



# Oracle CPQ Cloud

## What's New in 2015 R2

December 2015

# TABLE OF CONTENTS

- REVISION HISTORY .....4**
- OVERVIEW .....5**
- RELEASE FEATURE SUMMARY .....6**
- EASY ADMINISTRATION .....7**
  - Single Select Pick Lists in Configuration ..... 7*
    - Resource Admin ..... 7
    - External Image Support ..... 10
    - Filters ..... 11
    - Pick Maps ..... 14
    - System User for Authentication ..... 17
    - Steps to Enable ..... 17
    - Tips and Considerations ..... 17
    - Key Resources ..... 18
  - Dynamic BMQL Variables ..... 18*
    - Syntax ..... 18
    - Dynamic Variables in the WHERE Clause ..... 19
    - Exception Handling ..... 19
    - Example: Changing a Query Based on User Inputs ..... 20
    - The WHERE Clause as a Variable ..... 21
    - Making Everything Dynamic ..... 21
    - Steps to Enable ..... 21
    - Tips and Considerations ..... 21
  - Document Designer Enhancements ..... 22*
    - Search in Document Designer ..... 22
    - Table Column Width in Document Designer ..... 25
    - Microsoft Word Heading Styles ..... 26
    - Document Generator REST API ..... 27
    - Steps to Enable ..... 30
    - Key Resources ..... 30
  - Email Designer Enhancements ..... 30*
    - Email Designer Templates for Submit Sub-Action Notifications ..... 31
    - Search in Email Designer ..... 31
    - Table Column Width in Email Designer ..... 34
    - Email Generator REST API ..... 35
    - Steps to Enable ..... 38
  - Access Data Table REST API ..... 38*
    - Access Data Table REST API Metadata ..... 39
    - Steps to Enable ..... 40
    - Key Resources ..... 40
- ENTERPRISE EXCELLENCE .....41**
  - Secure Attributes ..... 41*
    - Transferring Values from Memory ..... 42
    - Tips and Considerations ..... 43
    - Functional Known Issues ..... 43
    - Steps to Enable ..... 44
  - Part Administration Enhancements ..... 44*
    - Part Display Number ..... 44
    - Part Visibility Using Company Associations ..... 45
    - Quick Keys ..... 46
    - Steps to Enable ..... 47
    - Tips and Considerations ..... 47

<i>Prevent Posting of Read-Only Fields</i> .....	47
Steps to Enable .....	47
<b>INTEGRATION .....</b>	<b>48</b>
<i>Salesforce Integration Enhancements</i> .....	48
Integration with Multiple CPQ Cloud Sites .....	48
Integration with Multiple CPQ Cloud Commerce Processes .....	49
Steps to Enable .....	49
Tips and Considerations.....	49
Key Resources .....	49
<i>Oracle Commerce On Premise Integration Enhancements</i> .....	49
Self-Service Configuration and Checkout for Complex Products .....	50
Customer-Specific Pricing .....	50
Custom Product Family Integrations.....	50
Steps to Enable .....	53
Tips and Considerations.....	53
Key Resources .....	53
<i>Decodebase64 BML Function</i> .....	54
Steps to Enable .....	54
<i>Web Services v1.0 Access Modifications</i> .....	54
Steps to Enable .....	54
<b>PRE-UPGRADE CONSIDERATIONS .....</b>	<b>55</b>
<i>Known Functionality</i> .....	55
Configuration Deployment Failures .....	55
Email Template Editor.....	55
Secure URL Addresses.....	55
Translation .....	55
Migration .....	55
<i>Resolved Known Issues</i> .....	55
<i>Functional Known Issues</i> .....	56
<i>Translation Status</i> .....	56
<b>POST-UPGRADE CONSIDERATIONS .....</b>	<b>57</b>
<i>Browser Support</i> .....	57
Supported Browsers .....	57
Certified Browsers .....	58
<i>Salesforce Managed Package Support</i> .....	58
<i>TLS Support for Salesforce Integrations</i> .....	58
<i>Training</i> .....	59
<i>Additional Information</i> .....	59
<i>Disclaimer</i> .....	59

## REVISION HISTORY

This document will continue to evolve as existing sections change and new information is added. All updates are logged below, with the most recent updates at the top.

Date	What's Changed	Notes
18 DEC 2015		Initial Document Creation
26 JAN 2016	<a href="#">Comparison Value Syntax</a>	The second filter example was changed to correctly use the <code>\$or</code> MongoDB operator. Additionally, a filter example that used the <code>concat</code> MongoDB operator has been removed because CPQ Cloud does not support the <code>concat</code> MongoDB operator in filters or queries.

## OVERVIEW

This guide outlines the information you need to know about new or improved functionality in Oracle CPQ Cloud 2015 Release 2 (2015 R2). Each section includes a brief description of the feature, the steps you need to take to enable or begin using the feature, any tips or considerations that you should keep in mind, and the resources available to help you.

### GIVE US FEEDBACK

We welcome your comments and suggestions to help us improve this document. Please send your feedback to [cpqcloud\\_documentation\\_us\\_grp@oracle.com](mailto:cpqcloud_documentation_us_grp@oracle.com).

## RELEASE FEATURE SUMMARY

Some of the new CPQ Cloud 2015 Release 2 features are automatically available to users after the upgrade and some require action from the company administrator or Oracle.

The table below offers a quick view of the actions required to enable each of the features.

Feature	Action Required to Enable Feature			
	Automatically Available	End User Action Required	Administrator Action Required	Oracle Service Request Required
<b>Easy Administration</b>				
<a href="#">Single Select Pick Lists in Configuration</a>	✓		✓	
<a href="#">Dynamic BMQL Variables</a>	✓		✓	
<a href="#">Document Designer Enhancements</a>	✓		✓	
<a href="#">Email Designer Enhancements</a>	✓		✓	
<a href="#">Access Data Table REST API</a>	✓		✓	
<b>Enterprise Excellence</b>				
<a href="#">Secure Attributes</a>	✓		✓	
<a href="#">Part Administration Enhancements</a>	✓		✓	
<a href="#">Prevent Posting of Read-Only Fields</a>				✓
<b>Integration</b>				
<a href="#">Salesforce Integration Enhancements</a>			✓	✓
<a href="#">Oracle Commerce On Premise Integration Enhancements</a>			✓	
<a href="#">Decodebase64 BML Function</a>	✓		✓	
<a href="#">Web Services v1.0 Access Modifications</a>	✓		✓	
<b>Upgrade Considerations</b>				
<a href="#">Pre-Upgrade Considerations</a>			✓	
<a href="#">Post-Upgrade Considerations</a>			✓	

## EASY ADMINISTRATION

Administrators are the main force behind keeping a CPQ Cloud site up-to-date. A simple and intuitive administration experience—for a robust and powerful product—makes it easy for customers to stay ahead of the curve.

New features to make administration easier include:

- Single Select Pick Lists in Configuration
- Dynamic BMQL Variables
- Document Designer Enhancements
- Email Designer Enhancements
- Access Data Table REST API

---

### SINGLE SELECT PICK LISTS IN CONFIGURATION

A new attribute type, Single Select Pick List, can be created in Configuration for Text, Integer, and Float data types. While similar to a Single Select Menu attribute on the user side, a Single Select Pick List attribute's menu options are managed within a defined resource.

**Note:** In this release, the only resources that can be referenced by a Single Select Pick List are CPQ Cloud Data Tables.

Single Select Pick Lists have several advantages:

- **Resource Admin:** Simple menu option maintenance through Data Tables
- **External Image Support:** Image Menus can reference external images
- **Filters:** Dynamic menu option filtering based on static values, other Configuration attribute values, and user input
- **Pick Maps:** Force set Configuration attribute values to Data Table values without creating a Recommendation rule or writing BMQL

**Note:** Single Select Pick Lists can be created as stand-alone attributes or as array-type attributes.

---

### RESOURCE ADMIN

When menu options for a Single Select Pick List attribute need to be added, modified, or removed, the admin simply needs to modify the Data Table from which the menu options are being retrieved. The changes will be live on the user side as soon as the Data Table is deployed.

When creating a Configuration attribute, if **Text**, **Float**, or **Integer** is selected for **Data Type**, a **Single Select Pick List** option is available for **Attribute Type**. Selecting **Single Select Pick List** will reveal the **Single Select Pick List Panel**.

Attribute Editor Product Family : DPL Demo

---

**Main Information**

\*Name:

\*Variable Name:

Array Type:

\*Data Type:  Text  Float  Integer  HTML  Date  Currency  Boolean

Attribute Type:  Text Field  Single Select Menu  Multi Select Menu  Single Select Pick List

---

**Single Select Pick List**

Domain:  \*Variable:  \*Display:

Filter:

Pick Maps: Picker Attribute:  Model Attribute:

---

[Back to Top](#)

The **Domain**, **Variable**, and **Display** drop-down menus determine the menu options of the Single Select Pick List on the user side.

Field	Description
<b>Domain</b>	<p>The resource (always a Data Table in 2015 R2) where the Single Select Pick List's menu options are defined.</p> <p><b>Note:</b> In the Domain drop-down menu, Data Table names are prefixed with <code>custom</code> and will always have their first letter capitalized. For example, a Data Table named <code>locationOptions</code> would be listed in the Domain drop-down menu as <code>customLocationOptions</code>.</p>
<b>Variable</b>	<p>The column in the Domain that contains the variable names of all possible menu options for the Single Select Pick List. As in Single Select Menu and Multi-Select Menu attributes, the variable name of a menu option that is selected will be returned to the system to be used in Configuration rules.</p> <p><b>Note:</b> The Variable menu contains a list of the selected Domain's column names that are the same type (Text, Integer, or Float) as the Single Select Pick List attribute.</p>
<b>Display</b>	<p>The column in the Domain that contains display names for all possible menu options for the Single Select Pick List. As in Single Select Menu and Multi Select Menu attributes, display names will determine the name of the menu option that is shown on the user side.</p> <p>A display name can be different from or the same as the variable name of a menu option. In Single Select Pick Lists, if all possible menu options have the exact same display name and variable name, the Display column can be the same as the Variable column.</p> <p><b>Note:</b> Display is only available for Single Select Pick Lists with a Data Type of Text.</p>



For example, a Single Select Pick List could be implemented so that a user could select a Country. A Data Table would need to be created with all possible menu options, as in Image 1.



Image 1

Within the Single Select Pick List, the admin can select the Data Table (Domain) that contains the menu options, the Data Table column that contains menu option variable names (Variable), and the Data Table column that contains menu option display names (Display), as in Image 2.

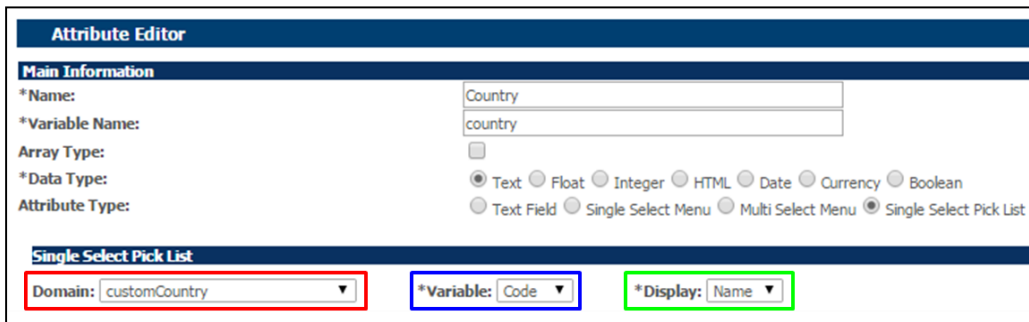


Image 2

On the user side (shown in Image 3), the display names for each variable name are shown in the Single Select Pick List. If **United States** was selected, “USA” would be returned to the system for use in Configuration rules since “United States” is the display name for the “USA” menu option variable name, as defined in the Country Data Table in Image 1.



Image 3

In this example, if new countries needed to be added to the Single Select Pick List as menu options, the admin could simply add a row for each new country menu option to the Data Table referenced by the Single Select Pick List attribute.

## EXTERNAL IMAGE SUPPORT

Using a Single Select Pick List attribute, admins can now create a single select Image Menu that references external images.

**Note:** Single Select Pick Lists that are Image Menus can also reference File Manager images. Single Select Menu and Multi-Select Menu attributes that are Image Menus still reference images from the File Manager and cannot reference external images.

As with Single Select Menu and Multi-Select Menu attributes, after a Single Select Pick List attribute is created, the admin can select **Yes** for **Image Menu** to make the attribute an Image Menu. When **Yes** is selected for **Image Menu**, an **Image** drop-down appears in the **Single Select Pick List** panel.



Single Select Pick List

Domain: customCountry \*Variable: Code \*Display: Name

\*Image: Image

The **Image** drop-down lists all string-type columns in the selected Domain. The column selected should contain links to images that will be shown along with the display names of corresponding menu options on the user side. Within the Domain column, links to external images must be full URLs. Links to File Manager images can be added via their full URLs or their File Manager BASE\_PATH/SECURE\_PATH URLs.

For example, a column could be added to the Country Data Table that contained links to images for each Country menu option, as in Image 4.



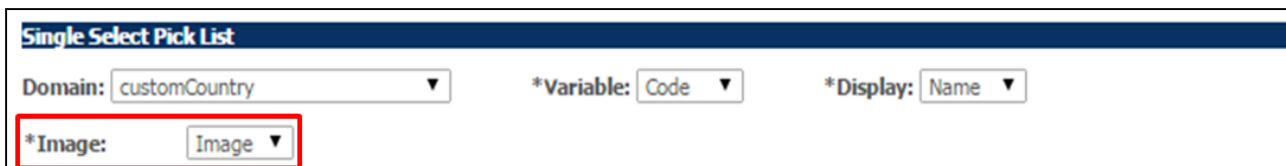
Filter: Country

Page Length: 50

#	Code	Name	Image
1	Canada	Canada	<a href="http://externalimage.com/canada">http://externalimage.com/canada</a>
2	Mexico	Mexico	<a href="http://externalimage.com/mexico">http://externalimage.com/mexico</a>
3	USA	United States	<a href="http://externalimage.com/usa">http://externalimage.com/usa</a>

Image 4

Within the Single Select Pick List, the admin can select the **Image** column in the **Image** drop-down menu, as in Image 5.



