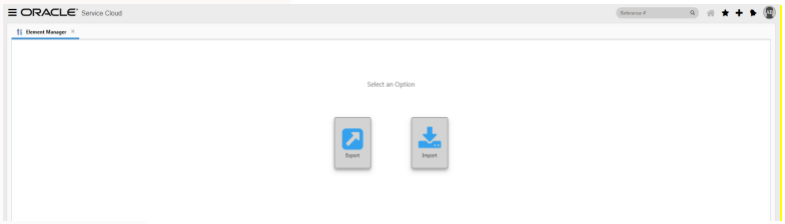
Element Manager

Element Manager allows configurable components to be grouped, packaged, migrated, and deployed across Oracle B2C Service instances in an automated way. It can be used to easily migrate configurations from development, test, and to production environments.

Element Manager allows the export and import of:

- Dashboards and Reports,
- Workspaces and Workflows,
- .NET Add-Ins and BUI Extensions

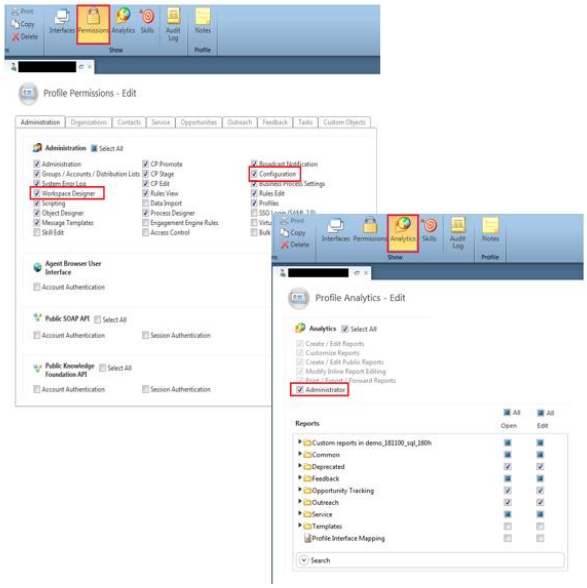
Element Manager is available via the Browser UI or using the available REST APIs.



Element Manager UI

Account Profile Settings

The account user using Element Manager needs to have the same administrator privileges as used to access the corresponding configuration managers. For example, to import/export Reports, the account user needs to have Analytics Administrator permission.



Element Manager: Account Profile Settings

To illustrate a use-case, Element Manager can be used to export a set of workspaces, that include custom reports, from the test environment to the production without having to worry about managing dependencies between workspaces and reports.

The REST APIs can also be used to automate time-consuming configurations by copying the configurations from site to site or interface to interface saving time and improving accuracy.

Additional details about Element Manager can be found in the product documentation: Element Manager

PACKAGED INTEGRATIONS

Oracle B2C Service Accelerators

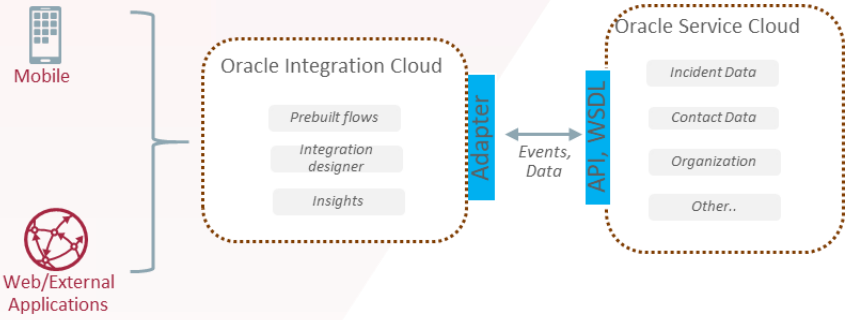
Oracle B2C Service Accelerators are designed to demonstrate how an integration scenario could be built using the public integration and extension capabilities of the Oracle B2C Service. Some of the accelerators available as sample reference sets are Twilio CTI Accelerator, IoT accelerator, Oracle Social Cloud Accelerator, Telephony and SMS Accelerator for Browser UI etc.

Let's say organizations want to provide their agents with a communication enabled dynamic desktop or UI interaction with customers, via calls or SMS. Organizations will need to integrate third-party telephony platform to achieve this capability.

One of the sample accelerators, Twilio CTI, helps customers by demonstrating the integration of the Agent Browser UI in Oracle B2C Service with Twilio, a third-party telephony and messaging service provider. This integration leverages Twilio's services with BUI UI controls to place and receive calls, manage an IVR, and send SMS messages, all with the relevant context.

Additional information on accelerators is available here: B2C Service Accelerators

Oracle iPaaS platform/adapters for Oracle B2C Service (formerly Service Cloud) Integrations



Oracle B2C Service is an ideal front-end for receiving and chronicling customer interactions initiated by phone, web, chat, email or social media. Customer service agents can use it to interact with customers, track the status of open issues, or—with the right integration—order replacement parts and dispatch technicians to customer locations. Links with the ERP system ensure a seamless flow of data and functions.