Single Select Pick List

Domain: customCountry \*Variable: Code \*Display: Name

\*Image: Image

Image 5

On the user side (shown in Image 6), images are pulled from the **Image** column in the Domain and shown to the user along with the corresponding display names of the menu options.



Image 6

**Note:** As in Single Select Menu and Multi-Select Menu image Menus, the **Image Menu Properties** panel can be used to modify the display options of Single Select Pick List Image Menus.

Single Select Pick Lists must be drop-down menus and cannot be grids like Single Select Menu and Multi-Select Menu attributes.

## FILTERS

Single Select Pick Lists can filter the menu options that are shown to the user based on static values or other Configuration attribute values. Menu option filtering is configured in the **Filter** field within the Single Select Pick List panel.

Within the **Filter** field, the admin can define a query to filter which menu options in the Domain are shown to the user.

## FILTER SYNTAX

The **Filter** field accepts a subset of MongoDB syntax. For more information on what MongoDB operators CPQ Cloud can recognize, log in to CPQ Cloud as a FullAccess user and see the “Query Specification Syntax” topic in the Online Help. For more information on MongoDB syntax, see [MongoDB’s Query Documents documentation](#).

The first value in the Filter statement must contain the name of the column in the Domain that will be compared against the second value in the Filter statement. The second value in the Filter statement must contain the value that the column in the Domain will be compared against. The data type of the

column in the Domain will be the data type that the system assumes for the value on the right side of the filter.

```
{Column:ComparisonValue}
```

Menu options in the Domain will only be shown on the user side if the column in the Domain and the value it is compared against are equal.

**Note:** You cannot use a non-array Single Select Pick List within an array Single Select Pick List's Filter. Within an array Single Select Pick List's Filter, you can reference:

- Non-array-type attributes
- Array-type attributes that have the same Array Control attribute as the Single Select Pick List

## COLUMN SYNTAX

The column in the Domain must be referenced by its name.

*Syntax:* No formatting required

*Example:* "CountryCode"

## COMPARISON VALUE SYNTAX

The value can contain:

- **Literal string values** (can include numbers, but it will still be treated as a string)  
*Syntax:* Within quotation marks  
*Example:* "USA"
- **Numeric values** (can be integers or floats)  
*Syntax:* No formatting required  
*Example:* 4.25
- **Configuration attribute values**  
*Syntax:* Attribute variable name within quotation marks and pipes  
*Example:* "|stateUSA|"

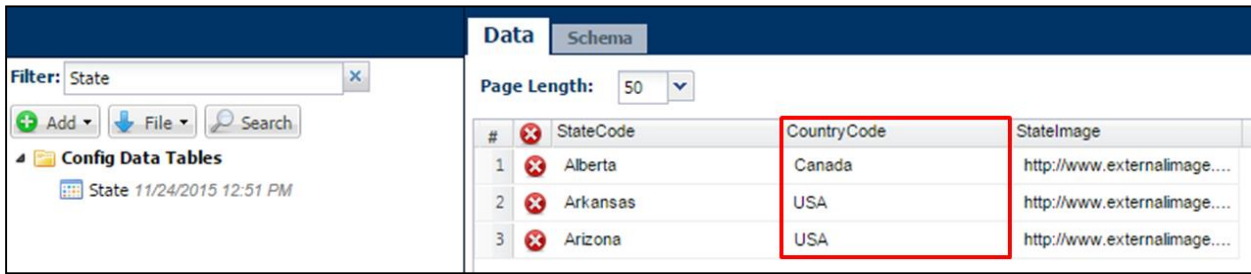
Filter examples:

- {CountryCode:"|country|"}  
Menu options in the Domain will be shown in the Single Select Pick List if the value in the CountryCode column matches the value of the country attribute.
  - {\$or:[{StateProvinceCode:"|state|"},{StateProvinceCode:"|province|"}]}
- Menu options in the Domain will be shown in the Single Select Pick List if the value in the StateProvinceCode column matches the value of the state attribute or the province attribute.

## FILTER USE CASE

For example, a State Single Select Pick List attribute could be filtered to only show State menu options that are within the value of the Country attribute. In this example, within the Domain that the State


attribute draws its menu options from, there should be a column with Country values for each State menu option, as in Image 7.



#	StateCode	CountryCode	StateImage
1	Alberta	Canada	http://www.externalimage....
2	Arkansas	USA	http://www.externalimage....
3	Arizona	USA	http://www.externalimage....

Image 7

Within the Single Select Pick List, the admin can define a Filter to compare the CountryCode column in the Domain to the value of the Country attribute, as in Image 8.



Single Select Pick List

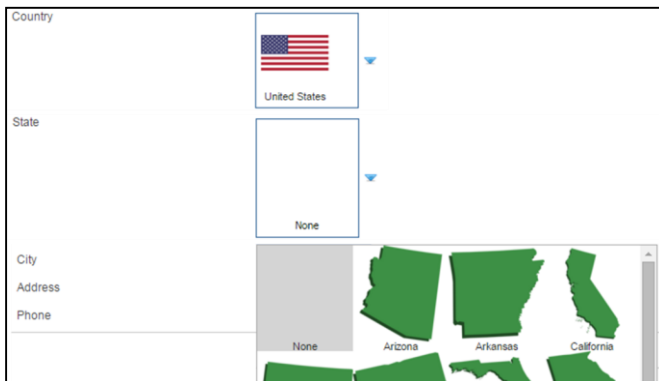
Domain: customState \*Variable: StateCode \*Display: StateCode

\*Image: StateImage

Filter: {CountryCode: "[country]"}

Image 8

On the user side (shown in Image 9), only menu entries whose CountryCode value matches the value of the Country attribute are shown.



Country: United States

State: None

City: None

Address: None

Phone: None

Map: Arizona, Arkansas, California

Image 9

## TYPE TO FILTER

Unlike Single Select Menu and Multi-Select Menu attributes, Single Select Pick Lists have a field on the user side in which the user can type. In addition to Filters that may be defined within Single Select Pick Lists, Text-type Single Select Pick Lists that are not Image Menus respond to user input in the field on the user side and filter out options that do not contain the character(s) entered by the user. This is known as “type to filter” functionality.

**Notes:**

- Type to filter functionality is case-sensitive.
- Type to filter functionality is automatically enabled on all Text-type Single Select Pick Lists that are not Image Menus.
- Type to filter cannot be turned off.
- Single Select Menu and Multi-Select Menu attributes do not have a field on the user side that accepts user input, so type to filter functionality does not exist with these attribute types.

For example, in Image 10, since there is no user input in the City field, all cities are shown in the drop-down menu. In Image 11, since the user entered a capital C in the City field, only cities that contain a capital C are shown in the drop-down menu.



Image 10

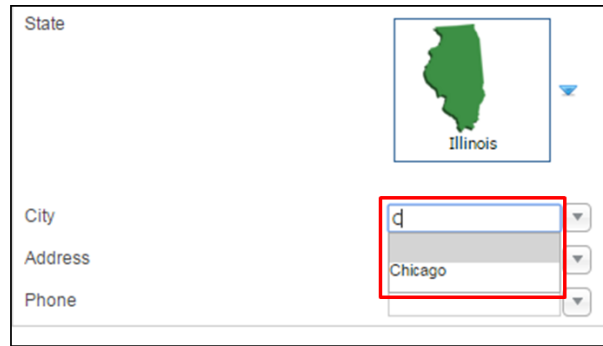


Image 11

**Note:** Integer and Float Single Select Pick Lists do not filter based on user input. Text Single Select Pick Lists that are also Image Menus do not filter based on user input.

PICK MAPS

Pick Maps allow admins to set one or more Configuration attribute values to a value defined within a column of the Single Select Pick List’s Domain, without creating any Recommendation rules that would require BMQL.

The **Picker Attribute** and **Model Attribute** menus determine Pick Map relationships. When a value is given to a Single Select Pick List, the Model Attribute is force-set to the value of the Picker Attribute value in the same row as the selected Single Select Pick List value.

Field	Description
Picker Attribute	<p>The column in the Domain that contains the value that will be force set on a Configuration Attribute. The row with the menu option that is selected in the Single Select Pick List on the user side will be used to determine the Picker Attribute value that contains the force set value.</p> <p><b>Note:</b> The Picker Attribute menu lists all of the selected Domain's columns.</p>

## Model Attribute

The Configuration attribute that will be force set with the Picker Attribute value.

### Notes:

- The Model Attribute menu will only contain Configuration attributes that are on the same level or a higher level of the Product Hierarchy as the Single Select Pick List attribute, and whose data type matches the data type of the Picker Attribute column in the Domain.
- Only one Pick Map can be created per Model Attribute (Configuration attribute).
- In a non-array type Single Select Pick List, you cannot select array-type attributes as Model Attributes in Pick Maps.
- In an array-type Single Select Pick List, you can only select array-type attributes that share the same Array Control attribute with the Single Select Pick List as Model Attributes in Pick Maps.

**Important:** In a Single Select Pick List attribute that will contain Picks Maps, the Variable column in the Domain should not contain duplicate values. In this scenario, the system will use the Picker Attribute column value within the same row as the selected menu option in the Variable column. If there are duplicate values in the Variable column, the user could select either identical menu option, and there is no way to guarantee that the intended Picker Attribute value will be used.

When a Pick Map relationship has been established using the **Picker Attribute** and **Model Attribute** menus, the Pick Map can be added to the Single Select Pick List by clicking the **Add** button.

Pick Maps: Picker Attribute: PhoneCode Model Attribute: Phone Add

Delete

Active Pick Maps are listed in the **Pick Maps** list.

Pick Maps: Picker Attribute: PhoneCode Model Attribute: Phone Add

PhoneCode -> Phone

Delete

An active Pick Map can be removed from the **Pick Maps** list (and therefore be deactivated) by selecting the Pick Map within the list and then clicking the **Delete** button.

Pick Maps: Picker Attribute: PhoneCode Model Attribute: Phone Add

PhoneCode -> Phone

Delete

For example, an Address Single Select Pick List could contain a Pick Map to set the value of a Phone attribute based on the value of the Address attribute.

In this example, within the Domain that the Address attribute draws its menu options from, there should be a column with Phone values for each Address menu option, as in Image 12.

#	StateCode	CityCode	AddressCode	PhoneCode
1	Arkansas	Bentonville	2501 Technology Circle Suite 5 Bentonville, AR 72712	
2	Arizona	Phoenix	2355 East Camelback Road Suite 950 Phoenix, AZ 85016	+1.602.333.9000
3	Arizona	Tucson	9070 South Rita Rd. Suite 1950 Tucson, AZ 85747	
4	California	Arcadia	440 East Huntington Drive Suite 400 Arcadia, CA 91006	+1.626.463.6685

Image 12

Within the Single Select Pick List, the admin can define a Pick Map to set the value of the Phone attribute to the PhoneCode value in the same row as the selected Address menu option of the Single Select Pick List on the user side, as in Image 13.

Single Select Pick List

Domain: customAddress \*Variable: AddressCode \*Display: AddressCode

Filter: {CityCode:"|city|"}

Pick Maps: Picker Attribute: PhoneCode Model Attribute: Phone Add

PhoneCode -> Phone

Delete

Image 13

On the user side (shown in Images 14 and 15), the Phone attribute is force set based on the value of the Address attribute.

City: Phoenix

Address: 2355 East Camelback R

Phone: +1.602.333.9000

Image 14

City: Arcadia

Address: 440 East Huntington Dr

Phone: +1.626.463.6685

Image 15

**Note:** Attributes are force set by Pick Maps using AJAX. As with all Configuration attributes affected by AJAX, when the attribute is force set, all Configuration rules that use the attribute in their conditions will be invoked.



## SYSTEM USER FOR AUTHENTICATION

---

A new system user, **system\_user\_cpq\_server\_as\_client**, has been added to the User Administration List of the Host Company.

<input type="checkbox"/>	<u>superuser</u>	Super User	FullAccess
<input type="checkbox"/>	<b>system_user_cpq_server_as_client</b>	FirstName	FullAccess
<input type="checkbox"/>	<u>tandrews</u>	Tim	RestrictedAccess

The **system\_user\_cpq\_server\_as\_client** user is used for OAuth authentication within CPQ Cloud to enable Single Select Pick List functionality. The **system\_user\_cpq\_server\_as\_client** user cannot be inactivated or modified—errors will occur when an admin attempts to inactivate the user and when an admin attempts to save changes made to the user.

## STEPS TO ENABLE

---

Single Select Pick Lists are automatically available on all 2015 R2 sites.

## TIPS AND CONSIDERATIONS

---

- For best performance, ensure that Filters on large Data Tables have suitable indexes defined. For more information on indexes, log in to CPQ Cloud as a FullAccess User and see the [Data Table Overview](#) topic in the Online Help.
- If a Single Select Pick List is on a desktop layout that has a corresponding mobile layout, the mobile layout will not be available on the user side and the desktop layout will always be used. Additionally, if a Single Select Pick List is on a desktop layout that does not have a corresponding mobile layout, a mobile layout cannot be created. The mobile layout will become available and new mobile layouts can be created as soon as all Single Select Pick List attributes are removed from the desktop layout.

Mobile layouts will still be available on the user side for Configuration Flow nodes that do not have a Single Select Pick List on the desktop layout. For example, if a Start Node has a Single Select Pick List on the desktop layout but the End Node in the sequence does not have a Single Select Pick List on the desktop layout, the desktop layout will be used for the Start Node but the mobile layout will be used for mobile users on the End Node.

- After a site restart, the first time a user enters a Configuration Flow on the user side that contains one or more Single Select Pick List attributes, the Single Select Pick List(s) will take longer to load than the rest of the page as the JavaScript cache is loaded. The rest of the page will function as expected, and the Single Select Pick List attributes will function as expected once they load. All subsequent times that a user enters a Configuration Flow on the user side that contains one or more Single Select Pick List attributes, Single Select Pick List attributes will load normally.
- Filters are not required. Single Select Pick Lists can be created without Filters.
- Pick Maps are not required. Single Select Pick Lists can be created without Pick Maps.
- If a Single Select Pick List array-type attribute has a Filter and/or a Pick Map that references other array-type attributes with the same array control attribute, neither the Single Select Pick List array-type attribute nor the array-type attributes being filtered-on or pick-mapped can be removed from an array set with the same array control attribute (until the Filter and/or Pick Map references are removed from the Single Select Pick List).

- If a Single Select Pick List array-type attribute has a Filter and/or a Pick Map that references other array-type attributes with the same array control attribute, neither the Single Select Pick List array-type attribute nor the array-type attributes being filtered-on or pick-mapped can have their array control attribute changed (until the Filter and/or Pick Map references are removed from the Single Select Pick List).
- If an array set contains a Single Select Pick List array-type attribute that has a Filter and/or a Pick Map that references other array-type attributes with the same array control attribute, the array set cannot be deleted (until the Filter and/or Pick Map references are removed from the Single Select Pick List).

---

## KEY RESOURCES

[MongoDB's Query Documents](#)

---

## DYNAMIC BMQL VARIABLES

CPQ Cloud's query language, BMQL, now allows the use of dynamic variables for column names, Data Table names, and WHERE clauses.

Using direct variable substitution, called *dynamic variables*, makes it simpler to write BML queries that change based on user input, without having to write a unique clause in the query for every possible permutation of inputs. As a result, scripts using this new feature are much more scalable and succinct.

When a BMQL call contains dynamic variables, the data types for the variables are validated and the dynamic variables are replaced with the input values at runtime. For more information on validation, see the section [Validation](#) below. Then BMQL returns a full SQL string that is executed.

---

## SYNTAX

To support dynamic variables in BMQL, a new optional parameter, `fieldMap`, is available. There are now two optional parameters: `contextOverride` and `fieldMap`.

The BMQL method now supports the following three signatures:

- `bmql(QueryString)`
- `bmql(QueryString, contextOverride)`
- `bmql(QueryString, contextOverride, fieldMap)`

When using `contextOverride` to specify a certain language and `fieldMap` to use variables inside a variable WHERE clause, the syntax for the entire call is:

```
bmql(QueryString, contextOverride, fieldMap);
bmql("select columnName from tableName WHERE $where", lang, fields);
```

- `contextOverride` (existing optional parameter)

For more information on this parameter, log in to CPQ Cloud as a FullAccess User and see the [BigMachines Query Language](#) topic in the Online Help.

- `fieldMap` (new optional parameter)

The fieldMap parameter is a string dictionary. It is used when the WHERE clause has been completely substituted with a string variable, and there are also variables within the WHERE clause.

In this case, each variable in the dynamic WHERE clause must be passed into the fieldMap dictionary and referenced by its key in the WHERE clause. All variables must be passed as string types, regardless of their data type in the Data Table.

**Example:**

```
lang = dict('string');
fields = dict("string");
x_var = "6.08";
put(fields, "$field1", x_var);
where = "float1 = $field1";
results = bmql("select columnName from tableName where $where",
lang, fields);
```

This parameter is optional.

**Note:** If the third parameter is used, the second parameter must also be defined.

## DYNAMIC VARIABLES IN THE WHERE CLAUSE

---

If variable substitution is needed within the WHERE clause, use the method signature which passes in the values of each of the variables.

These variables must be defined in a string Dictionary and passed in as the third parameter. In this case, a second parameter must also be defined.

For example, if the \$where variable used the declared variables x\_var and y\_var, the variables would need to be added to a dictionary prior to the BMQL call.

**Example:**

```
table = `dataTable`; //dataTable is the name of an existing
data table
columns = `columnName`;
fields = dict("string");
put(fields, "$field1", x_var);
put(fields, "$field2", y_var);
where = "x = $field1 AND y = $field2";
results = bmql("SELECT $columns FROM $table WHERE $where", lang,
fields);
```

## OPERATORS

For complete information about operators, log in to CPQ Cloud as a FullAccess User and see the [BigMachines Query Language](#) topic in the Online Help.

## EXCEPTION HANDLING

---

There are two exception handling situations.

- Variables are used in the \$where clause and no field dictionary is passed in. This BML will validate, but will throw an error when run.
- No new variables are used, but a field dictionary is passed in. This BML will not throw any errors, and will work. Optional parameters are ignored if they are not needed.

#### EXAMPLE: CHANGING A QUERY BASED ON USER INPUTS

---

A customer stores data in different Data Tables for pricing in different regions. A function sets the value of the variable `table` to the returned value of a Util Library function named `passed_string1`. Currently that code looks like this:

```

bmqlReturn = "nothing";
table = util.passed_string1();

if(table == "sorting_rename_1") {
    results = bmql("select str from sorting_rename_1 where str =
'a'");
    for result in results {
        bmqlReturn = get(result, "str");
    }
}
elif(table == "sorting_rename_2") {
    results = bmql("select str from sorting_rename_1 where str =
'a'");
    for result in results {
        bmqlReturn = get(result, "str");
    }
}
elif(table == "sorting_rename_3") {
    results = bmql("select str from sorting_rename_1 where str =
'a'");
    for result in results {
        bmqlReturn = get(result, "str");
    }
}
else {
    bmqlReturn = "not a valid table";
}
return bmqlReturn;

```

With dynamic variables, the admin can write something like this instead:

```

bmqlReturn = "nothing";
table = util.passed_string1();

results = bmql("select str from $table where str = 'a'");
for result in results {
    bmqlReturn = get(result, "str");
}

return bmqlReturn;

```

Using dynamic variables, the code is much simpler and much more scalable.

## THE WHERE CLAUSE AS A VARIABLE

---

The entire WHERE clause can be a string variable.

## MAKING EVERYTHING DYNAMIC

---

In this example, everything that can be dynamic is dynamic.

```
bmqlReturn = "nothing";
select = "string1";
from = "uploadXMLtable";
lang = dict("string");
fields = dict("string");
x_var = "6.08";
put(fields, "$field1", x_var);
where = "float1 = $field1";
results = bmql("select $select from $from where $where", lang, fields)
for result in results {
    bmqlReturn = get(result, "string1")
}
return bmqlReturn;
```

Use the link in the BMQL Function Wizard to see Data Table names and associated column details.

## STEPS TO ENABLE

---

Dynamic BMQL variables are automatically available on all 2015 R2 sites.

## TIPS AND CONSIDERATIONS

---

There are several issues to consider when using dynamic variables in BMQL queries.

### VALIDATION

Most of the validation in BMQL that uses dynamic variables will occur when the code is executed. Previously, validation occurred when the code was checked or saved.

This change is required because variables aren't replaced with their values until runtime. However, syntax will still be checked when code is saved to confirm that variables are in the right places and the standard keywords are still there.

Therefore, when using variables for columns, Data Tables, or the entire WHERE clause, more rigorous testing of the BMQL should be done and potential errors should be handled by using the `hasError(returnString)` and `getMessage(returnString)` BML functions.

String literal BMQL calls are not affected by these changes and will continue to perform a full validation when checked and saved.

### SQL INJECTION

Continue to follow existing best practices regarding SQL injection. Do not allow any user-generated data to be used directly in a BMQL statement.

In a fully dynamic where clause, you must put variables into the fields array. These variables in the fields array will have SQL characters escaped.

For example:

```
fields = dict("string");
dict.put(fields, "$scal", commerceAttribute1);
where = "field1 = $scal AND field2 = 'someValue'";
results = bmql("SELECT col1 FROM table WHERE $where", lang, fields);
```

**Important:** Each customer is individually responsible for writing and testing their own dynamic BMQL calls to ensure that they are safeguarded from potential SQL injections.

---

## DOCUMENT DESIGNER ENHANCEMENTS

The Document Designer, a drag-and-drop document template-creation and administration tool, was introduced in 2015 R1. As part of continuing improvements to Document Designer, several new features have been added in this release.

- Search for static text or dynamic attributes within a Document Designer template using a new Search field.

Many admins use browser search functionality to locate page Elements. However, this may skip over Elements that are hidden in the Document Designer. New Document Designer search functionality will search Elements in Layouts that are collapsed in the editor.

- In 2015 R1, the Document Designer allowed administrators to define Table column widths in pixels. In this release, administrators can also define Table column widths in inches, millimeters, centimeters, or as a percentage of the page width.
- Use Heading styles that leverage existing heading definitions in Microsoft Word. If a user generates a DOCX document that used these heading definitions, the resulting file will contain native Microsoft Word headings and a table of contents that will respond to the user's updates.
- A new REST API makes the retrieval of a Document Designer output document available to external systems. The API call must contain simple parameters, including the identifier of an existing Transaction.

---

## SEARCH IN DOCUMENT DESIGNER

Admins can now use the new **Search** field, in the **Properties** Panel of the Document Designer editor, to search for characters, words, phrases, and so on in a Document Designer template.



In addition to what can be found by a browser search, Document Designer search functionality will also search Elements in Layouts that are collapsed in the Template Flow. With this functionality, the entire template is always searched, even if some parts of it are not visible to the user.

The following table describes which Elements and template components can be searched:

Document Designer Element/Component	Plain Text Search	Match Case	Attribute Search
Text*	✓	✓	✓
Heading*	✓	✓	✓
Table of Contents Title	✓	✓	✓
Inline XSL Snippet*	✓	✓	
Global XSL Snippet	✓	✓	
Embed Document (displayed file path only)	✓	✓	

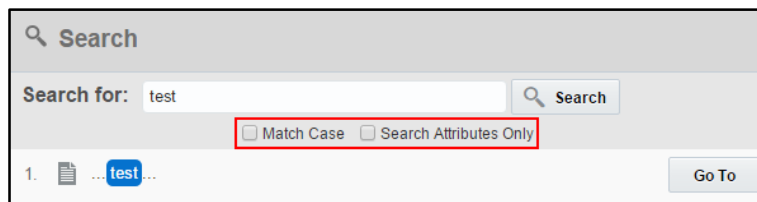
\*Applies to Elements both inside and outside of Table cells

## SEARCH TYPES

- Plain Text**  
 Type of search executed when *Search Attributes Only* checkbox is unselected (default). Search term is compared against the text content of textual Elements, the XSL content of the Global XSL Snippet and Inline XSL Snippet Elements, and the displayed file path of Embed Document Elements. Attribute labels in textual Elements are considered part of the normal flow of text.
- Attribute**  
 Type of search executed when *Search Attributes Only* checkbox is selected. Search term is compared against only the variable names and labels of Attributes contained within textual Elements.
- Match Case**  
 When *Match Case* checkbox is selected (unselected is default), case sensitivity is applied to both *Plain Text* and *Attribute* searches.

Type text into the **Search** field and either press **Enter** or click the magnifying glass to open the **Search** dialog box, shown below. If you opened the dialog box without entering text into the **Search** field first, enter content in the **Search for:** field and either press **Enter** or click **Search** to start the search.

In the **Search** dialog box, you can specify additional parameters, such as **Match Case** and **Search Attributes Only**.

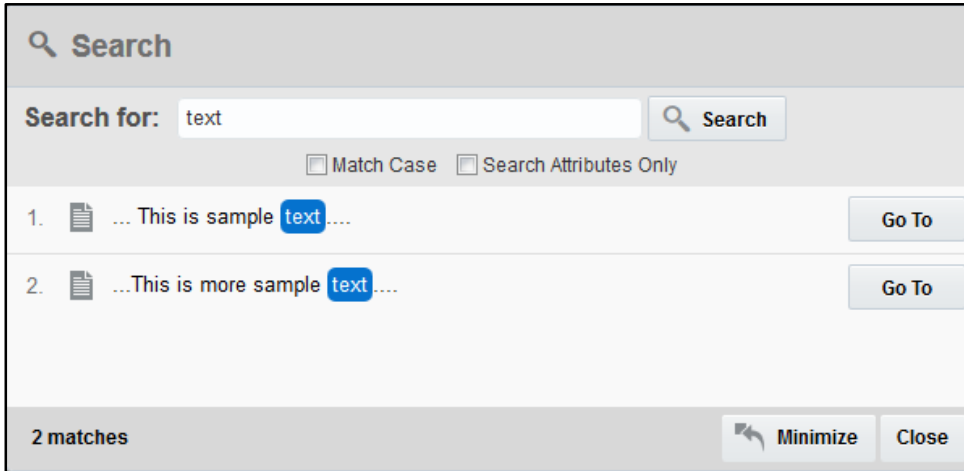


Click **Minimize** to hide the **Search** dialog box and make the **Search** panel on the left active.