For example, when customers order parts during a service call, customer service agents need to verify that those parts are in stock. Accurate up-to-date data is required by the agent and real-time integrations into ERP can solve that issue. Oracle B2C Service can interface with inventory, order and other modules within Oracle E-Business Suite, SAP, Siebel, PeopleSoft, and JD Edwards to confirm the status of inventory and orders. Oracle B2C Service can also access Field Service data in an ERP system such as Oracle E-Business Suite to request a technician dispatch, improve service contract renewal rates, and initiate and track RMAs. This is where Oracle Integration Cloud helps our customers in providing this connected experience to our customers.

Oracle Integration Cloud simplifies building integrations in the cloud, connect securely to applications and services both in the cloud and on-premises, to avoid point-to-point integration challenges. Oracle Integration Cloud provides native connectivity to Oracle Software as a Service (SaaS) applications, such as Oracle Engagement Cloud, Oracle B2C Service, and so on. Oracle also provides an extensive library of adapters to Oracle and 3rd party SaaS and on-premises applications to deliver new business services faster.

The adapters provide out-of-box integration capability to the required application, easier access to various business events within the application and better manageability across the integration ecosystem of the Organization.

The Oracle B2C Service Adapter is one of the key adapters which customers can leverage from the extensive portfolio of ICS adapters. Currently, The Oracle B2C Service adapter builds on the Connect Web Services API for SOAP to provide real-time integration with the Oracle B2C Service Platform.

Oracle B2C Service adapters provide the following benefits when integrating Oracle B2C Service with other business applications.

- Integrates easily with the OOracle B2C Servicenatively and enables creation of reusable service assets.
- Generates automatic mapping to the exposed business object or event subscription.
- Supports the RightNow Object Query Language (ROQL) to query metadata information.
- Supports custom attributes in business objects.
- Automatically handles security policy details required to connect to the Oracle B2C Service applications.
- Provides standard error handling and integration analytical capabilities.

– Enables CRUD (create, get, update, and destroy) operations against business objects in the Oracle B2C Service applications.

Additional information is available in the integration guide: Integrating with Oracle B2C Service

CONCLUSION

This integration guide outlined various Integration APIs and approaches to integrate with the Oracle B2C Service Platform.

For more help or if you need additional assistance with these best practices, contact your Oracle account team or Oracle B2C Service Customer Care for recommendations on potential next steps.

Additionally, you may find on several useful resources that are available to learning more about the B2C Service products and features, API guide and access to the B2C Service Community listed below:

- Product Documentation and Tutorials
- API Documentation
- Oracle B2C Service Community
- Agent Browser UI Extensibility Framework Quick Start Guide

APPENDIX

Oracle B2C Service offers several shared services that include APIs not covered in this brief. These can be accessed from the Technical Documentation:

Customer Portal Framework is the web application framework and out-of-the-box web page set for managing and creating web experiences for your customers on any web browser. This includes personal computers, smartphones, simple browsers, and even custom embedded web browsers within devices. It provides a modern model-view-controller architecture with open standards along with prebuilt themes, templates, pages, and widgets that can be rearranged and branded to meet your requirements.

Connect Knowledge API (Knowledge API) is a backward-compatible, public API that enables customers and partners to leverage the Oracle B2C Service knowledge base from any external application, device, or service where knowledge is needed. The API can also be used when customizing a web self-service experience through the Customer Portal framework. It gives developers the ability to search for content in the knowledge base, rate content, obtain Smart Assistant recommendations, and even obtain the most popular answers in the knowledge base.

Knowledge Advanced REST API provides multiple public REST APIs that you can use to access data stored in Oracle B2C Service and construct integrations to other systems. You can use these REST APIs to interact, create, and edit knowledge articles; search for knowledge articles; make recommendations; and view user information from external applications.

Intelligent Advisor Determinations API is a suite of web service interfaces that can be activated for any deployed policy model. These web services provide high performance operations for interacting with the rules, interview and data mapping within any deployed policy model.

Chat APIs provide application developers with a set of web services that can be used to securely interact with Oracle B2C Service Chat.

Oracle Field Service provides Oracle REST APIs to view and manage data stored in Oracle Field Service. It also allows customers to create subscription for one or more Oracle Field Service events. Events are generated when there is a change in the application, for example, activity created, activity completed, inventory installed, and so on.

ORACLE CORPORATION

Worldwide Headquarters
500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries
TELE + 1.650.506.7000 + 1.800.ORACLE1
FAX + 1.650.506.7200
oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com. Outside North America, find your local office at oracle.com/contact.

 blogs.oracle.com/oracle  facebook.com/oracle  twitter.com/oracle

Integrated Cloud Applications & Platform Services

Oracle B2C Service Integration and Extensibility Frameworks
June 2019
Author: Shiv Tenneti
Contributing Authors: Prashanth Rao

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