The results of a search appear in the **Search** dialog box. Search looks for a string throughout the template—including the text content of all textual Elements like Text Elements, Table of Contents

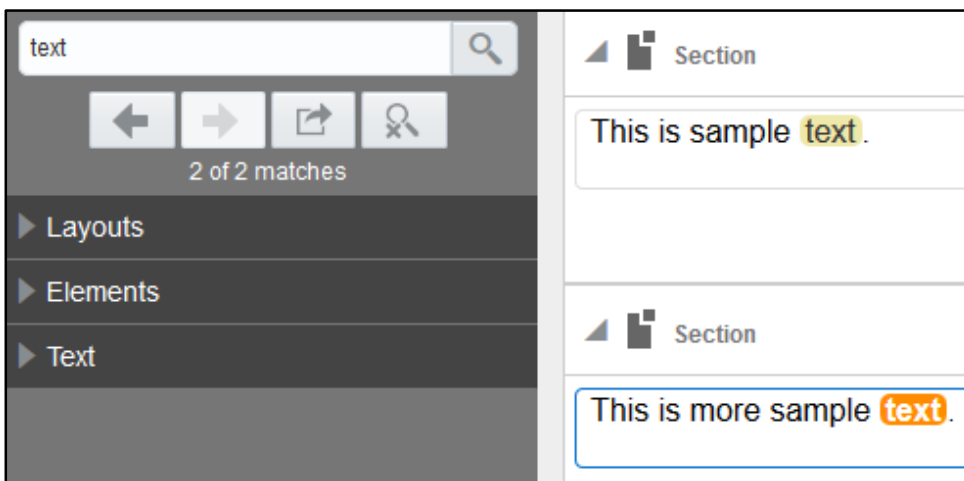
Layouts, attribute labels, Heading Elements, and XSL Snippet Elements. The string does not have to be visible to the user for it to be found.

The first time you search for a string, the **Search** dialog box appears. In the Template Flow, all matching keywords (in open Layouts) will be highlighted. Collapsed Layouts will not be highlighted.



Each result in the **Search** dialog box has a **Go To** button associated with it. Clicking **Go To** for any result:

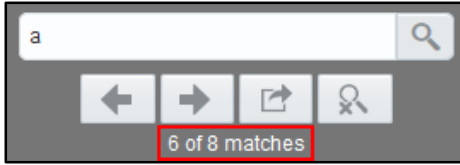
- Minimizes the **Search** dialog box
- Makes the **Search** panel on the left active
- Expands a collapsed Section/Layout, if necessary
- Selects/navigates to the specific match/Element



#### USING THE SEARCH PANEL

When searching from the **Search** panel on the left, all the matching keywords in open Layouts will be highlighted. You can also see which result you are currently viewing, out of the total number of results.





The **Search** panel includes the following buttons:

Button	Button Name	Description
	Previous	Move between matching items. If the subsequent item is in a collapsed Layout, the Layout expands and you will be taken to the matching item. <b>Note:</b> When search is on the first match, the <b>Previous</b> button is disabled.
	Next	Move between matching items. If the subsequent item is in a collapsed Layout, the Layout expands and you will be taken to the matching item. <b>Note:</b> When search is on the last match, the <b>Next</b> button is disabled.
	Search Results List	Return to the <b>Search Results</b> list.
	Cancel	Close the <b>Search Results</b> list.

#### SEARCH TIPS AND CONSIDERATIONS

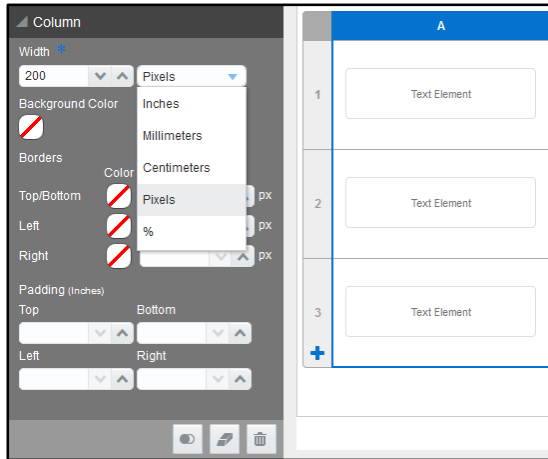
When searching for items that are contained within an XSL Snippet, specific tags and terms can be entered into the **Search** field. The entire XSL Snippet does not need to be entered in the **Search** Field. For example, if an XSL Snippet contained `<xsl>data</xsl>`, search terms could be `xsl` or `data`.

For multi-language templates, search is confined to the currently selected language.

The **Search** field can be used to search for attribute labels in the template, but not attribute values. The attribute values are inserted into the template when it is generated.

#### TABLE COLUMN WIDTH IN DOCUMENT DESIGNER

In the Document Designer, in addition to pixels, you can now choose the width of a Table column in inches, millimeters, centimeters, and percentage (%) of page width.



**Note:** When specifying a column width as a percentage of the total page width, no column can have a width of more than 100%.

After changing % values for different column widths, the effect can be seen in the document preview/print.

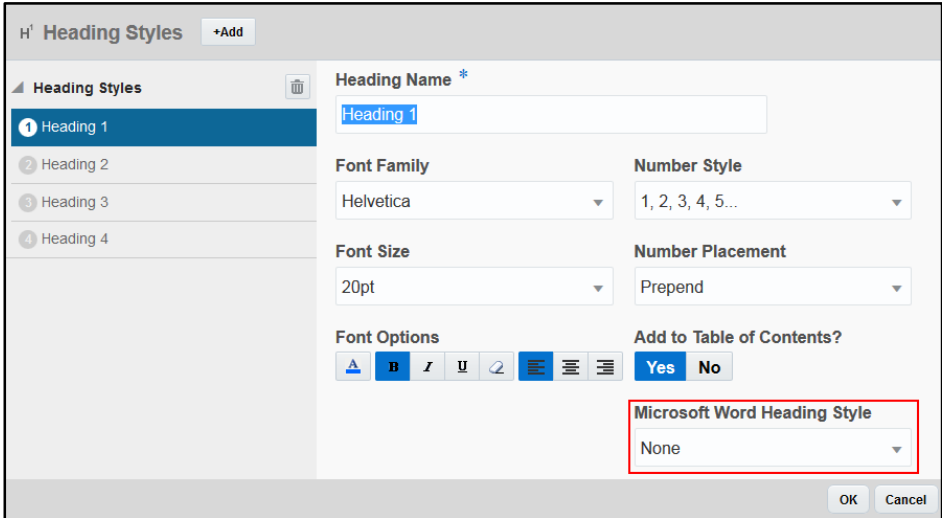
There is no validation if the sum of all column widths, in percent, is greater than 100% or not. Since columns can be dynamically shown or hidden, it is up to the admin to make sure column widths do not exceed page margins.

If the sum of column widths for all the columns is greater than 100%, content on the right will be cut off.

All columns in a Table must use the same units for width. Changes to the width unit for a single column will apply to all columns in the same Table.

**MICROSOFT WORD HEADING STYLES**

Heading styles can now be replicated from Microsoft Word. Instead of having to recreate the style elements of your Microsoft Word styles in all the Headings in Document Designer, you can create documents in Document Designer that match your existing Microsoft Word document styles.



These Heading Styles will display in the selected Microsoft Word Heading Style when the printed output is DOCX, or if you are opening RTF output in Microsoft Word. The Heading Styles will reflect the Document Designer settings if the document is output as a PDF.

#### USING MICROSOFT WORD HEADING STYLES

The Heading Style will display in the corresponding Microsoft Word Heading Style when the printed output is RTF or DOCX. The Heading Style will be the values specified in the Document Designer if the document is printed as a PDF.

The CPQ Cloud default Heading Styles mimic the Heading Styles in the default installation of Microsoft Word, version 2013. Your installation of Microsoft Word may have different styles.

**Note:** When a Heading Style has an equivalent Microsoft Word style, Heading Style properties can still be modified in the Document Designer. These edits will only appear when a document is printed as a PDF.

For more information on Headings, log in to CPQ Cloud as a FullAccess and see the [Document Designer: Heading Styles and Style Sets](#) topic in Online Help.

#### MICROSOFT WORD HEADING STYLES AND THE TABLE OF CONTENTS

The Table of Contents is equivalent to a Microsoft Word Table of Contents when printing in DOCX format. The Title Element will follow the built-in Word style for TOC Heading.

Just as with Document Designer Headings, after generating the Word output, if the Section Headings change, that change will be reflected in the Table of Contents. If content is moved or modified, page numbers in the Table of Contents will update as well.

**Notes:** As with all document changes, the Table of Contents in the DOCX file must be updated before it shows any changes.

#### MICROSOFT WORD HEADING STYLES TIPS AND CONSIDERATIONS

- If you change the language of a multi-language template, the Table of Contents will reflect that change in the output.
- You can have more than one Table of Contents Layout in the Template Flow. When printing in the DOCX format, it will show each Table of Contents in the style chosen in the Table of Contents properties. These styles could be the same, or different.

#### DOCUMENT GENERATOR REST API

---

A new REST API, documentGenerator, has been exposed to allow an external system to send a REST call to request a published Document Designer document from CPQ Cloud based on simple parameters. When executed correctly, the external system will receive the URL of the document in the Location header of the REST response from CPQ Cloud.

The call must contain HTTP Basic Authentication with valid CPQ Cloud FullAccess credentials in the header, as well as the Commerce Process, Document Designer template, Transaction ID, language, and output type that the document will be generated by specified in the body.

documentGenerator API											
<b>Description</b>	<p><i>POST</i>—Returns the URL of a document generated by a Document Designer template.</p> <p><i>GET</i>—Returns schema for the resource (REST call does not require a payload).</p>										
<b>URI Endpoint</b>	https://{siteurl}/rest/v1/documentGenerator										
<b>Endpoint Parameter</b>	<table border="1"> <tr> <td>{siteurl}</td> <td>The base URL of the CPQ Cloud site.</td> </tr> </table>	{siteurl}	The base URL of the CPQ Cloud site.								
{siteurl}	The base URL of the CPQ Cloud site.										
<b>Headers</b>	<p>Accept (<i>Basic Authentication with valid CPQ Cloud FullAccess credentials</i>)</p> <p>Content-Type: application/json</p>										
<b>Payload Structure (Parameters Bolded and Red)</b>	<pre>{   "processVarname": "[Process Var Name]",   "templateName": "[Template Name]",   "transactionId": [Transaction ID],   "languageCode": [Language Code],   "outputFormat": "[Output Format]" }</pre>										
<b>Payload Parameters</b>	<table border="1"> <tr> <td>[Process Var Name]</td> <td>The variable name of the Commerce Process. <i>All Commerce Processes' variable names are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i></td> </tr> <tr> <td>[Template Name]</td> <td>The name of the Document Designer Template. <i>All Document Designer template names (and the Commerce Processes they are in) are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i></td> </tr> <tr> <td>[Transaction ID]</td> <td>The Transaction ID of the Transaction from which the document will be generated.</td> </tr> <tr> <td>[Language Code]</td> <td>The language code of the language in which the document will be generated. <i>All language codes are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i></td> </tr> <tr> <td>[Output Format]</td> <td>The output type in which the document will be generated. <i>All output formats are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i></td> </tr> </table>	[Process Var Name]	The variable name of the Commerce Process. <i>All Commerce Processes' variable names are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>	[Template Name]	The name of the Document Designer Template. <i>All Document Designer template names (and the Commerce Processes they are in) are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>	[Transaction ID]	The Transaction ID of the Transaction from which the document will be generated.	[Language Code]	The language code of the language in which the document will be generated. <i>All language codes are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>	[Output Format]	The output type in which the document will be generated. <i>All output formats are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>
	[Process Var Name]	The variable name of the Commerce Process. <i>All Commerce Processes' variable names are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>									
	[Template Name]	The name of the Document Designer Template. <i>All Document Designer template names (and the Commerce Processes they are in) are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>									
	[Transaction ID]	The Transaction ID of the Transaction from which the document will be generated.									
	[Language Code]	The language code of the language in which the document will be generated. <i>All language codes are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>									
[Output Format]	The output type in which the document will be generated. <i>All output formats are listed in the REST API metadata schema. See the <a href="#">Document Generator REST API Metadata</a> section for more information.</i>										
<b>HTTP Methods</b>	POST, GET (Does not require a payload)										

documentGenerator API	
Success Response	<p><b>204 – Success</b></p> <p><i>POST</i>—The URL of the generated document will be in the response’s Location header.</p> <p><i>GET</i>—The schema of the resource will be in the response body.</p>
Failure Responses	<p><b>400 – Bad request</b></p> <p>An error message will say whether the Commerce Process variable name, Transaction ID, template, language, output, or a combination of those parameters were not found, or if there was an error deploying or printing the template.</p> <p><b>401 – Unauthorized</b></p> <p>Valid CPQ Cloud FullAccess credentials were not provided in the header.</p>
Sample URI Endpoint	https://site.oracle.com/rest/v1/documentGenerator
Sample Payload (Parameters Bolded and Red)	<pre>{   "processVarname": "northAmericaQuotes",   "templateName": "Proposal Template",   "transactionId": 12345678,   "languageCode": -1,   "outputFormat": "PDF" }</pre>
Notes	<ul style="list-style-type: none"> <li>• Even if a CPQ Cloud site only has one supported language, that language’s code must still be included in the body of the POST REST call.</li> <li>• Only deployed Document Designer templates can produce a document via a REST call.</li> </ul>

#### DOCUMENT GENERATOR REST API METADATA

In addition to getting the schema through a GET call, when writing documentGenerator REST API calls, admins can access the resource’s metadata schema through the Interface Catalogs (Admin > Interface Catalogs). Clicking the **documentGenerator** resource will open the metadata schema in a new window.

The screenshot shows the 'Interface Catalogs' search interface. The search criteria are: Name: documentGenerator, Description: (empty), Interface Type: REST. A 'Search' button is present. Below the search criteria, a table lists the results:

Interface Type	Name	Description
REST	documentGenerator	

At the bottom right of the interface, there are 'Back To Top' and 'Back' buttons.

The metadata schema includes all possible Commerce Process variable names (`processVarname`), Document Designer template names (`templateName`), language codes (`languageCode`), and output formats (`outputFormat`) in each corresponding `description` object that can be used in a REST call to the `documentGenerator` API, and should be used to inform REST calls that are being written.

```

    "type" : "integer",
    "description" : "Resource identifier"
  }
  "processVarname" : {
    "title" : "Process Varname",
    "type" : "string",
    "description" : "The available process var name values are (separated by a comma) :
transaction_bmClone_2, migration, formula"
  }
  "templateName" : {
    "title" : "Template Name",
    "type" : "string",
    "description" : "Available template names by process are provided in the <process var name>:
[<template name>|^|<template name>] format. The different template names for a process are separated by |^|.
transaction_bmClone_2:[t1|^|test], migration:[test4]"
  },
  "transactionId" : {
    "title" : "Transaction ID",
    "type" : "integer",
    "description" : "Commerce transaction identifier"
  },
  "languageCode" : {
    "title" : "Language Code",
    "type" : "integer",
    "description" : "The following available integer values to specify the language for the template
are in the form <language name>:<language code>. : English:-1, German:0, Chinese (Simplified) [China]:5,
Chinese (Traditional) [Taiwan]:6, Japanese [Japan]:19, Polish [Poland]:21"
  },
  "outputFormat" : {
    "title" : "Output Format",
    "type" : "string",
    "description" : "The output format options available are as follows (separated by comma): PDF,
DOCX, RTF"
  }
},
"links" : [ /

```

**Sample documentGenerator Resource Metadata Schema**

**STEPS TO ENABLE**

The Document Designer enhancements are automatically available on all 2015 R2 sites.

**KEY RESOURCES**

2015 R1 Release Readiness [Document Designer](#) video

**EMAIL DESIGNER ENHANCEMENTS**

As part of continuing improvements to the Email Designer, formerly known as the Email Template Editor, several new features have been added in this release.

- Set up standard email notifications, such as system-generated email notifications, using Email Designer templates. For example, when a Transaction is approved, rejected, or revised, CPQ Cloud can send an Email Designer template-generated email to relevant users.

- Search for static text or dynamic attribute values within Email Designer templates using a new Search field.

In addition to what can be found by a browser search, an Email Designer search will also search Elements that are in a collapsed section of the editor. With this functionality, the entire template is always searched, even if some parts of it are not visible to the user.

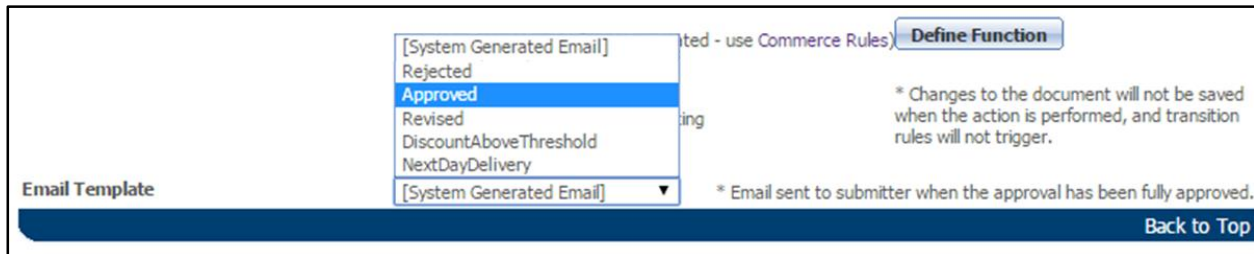
- Define the width of a Table column in inches, millimeters, and centimeters, in addition to the existing ability to define column width in pixels.
- Call a REST API from an external system to produce an HTML output from an Email Template based on simple parameters.

## EMAIL DESIGNER TEMPLATES FOR SUBMIT SUB-ACTION NOTIFICATIONS

Email Designer templates can now be used to design the following notification emails that are automatically sent by CPQ Cloud when a Submit sub-action (Revise, Reject, or Approve) is invoked:

- The "You no longer need to Approve" email sent to an approver when the Revise sub-action is invoked.
- The "Your Submission has been Rejected" email sent to the user that submitted the Transaction when the Reject sub-action is invoked.
- The "Your Submission has been Approved" email sent to the user that submitted the Transaction when the final Approve sub-action is invoked.

An Email Template menu has been added to the **General** tab of the Revise, Reject, and Approve sub-action's admin page to allow admins to associate an Email Template with the sub-action.



**[System Generated Email]** will always be the first option in the **Email Template** drop-down menu. If **[System Generated Email]** is selected, CPQ Cloud will continue to send the default, system-generated email to the appropriate recipient whenever the sub-action is invoked.

Deployed Email Designer templates in the Commerce process will appear below **[System Generated Email]** in the **Email Template** drop-down menu. If an Email Designer template is selected, CPQ Cloud will use the template to build the email that is sent to the appropriate recipient whenever the sub-action is invoked.

## SEARCH IN EMAIL DESIGNER

Admins can now use the new **Search** field, in the **Properties** Panel of the Email Designer editor, to search for characters, words, phrases, and so on in an Email Designer template.



In addition to what can be found by a browser search, Email Designer search functionality will also search Elements that are collapsed in the Template Flow. With this functionality, the entire template is always searched, even if some parts of it are not visible to the user.

The following table describes which Elements and template components can be searched:

Email Designer Element/Component	Plain Text Search	Match Case	Attribute Search
Text*	✓	✓	✓
Subject	✓	✓	✓

\*Applies to Text Elements both inside and outside of Table cells

### SEARCH TYPES

- Plain Text**  
 Type of Search executed when Search Attributes Only checkbox is unselected (default). Search term is compared against the text content of textual Elements and attribute labels in textual Elements, which are considered part of the normal flow of text.
- Attribute**  
 Type of Search executed when Search Attributes Only checkbox is selected. Search term is compared against only the variable names and labels of Attributes contained within textual Elements.
- Match Case**  
 When Match Case checkbox is selected (unselected is default), case sensitivity is applied to both Plain Text and Attribute Searches.

Type text into the **Search** field and either press **Enter** or click the magnifying glass to open the **Search** dialog box, shown below.



In the **Search** dialog box, you can specify, additional parameters such as **Match Case** and **Search Attributes Only**.

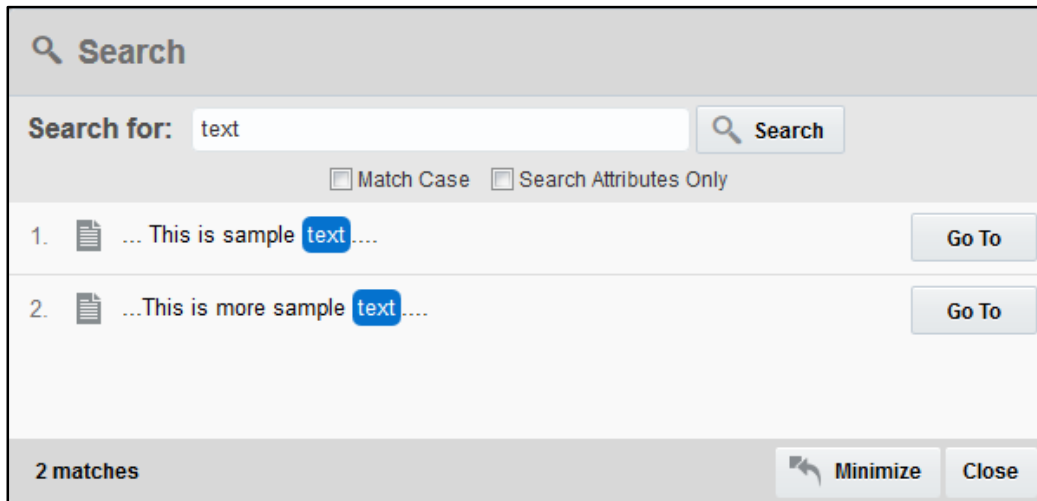


Click **Minimize** to hide the **Search** dialog box and make the **Search** panel on the left active.

Enter content in the **Search for:** field, if necessary, and either press **Enter** or click **Search** to start the search.

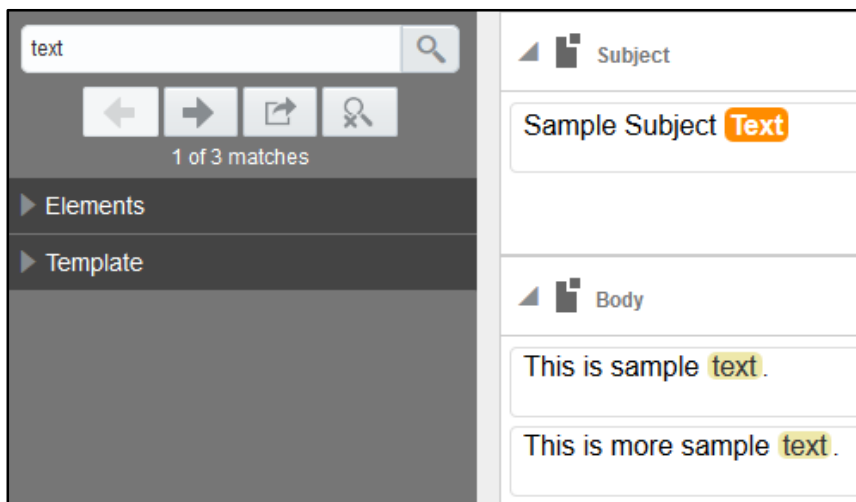
The results of a search appear in the **Search** dialog box. Search looks for a string throughout the template—including the text content of all Textual elements like the Text, the Subject, and attribute labels. The string does not have to be visible to the user for it to be found.

The first time you search for a string, the **Search** dialog box appears. In the Template Flow, all matching keywords will be highlighted.



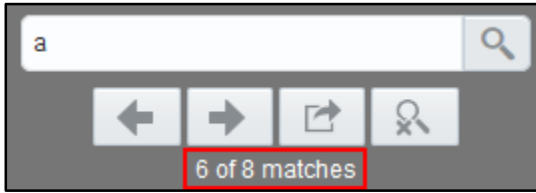
Each result in the **Search** dialog box has a **Go To** button associated with it. Clicking **Go To** for any result:

- Minimizes the **Search** dialog box
- Makes the **Search** panel on the left active
- Expands a collapsed section, if necessary
- Selects/navigates to the specific match/element



## USING THE SEARCH PANEL

When searching from the **Search** panel on the left, all the matching keywords in open Sections will be highlighted. You can also see which result you are currently viewing, out of the total number of results.



The **Search** panel includes the following buttons:

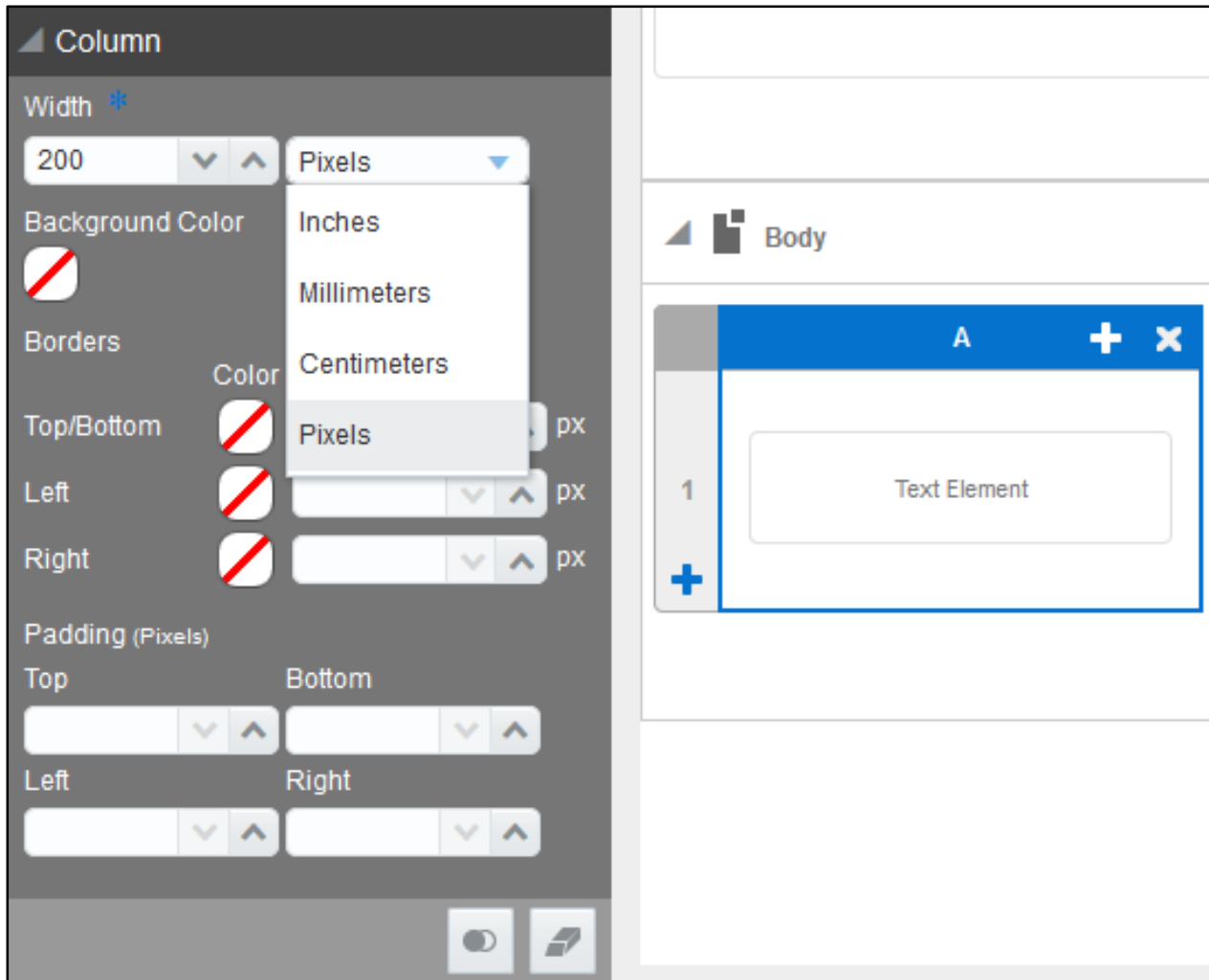
Button	Button Name	Description
	Previous	Move between matching items. <b>Note:</b> When search is on the first match, the <b>Previous</b> button is disabled.
	Next	Move between matching items. <b>Note:</b> When search is on the last match, the <b>Next</b> button is disabled.
	Search Results List	Return to the <b>Search Results</b> list.
	Cancel	Close the <b>Search Results</b> list.

### SEARCH TIPS AND CONSIDERATIONS

- For multi-language templates, search is confined to the currently selected language.
- The **Search** field can be used to search for attribute labels in the template, but not attribute values. The attribute values are inserted into the template when it is generated.

### TABLE COLUMN WIDTH IN EMAIL DESIGNER

In the Email Designer, you can now choose the width of a Table column in pixels, inches, millimeters, and centimeters.



All columns in a Table must use the same units for width. Changes to the width unit for a single column will apply to all columns in the same Table.

#### [EMAIL GENERATOR REST API](#)

A new REST API, `emailGenerator`, has been exposed to allow an external system to send a REST call to request an HTML Email Designer email from CPQ Cloud based on simple parameters. When executed correctly, the external system will receive the URL of the generated HTML email in the Location header of the REST response from CPQ Cloud.

The call must contain HTTP Basic Authentication with valid CPQ Cloud FullAccess credentials in the header, as well as the Commerce Process, Email Designer template, Transaction ID, and language that the email will be generated by specified in the body.

emailGenerator API									
<b>Description</b>	<p><i>POST</i>—Returns the URL of an HTML email generated by an Email Designer template.</p> <p><i>GET</i>—Returns schema for the resource (REST call does not require a payload).</p>								
<b>URI Endpoint</b>	https://{siteurl}/rest/v1/emailGenerator								
<b>Endpoint Parameters</b>	<table border="1"> <tr> <td>{siteurl}</td> <td>The base URL of the CPQ Cloud site.</td> </tr> </table>	{siteurl}	The base URL of the CPQ Cloud site.						
{siteurl}	The base URL of the CPQ Cloud site.								
<b>Headers</b>	<p>Accept (<i>Basic Authentication with valid CPQ Cloud FullAccess credentials</i>)</p> <p>Content-Type: application/json</p>								
<b>Payload Structure (Parameters Bolded and Red)</b>	<pre>{   "processVarname": "[Process Var Name]",   "templateName": "[Template Name]",   "transactionId": [Transaction ID],   "languageCode": [Language Code] }</pre>								
<b>Payload Parameters</b>	<table border="1"> <tr> <td>[Process Var Name]</td> <td>The variable name of the Commerce Process. <i>All Commerce Processes' variable names are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i></td> </tr> <tr> <td>[Template Name]</td> <td>The name of the Email Designer Template. <i>All Email Designer template names (and the Commerce Processes they are in) are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i></td> </tr> <tr> <td>[Transaction ID]</td> <td>The Transaction ID of the Transaction from which the email will be generated.</td> </tr> <tr> <td>[Language Code]</td> <td>The language code of the language in which the email will be generated. <i>All language codes are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i></td> </tr> </table>	[Process Var Name]	The variable name of the Commerce Process. <i>All Commerce Processes' variable names are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i>	[Template Name]	The name of the Email Designer Template. <i>All Email Designer template names (and the Commerce Processes they are in) are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i>	[Transaction ID]	The Transaction ID of the Transaction from which the email will be generated.	[Language Code]	The language code of the language in which the email will be generated. <i>All language codes are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i>
	[Process Var Name]	The variable name of the Commerce Process. <i>All Commerce Processes' variable names are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i>							
	[Template Name]	The name of the Email Designer Template. <i>All Email Designer template names (and the Commerce Processes they are in) are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i>							
	[Transaction ID]	The Transaction ID of the Transaction from which the email will be generated.							
[Language Code]	The language code of the language in which the email will be generated. <i>All language codes are listed in the REST API metadata schema. See the <a href="#">Email Generator REST API Metadata</a> section for more information.</i>								
<b>HTTP Methods</b>	POST, GET (Does not require a payload)								
<b>Success Response</b>	<p><b>204 – Success</b></p> <p><i>POST</i>—The URL of the generated email will be in the response's Location header.</p> <p><i>GET</i>—The schema of the resource will be in the response body.</p>								

emailGenerator API	
<b>Failure Responses</b>	<p><b>400 – Bad request</b> An error message will say whether the Commerce Process variable name, Transaction ID, template, language, or a combination of those parameters were not found, or if there was an error deploying or generating the template.</p> <p><b>401 – Unauthorized</b> Valid CPQ Cloud FullAccess credentials were not provided in the header.</p>
<b>Sample URI Endpoint</b>	https://site.oracle.com/rest/v1/emailGenerator
<b>Sample Payload (Parameters Bolded and Red)</b>	<pre>{   "processVarname": "northAmericaQuotes",   "templateName": "Discount Approval Template",   "transactionId": 12345678,   "languageCode": -1 }</pre>
<b>Notes</b>	<ul style="list-style-type: none"> <li>• Even if a CPQ Cloud site only has one supported language, that language's code must still be included in the body of the POST REST call.</li> <li>• Only deployed Email Designer templates can produce an HTML email via a REST call.</li> </ul>

#### EMAIL GENERATOR REST API METADATA

In addition to getting the schema through a GET call, when writing emailGenerator REST API calls, admins can access the resource's metadata schema through the Interface Catalogs (Admin > Interface Catalogs). Clicking the **emailGenerator** resource will open the metadata schema in a new window.

The screenshot shows the 'Interface Catalogs' search interface. The search criteria are: Name: emailGenerator, Description: (empty), Interface Type: REST. A 'Search' button is present. Below the search criteria, a table lists the results:

Interface Type	Name	Description
REST	emailGenerator	

The 'emailGenerator' entry in the table is highlighted with a red box. At the bottom right of the interface, there are 'Back To Top' and 'Back' buttons.

The metadata schema includes all possible Commerce Process variable names (`processVarname`), Email Designer template names (`templateName`), and language codes (`languageCode`) in each corresponding `description` object that can be used in a REST call to the `emailGenerator` API, and should be used to inform REST calls that are being written.

```

    "type" : "integer",
    "description" : "Resource identifier"
  }
  "processVarname" : {
    "title" : "Process Varname",
    "type" : "string",
    "description" : "The available process var name values are (separated by a comma) :
transaction_bmClone_2, migration, formula"
  }
  "templateName" : {
    "title" : "Template Name",
    "type" : "string",
    "description" : "Available template names by process are provided in the <process var name>:
[<template name>|^|<template name>] format. The different template names for a process are separated by |^|.
transaction_bmClone_2:[t1|^|test], migration:[test4]"
  },
  "transactionId" : {
    "title" : "Transaction ID",
    "type" : "integer",
    "description" : "Commerce transaction identifier"
  }
  "languageCode" : {
    "title" : "Language Code",
    "type" : "integer",
    "description" : "The following available integer values to specify the language for the template
are in the form <language name>:<language code>. : English:-1, German:0, Chinese (Simplified) [China]:5,
Chinese (Traditional) [Taiwan]:6, Japanese [Japan]:19, Polish [Poland]:21"
  },
}

```

**Sample emailGenerator Resource Metadata Schema**

**STEPS TO ENABLE**

The Email Designer enhancements are automatically available on all 2015 R2 sites.

**ACCESS DATA TABLE REST API**

A new REST API, `custom{DataTable}`, has been exposed to allow an external system to access deployed data in CPQ Cloud Data Tables. When executed correctly, the external system will receive the CPQ Cloud Data Table’s deployed data in the REST response body.

**Note:** Currently only Read operations are supported when interacting with the `custom{DataTable}` API.

The call must contain HTTP Basic Authentication with valid CPQ Cloud FullAccess credentials in the header, as well as the Data Table name in the URI endpoint.

custom{DataTable} API	
<b>Description</b>	Returns the deployed data of a CPQ Cloud Data Table.

custom{DataTable} API		
URI Endpoint	https://{siteurl}/rest/v1/custom{DataTable}{Query}	
Endpoint Parameters	{siteurl}	The base URL of the CPQ Cloud site.
	{DataTable}	The name of the CPQ Cloud Data Table (first character must be capitalized).
	{Query}	Optional query specifications to organize or filter Data Table return data. Query specifications that follow CPQ Cloud query and pagination parameters syntax, and query specifications that follow a subset of MongoDB syntax can be used. For more information on querying CPQ Cloud REST APIs, log in to CPQ Cloud as a FullAccess user and see the “Query Specification Syntax” topic in the Online Help. For more information on MongoDB syntax, see <a href="#">MongoDB’s Query Documents documentation</a> .
Headers	Accept (Basic Authentication with valid CPQ Cloud FullAccess credentials)	
	Content-Type: application/json	
HTTP Method	GET	
Success Response	<b>204 – Success</b> The Data Table’s deployed data will be in the response body.	
Failure Responses	<b>400 – Bad request</b> The name of the Data Table was invalid or the Data Table had not been deployed.	
	<b>401 – Unauthorized</b> Valid CPQ Cloud FullAccess credentials were not provided in the header.	
Sample URI Endpoint	https://site.oracle.com/rest/v1/customPricing2015Q4	

## ACCESS DATA TABLE REST API METADATA

Admins can access a Data Table resource’s metadata schema through the Interface Catalogs (Admin > Interface Catalogs).

Each deployed Data Table’s resource will be listed in the Interface Catalogs using the custom{DataTable} naming format, where {DataTable} is the name of the Data Table with the first character capitalized. Clicking the name of a Data Table’s resource will open will open the metadata schema in a new window.

**Interface Catalogs**

**Search**

Name

Description

Interface Type: REST ▾

Interface Type	Name	Description
REST	customPricing2015Q4	

[Back To Top](#)

The metadata schema includes Data Table column names (`title`), column data types (`type`), and column descriptions (`description`).

```

{
  "id": {
    "title": "id",
    "type": "integer",
    "description": ""
  },
  "pricelista": {
    "title": "PricelistA",
    "type": "number",
    "description": "Pricing multipliers for customers in Price List A"
  },
  "pricelistb": {
    "title": "PricelistB",
    "type": "number",
    "description": "Pricing multipliers for customers in Price List B"
  },
  "pricelistc": {
    "title": "PricelistC",
    "type": "number",
    "description": "Pricing multipliers for customers in Price List C"
  }
},
"links": [ ]

```

**Sample custom{DataTable} Resource Metadata Schema**

**Note:** Columns with a Float data type have their data types described as `number` in the metadata schema. String and Integer columns are described as `string` and `integer` respectively in the metadata schema.

**STEPS TO ENABLE**

---

The custom{DataTable} REST API is automatically available on all 2015 R2 sites.

**KEY RESOURCES**

---

[MongoDB's Query Documents](#)



CPQ Cloud helps sales organizations sell more, sell faster, and sell anywhere by providing sales reps with modern configuration, pricing, and quoting technology. Oracle is continually upgrading the functionality of CPQ Cloud to meet the needs of customers who must conform to a variety of regulatory and compliance paradigms.

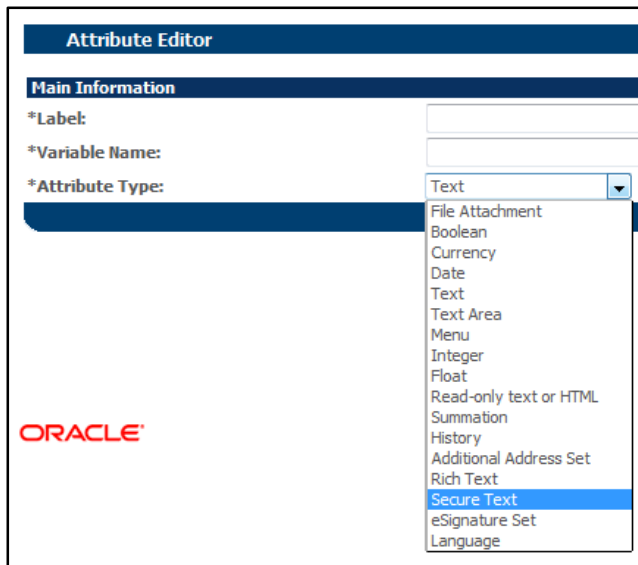
New features to reinforce enterprise excellence include:

- Secure Attributes
- Part Administration Enhancements
- Internal Property to Prevent Posting of Read-Only Fields

### SECURE ATTRIBUTES

With Secure Attributes, a new Commerce attribute type, data a user inputs is encrypted when it reaches the CPQ Cloud Server and remains encrypted while in memory. The encrypted value remains available to currently active Commerce actions. During this active request processing time, the customer can have the value sent to another application, such as for secure data storage.

In the CPQ Cloud database, the data for Secure Attribute fields will appear as the value “\_mask\_text” when viewed in XML from Web Services.



With a Secure Attribute field on a Commerce layout, CPQ Cloud can capture a value as it is being entered—while masking the entry as if it were a password—and use the Java RSS encryption standard to encrypt the data, without ever storing the original value in CPQ Cloud.

While a CPQ Cloud action (such as Save) is active, the encrypted data is temporarily stored in memory. While it is in memory, it can be transferred to the customer’s database, which will handle data storage, security, and any further encryption and decryption.

CPQ Cloud encryption uses standard Java libraries, including RSA standard and Optimal Asymmetric Encryption Padding. The public key, an SSL certificate with a minimum key length of 2048, must be uploaded to the Commerce Process. The private key, for the customer’s encryption and decryption, remains with the customer.

The screenshot shows the 'Process Administration' interface with the 'Integration' tab selected. The 'Process Manager Settings' section includes fields for Process Name, Variable Name, Description, Tab Label, Page Length, Auto Fill Options, and Fiscal Year Start Date. The 'Secure Attribute Encryption Key' field is highlighted with a red box and contains a 'Choose File' button and the text 'No file chosen'. A 'Description' column on the right provides details for several fields, including the Page Length and Fiscal Year Start Date. At the bottom right, there are buttons for 'Translations', 'Apply', 'Update', and 'Back', along with a 'Back to Top' link.

**Note:** You cannot use Secure Attributes without a Secure Attribute Encryption Key.

If you have a Secure Attribute on a layout, but no Secure Attribute Encryption Key, the following error message appears: “Unable to encrypt secure attribute. Please contact System Administrator.” when the user performs any action.

Public keys are validated and stored in a secure keystore. At runtime, public keys are retrieved from the keystore and used for encryption. The private key should be retained and used to decrypt Secure Attribute values.

**Note:** CPQ Cloud does not hold the private key; it is retained by the customer’s system.

**Important:** You cannot migrate public keys between CPQ Cloud environments. The keys can be the same for each environment, but they must be uploaded to each Commerce Process on each environment individually.

## TRANSFERRING VALUES FROM MEMORY

After a value has been entered into a Secure Attribute field, the encrypted value must be captured from memory before a subsequent action removes the value from memory. For example, the value could be captured and moved to a secure server.

There are several ways to capture the value, but no matter which way you choose, add the capture method to an action to ensure that the value can’t be lost from memory. For example, by adding the capture method to the Save action, you will always capture the value of the Secure Attribute when the Transaction is saved.

Strategies for transferring the captured value include:

- BML function
- Integration XSL files that call a BML function
- Advanced rules that call a BML function

When capturing the value from memory, BML refers to the value's unique ID (based on the Secure Attribute's name).

## TIPS AND CONSIDERATIONS

---

- Each Commerce Process can have a Secure Attribute Encryption Key. The Keys can be the same for each Commerce Process, or unique to each Commerce Process.
- The Java framework used to support RSA standard and Optimal Asymmetric Encryption Padding is the Java Cryptography Extension (JCE) framework.
- Existing Secure Attributes are not impacted when uploading a public key, either for the first time or any subsequent time.
- Validation of the decrypted data can be performed on your server, but not as the value is being entered into CPQ Cloud.
- CPQ Cloud is not PCI-compliant, but using one of the transfer strategies discussed above you can integrate CPQ Cloud with a PCI-compliant system that you build and manage.

## FUNCTIONAL KNOWN ISSUES

---

### SECURE ATTRIBUTES LOSE VALUE IN MOBILE

In mobile Commerce, if a user enters a value into a Secure Attribute and then switches to a new tab on the mobile layout before saving the Transaction, the value of the Secure Attribute will be lost without the user being warned.

**Workaround:** After entering a value in a Secure Attribute while in Mobile Commerce, users should save the Transaction before switching to a new tab on the mobile layout. It is recommended that users save the Transaction immediately after entering a value in a Secure Attribute to avoid any chance of this issue occurring.

### SECURE ATTRIBUTE VALUES ONLY AVAILABLE TO FIRST AJAX RULES EXECUTION OR SAVE

On a cached Transaction, if a user edits a Secure Attribute and then multiple, separate AJAX rule executions take place before an action is invoked, the Secure Attribute's encrypted value will only be available to the rules in the first AJAX rule execution to run. Additionally, when a Secure Attribute is edited and then the Transaction is saved, the Secure Attribute's encrypted value will not be available to subsequent AJAX rule executions.

**Workaround:** It is recommended not to use a Secure Attribute's encrypted value in AJAX rules. However, if a Secure Attribute's encrypted value must be available to an AJAX rule, the AJAX rule should be configured so that it will be executed before another AJAX rule execution or a potential save (both invoked by rules or by the user).

## STEPS TO ENABLE

---

The ability to create Secure Attributes is automatically available on all 2015 R2 sites.

After creating Secure Attributes and adding the Secure Attribute Encryption Key to the Commerce Process, add the attribute to a Commerce layout so that it can be modified by users.

---

## PART ADMINISTRATION ENHANCEMENTS

Two new features have been added to Parts administration functionality in CPQ Cloud:

- A part can be identified by an alternate part number, called a Part Display Number.
- Parts can now be associated with one or more Companies, including the Host Company, so that only those specific Companies' users can access the parts.

---

## PART DISPLAY NUMBER

When CPQ Cloud customers use Partner Organizations—other companies that sell products on their behalf—there may be instances when a Partner Organization uses a different part number for the same item. Or, a part can be called by the same name but may refer to different physical parts altogether for different companies.

Now you can add a second part identifier, called the Part Display Number, to a part.

**Note:** Only one value is allowed in the **Part Display Number** field for each part.

With the new **Part Display Number** field on a part's administration page, different parts can display the same part number—the Part Display Number—to sales users, while still having unique part numbers for use in CPQ Cloud parts administration.

Part Administration	
Product Information	
*Part Number:	CONSULTING-098
Part Display Number:	SERVICE-01

In the example in Table 1, three companies—the Host Company and two Partner Organizations—have three different parts that share the same Part Display Number, which is shown to each company's sales users.

	Host Company	Partner Organization 1	Partner Organization 2
Part Number	SERVICE-01	CONSULTING-098	PROFSERVICE-22
Part Display Number	SERVICE-01	SERVICE-01	SERVICE-01

Table 1

The Part Display Number attribute can be used like any other attribute:

- In Document Designer and Email Designer templates
- In Commerce rules
- In BML and BMQL

The `_part_display_number` field is available in parts Bulk Upload and Bulk Download, so Part Display Numbers can be added to parts in bulk.

For more information on bulk uploading and downloading, log in to CPQ Cloud as a FullAccess user and see the [Bulk Downloads](#) and [Bulk Uploads](#) topics in the Online Help.

## PART VISIBILITY USING COMPANY ASSOCIATIONS

---

New Company Associations allow admins to restrict the visibility of a part to users from selected Companies, including the users of the Host Company.

The screenshot displays the 'Part Administration' interface. The 'Product Information' section includes fields for Part Number (Part123), Part Display Number (Hammer), Direct Buy (Direct Buy), Description, Extended Description 1, Extended Description 2, Units, and Lead Time (Days). The 'Company Associations' section features two lists: 'Available' and 'Selected'. The 'Available' list contains items like 'a\_Co1BM [A1BM1]' through 'a\_Co7BM [A7BM1]'. The 'Selected' list contains '111 [Company One]', '222 [Company Two]', and 'slc09mea [Host Company]'. Navigation buttons (right arrow, left arrow) are positioned between the lists.

Available:	Selected:
a_Co1BM [A1BM1]	111 [Company One]
a_Co2BM [A2BM1]	222 [Company Two]
a_Co3BM [A3BM1]	slc09mea [Host Company]
a_Co4BM [A4BM1]	
a_Co5BM [A5BM1]	
a_Co6BM [A6BM1]	
a_Co7BM [A7BM1]	

If a Company has been denied access to a part, either through a part filter or the **Company Associations** field, it will not be available to the Company’s users in Parts Search or the Quick Key Line Items list. Table 2 describes part visibility in relation to Company Associations.

Users	Part Not Associated With any Company	Part Associated With Company 1	Part Associated With Company 2
All Users not in Company 1 or 2	Visible	Not Visible	Not Visible
Company 1 Users Only	Visible	Visible	Not Visible
Company 2 Users Only	Visible	Not Visible	Visible

**Table 2**

In the previous example in Table 1

three companies shared the same Part Display Number. By setting up visibility restrictions, an admin can ensure that the only SERVICE-01 part each user would see is the SERVICE-01 part from their company.

Company Associations are available in parts Bulk Upload and Bulk Download, so this information can be added to parts in bulk. If more than one company is associated with a part, the company names are comma delimited in the CSV file.

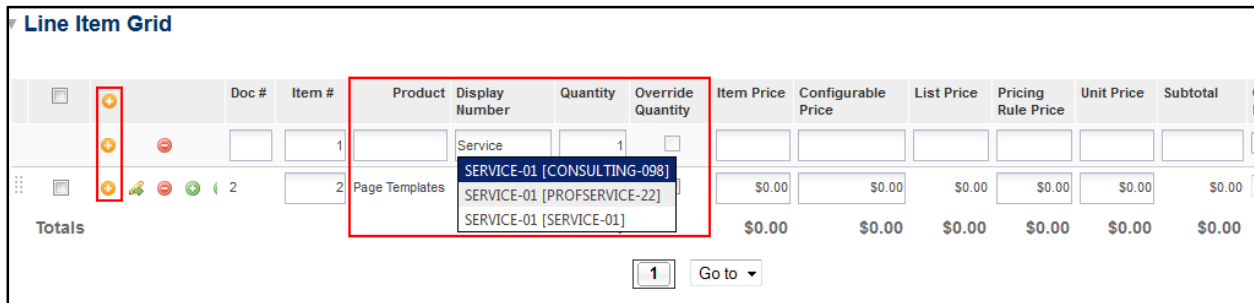
For more information on bulk uploading and downloading, log in to CPQ Cloud as a FullAccess user and see the [Bulk Downloads](#) and [Bulk Uploads](#) topics in the Online Help.

**QUICK KEYS**

Quick Key search, activated when the user clicks the Quick Key icon (+) can still perform a Part Number search. To do so, the **Part Number** column must be on the Line Item Grid of a Commerce layout.

Now, Quick Key search can also use the **Part Display Number** field during Line Item searches. Both the **Part Number** and the **Part Display Number** columns must be on the Line Item Grid of a Commerce layout in order to search Part Display Numbers.

The Quick Key search results will show the Part Display Number first (then the part number in brackets) using the following format: Part Display Number [Part Number].



## STEPS TO ENABLE

---

The **Part Display Number** field and **Company Associations** fields are automatically available on all 2015 R2 sites.

## TIPS AND CONSIDERATIONS

---

- Existing parts filters remain unchanged and in effect.
- Creating part/company associations may have a minor impact on Bulk Upload and Bulk Download performance.
- Parts that are not associated with any companies (the default for all existing parts) are accessible to all companies.
- The only parts available to a user are:
  - Parts that pass all existing part filters
  - Parts that are either not associated with any companies, or are associated with the user's company

---

## PREVENT POSTING OF READ-ONLY FIELDS

A new internal property gives CPQ Cloud customers the option to prevent Commerce read-only fields from being posted back to the server after a Commerce action is invoked.

While enabling this property will have nominal impact on performance, it is recommended not to post information back to the server when the data will not have changed since it was last posted.

## STEPS TO ENABLE

---

To enable this property, open a ticket on [My Oracle Support](#).

## INTEGRATION

Leverage the power of CPQ Cloud by integrating with other software applications. Integrations can be created with out-of-the-box point-to-point solutions, or customized for individual IT landscapes.

New integration features include:

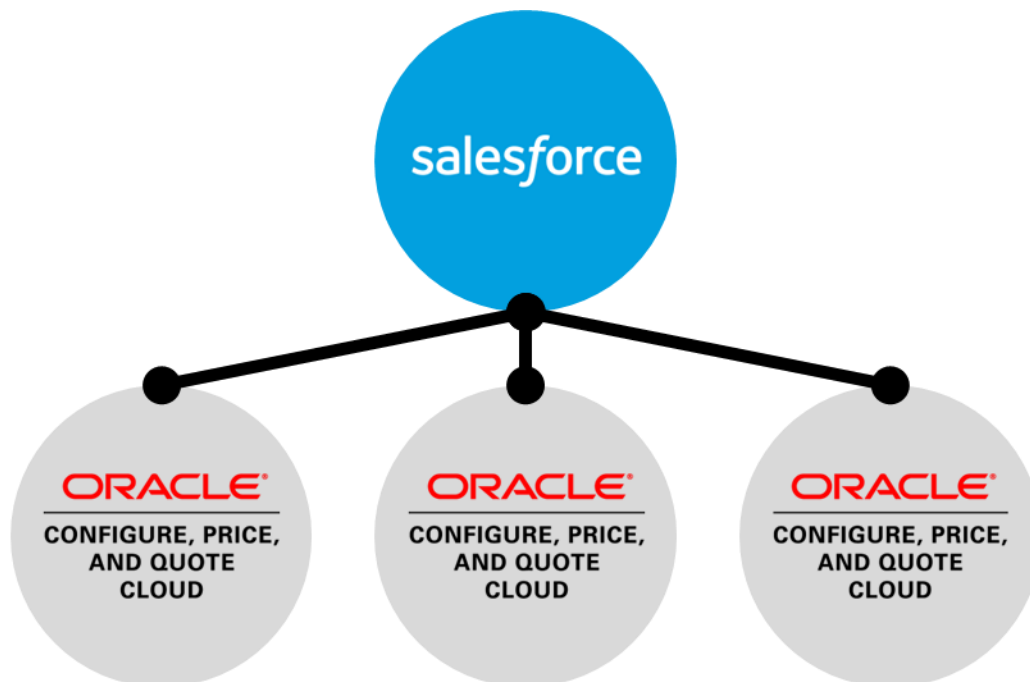
- Salesforce Integration Enhancements
- Oracle Commerce On Premise Integration Enhancements
- Decodebase64 BML Function
- Web Services v1.0 Access Modifications

### SALESFORCE INTEGRATION ENHANCEMENTS

The Oracle CPQ Cloud-Salesforce Integration Managed Package has been updated to v6.0 to allow organizations with one Salesforce site to simultaneously integrate with multiple CPQ Cloud sites, or integrate with multiple Commerce Processes on the same CPQ Cloud site, without requiring complex customizations.

### INTEGRATION WITH MULTIPLE CPQ CLOUD SITES

When multiple CPQ Cloud sites are integrated with a single Salesforce instance, each separate CPQ Cloud site can configure products and create quotes that are synced to the same location in Salesforce while maintaining separate product, pricing, and document requirements in CPQ Cloud.

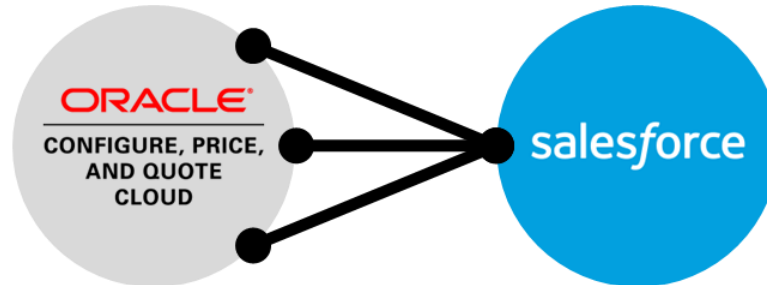




## INTEGRATION WITH MULTIPLE CPQ CLOUD COMMERCE PROCESSES

---

When multiple Commerce Processes on the same CPQ Cloud site are integrated with a single Salesforce instance, each separate CPQ Cloud Commerce Process can create quotes that are synced to the same location in Salesforce while maintaining separate pricing and document requirements in CPQ Cloud.



## STEPS TO ENABLE

---

The Oracle CPQ Cloud-Salesforce Integration Managed Package v6.0 must be installed in Salesforce to allow the site to integrate with multiple CPQ Cloud sites or multiple Commerce Processes on the same CPQ Cloud site. Additional changes must also be made to CPQ Cloud and Salesforce.

See the [Salesforce Integration Guide v6.0](#) on My Oracle Support for step-by-step instructions on upgrading to v6.0 of the Managed Package, or for step-by-step instructions for the initial implementation of a CPQ Cloud-Salesforce integration using v6.0 of the Managed Package.

## TIPS AND CONSIDERATIONS

---

**Important:** CPQ Cloud will no longer release updates to Salesforce integration Managed Packages prior to v5.0. With the release of 2015 R2, only Managed Packages v5.x and v6.x are officially supported. Although Salesforce integrations that use a Managed Package prior to v5.0 are still expected to function, new issues that arise in these versions will not be addressed by CPQ Cloud.

## KEY RESOURCES

---

[Salesforce Integration Guide v6.0](#)

## ORACLE COMMERCE ON PREMISE INTEGRATION ENHANCEMENTS

---

CPQ Cloud's existing integration with Oracle Commerce On Premise (Commerce) has been extended to further streamline the eCommerce process by offering a self-service configuration and checkout option for complex products, as well as customer-specific pricing.

To make these CPQ Cloud-Commerce Integration enhancements possible, the CPQ Cloud admin can now create custom Product Family integrations.

## SELF-SERVICE CONFIGURATION AND CHECKOUT FOR COMPLEX PRODUCTS

---

Once implemented, Commerce customers will be able to choose to configure a product, which will open a CPQ Cloud Configurator in the Commerce user interface within an embedded iframe. Preselected attribute values, associated with a Commerce stock keeping unit, can be present on the Configurator when it is opened in Commerce.

After configuring the product, the Commerce customer can reconfigure the product, check out, or request a quote, depending on the implementation.

Reconfiguration of the product will reopen the CPQ Cloud Configurator in the Commerce user interface within an embedded iframe. If the customer requests a quote, the remainder of the integration workflow will function in the same way as it did before the 2015 R2 CPQ Cloud-Commerce integration enhancements were enabled.

## CUSTOMER-SPECIFIC PRICING

---

Once implemented, when pricing a product configured by a Commerce customer, CPQ Cloud will apply any CPQ Cloud Pricing Rules that are linked to the customer's Customer Org ID.

## CUSTOM PRODUCT FAMILY INTEGRATIONS

---

In order for CPQ Cloud Configuration data to be sent from CPQ Cloud to Commerce after a product has been configured, and for the Commerce customer to be redirected to the Commerce pricing page, custom Product Family integrations can now be created.

**Note:** Custom Product Family integrations can also be used to send Configuration data to a system other than Commerce, and to redirect a user to page within that system.

In the context of a CPQ Cloud-Commerce integration, when the Add to Cart custom Product Family integration is created, an Add to Cart action will be placed in the sticky action bar of the Configurator. When the Add to Cart action is invoked by the Commerce customer, data relating to the configured product will be sent to a Commerce endpoint URL (which is defined in a CPQ Cloud Data Table) via a REST API call, and the customer will be taken to the Commerce pricing page.

**Note:** The Add to Cart custom Product Family integration/action implemented as a part of CPQ Cloud-Commerce integration is separate from the Add to Cart Process Invocation action on sites built on the Reference Application or a QuickStart vertical template. The Add to Cart Process Invocation action will be hidden as a part of CPQ Cloud-Commerce implementation. See the [CPQ to Oracle Commerce \(ATG\) Integration white paper](#) for more information.

## EDIT INTEGRATION PAGE CHANGES

### ***Custom Integration Option***

To enable custom Product Family integrations, a new **Custom** integration type has been added to each Product Family's **Edit Integration** page (Admin > Catalog Definition > List > Integrations > List).

Edit Integration		Product Family : abc
<b>Integration Information</b>		
*Name:	<input type="text"/>	This will be the name of the button to click to run this integration on the model configuration page.
*Variable Name:	<input type="text"/>	
Integration Type	<input checked="" type="radio"/> None <input type="radio"/> Http Post <input type="radio"/> Custom	

**Note:** When a custom Product Family integration is created, an integration action with the same name as the custom Product Family integration will be placed in the sticky action bar of the Configurator (and therefore will also be present in the Configurator embedded in an iframe within Commerce in a CPQ Cloud-Commerce integration).

On the **Edit Integration** page, when **Custom** is selected for **Integration Type**, **Action** and **Endpoint URL** options appear on the **Edit Integration** page.

<b>Action</b>	<input checked="" type="radio"/> None <input type="radio"/> Define Advanced Function	<input type="button" value="Define Function"/>
<b>Endpoint URL</b>	<input checked="" type="radio"/> Simple <input type="radio"/> Define Advanced Function	<input type="button" value="Define Function"/>

These options determine if the custom Product Family integration is active or inactive, what the success and error messages of the integration will be if the integration is active, and where the user will be redirected to when the integration action is invoked in the Configurator.

**Important:** In the table below, the **Endpoint URL** field refers to the where the user will be redirected to when the integration is invoked.

The Partner URL, which is added to a CPQ Cloud Data Table as a part of the CPQ Cloud-Commerce implementation, is the literal Commerce Endpoint URL where the REST payload will be sent when the integration is invoked.

The terms “Partner URL” and “Endpoint URL” cannot be used interchangeably when discussing custom Product Family integrations.

	Description	Options	Advanced Function Notes
<b>Action</b>	Depending on the option selected, determines whether or not the integration is active or inactive, and what the success and error messages will be for the integration.	<p><b>None:</b> The integration is inactive. Nothing is sent to the Partner URL when the integration action is invoked. If an Endpoint URL is defined, the destination redirect will still occur when the integration action is invoked.</p> <p><b>Define Advanced Function:</b> The integration is active and CPQ Cloud will send a REST payload to the partner URL when the integration action is invoked. The admin can define a BML function to determine the success and error messages of the integration.</p>	<p>When <b>Define Advanced Function</b> is selected for <b>Action</b>, clicking the <b>Define Function</b> button will open a BML editor with the following characteristics:</p> <ul style="list-style-type: none"> <li>• configXML (an XML file generated by CPQ Cloud based on current Configuration data) is the default input.</li> </ul> <p>Admins can access the current configXML file of a Configuration on the <b>Configuration XML</b> tab of the <b>Pipeline Viewer</b>.</p> <ul style="list-style-type: none"> <li>• The expected return type is String.</li> <li>• Expected return format:</li> </ul> <pre>Status=[Success/Error]~Message=[any string]</pre> <p>where [Success/Error] specifies whether or not the message is a success message or an error message, and where [any string] is the body of the success/error message.</p> <p><i>For example:</i></p> <pre>Return "Status=Success~Message=Working "; or Return "Status=Error~Message=Error in Web Service call"</pre>
<b>Endpoint URL</b>	The location where the user will be directed to after invoking the integration action in the Configurator.	<p><b>Simple:</b> The admin can define a static URL destination.</p> <p><b>Define Advanced Function:</b> The admin can define a BML function to determine the URL destination.</p>	<p>When <b>Define Advanced Function</b> is selected for <b>Endpoint URL</b>, clicking the <b>Define Function</b> button will open a BML editor with the following characteristics:</p> <ul style="list-style-type: none"> <li>• configXML (an XML file generated by CPQ Cloud based on current Configuration data) is the default input.</li> <li>• The expected return type is String.</li> <li>• The expected return format is a simple string, which will be a full URL.</li> </ul>

See the [CPQ to Oracle Commerce \(ATG\) Integration](#) white paper for complete setup steps for custom Product Family integrations as they relate to CPQ Cloud-Commerce integrations.

### Hide in Reconfiguration Option

A new **Hide in Reconfiguration** option has been added to each Product Family's **Edit Integration** page (Admin > Catalog Definition > List > Integrations > List).

Edit Integration		Product Family : abc
<b>Integration Information</b>		
*Name:	<input type="text"/>	This will be the name of the button to click to run this integration on the model configuration page.
*Variable Name:	<input type="text"/>	
Integration Type	<input type="radio"/> None <input type="radio"/> Http Post <input checked="" type="radio"/> Custom	
<b>Hide in Reconfiguration</b>	<input checked="" type="radio"/> No <input type="radio"/> Yes	This will enable or disable the display of the button on the product configuration page during reconfiguration.

When **No** is selected for **Hide in Reconfiguration**, the integration action (Add to Cart in CPQ Cloud-Commerce integrations) will be present during Configuration and reconfiguration of the product.

When **Yes** is selected for **Hide in Reconfiguration**, the integration action (Add to Cart in CPQ Cloud-Commerce integrations) will be hidden from the user during reconfiguration of the product.

#### Notes:

- If an advanced function was defined and saved for the **Action** of a custom Product Family integration, and then **None** is selected for the **Action** and the integration is applied/updated, the advanced function that was previously defined will be cleared.
- If an advanced function was defined and saved for the **Endpoint URL** of a custom Product Family integration, and then **Simple** is selected for the **Endpoint URL** and the integration is applied/updated, the advanced function that was previously defined will be cleared.

#### STEPS TO ENABLE

---

The CPQ Cloud-Commerce integration enhancements are automatically available on all 2015 R2 sites, but steps must be taken in both CPQ Cloud and Commerce to implement the enhancements. See the [CPQ to Oracle Commerce \(ATG\) Integration](#) page on My Oracle Support for a white paper on enabling these enhancements to the CPQ Cloud-Commerce integration.

#### TIPS AND CONSIDERATIONS

---

After the 2015 R2 CPQ Cloud-Commerce integration enhancements have been enabled, Commerce customers will still be able to request a quote for simple products, depending on the implementation.

#### KEY RESOURCES

---

[CPQ to Oracle Commerce \(ATG\) Integration](#)

---

## DECODEBASE64 BML FUNCTION

A new BML function, `decodebase64`, can now be added to BML functions. When a Base64-encoded string is input into the `decodebase64` function, the function decodes the string and returns its plain text format.

This function allows CPQ Cloud BML functions to interpret Base64-encoded strings that are sent by partner systems in integrations.

Decodebase64	
<b>Syntax</b>	<code>decodebase64 ("string")</code>
<b>Parameter</b>	<code>string</code>   A Base64-encoded string.
<b>Return Type</b>	String
<b>Return Description</b>	The plain text string format of the Base64-encoded string parameter.
<b>Example</b>	<i>Input:</i> <code>decodebase64 ("YWJj")</code> <i>Return:</i> <code>abc</code>

---

## STEPS TO ENABLE

The `decodebase64` BML function is automatically available on all 2015 R2 sites.

---

## WEB SERVICES V1.0 ACCESS MODIFICATIONS

Parts, Price Book Associations, and Data Tables SOAP API endpoints can now only be accessed by external systems with valid CPQ Cloud Host Company FullAccess user credentials. Previously, systems with any valid CPQ Cloud user credentials could access these endpoints.

---

## STEPS TO ENABLE

The Web Services v1.0 access modifications are automatically available on all 2015 R2 sites.

## PRE-UPGRADE CONSIDERATIONS

---

### KNOWN FUNCTIONALITY

#### CONFIGURATION DEPLOYMENT FAILURES

---

In some cases, having the **Recreate buyside cache after deployment** Configuration setting (Admin > Configuration Settings) set to **Yes** may cause Configuration deployments to fail. It is recommended that **Recreate buyside cache after deployment** is set to **No** on all sites.

#### EMAIL TEMPLATE EDITOR

---

The Email Template Editor is now known as the Email Designer.

When upgrading to 2015 R2 from a release older than 2015 R1, Email Templates that do not have any translations defined will become single-language templates with the site's base language.

Email Templates that have at least one translation defined will become multi-language templates with the site's base language as the default language.

#### SECURE URL ADDRESSES

---

Confirm that all references to your CPQ Cloud URL, such as in customizations or third-party tools, use `https://` in the URL.

#### TRANSLATION

---

For some system defined messages and components, some strings have been removed and others have been added. If strings have been previously translated, some strings will no longer appear and other strings will appear in English. The strings that appear in English are new, and need to be translated.

Most of these messages and components are on the admin side of CPQ Cloud, but you should review both your end-user and admin pages before deploying your updated installation to confirm that all strings appear in the desired language.

#### MIGRATION

---

When migrating from one site to another using the Migration Center, both sites must be on the same major release. Content may only be migrated across minor releases within the same major release. Migration across major releases cannot occur.

- "Major release" = A major product release, e.g. 2015 R2
- "Minor release" = A release update, e.g. 2015 R2 Update 1

#### RESOLVED KNOWN ISSUES

---

For information on bugs fixed in 2015 R2, refer to the 2015 R2 Resolved Known Issues document, available on [My Oracle Support](#) and the CPQ Cloud Online Help.

---

## FUNCTIONAL KNOWN ISSUES

Use the links below to review functional known issues in 2015 R2 and their workarounds:

- [Secure Attributes Lose Value in Mobile](#)
- [Secure Attribute Values only Available to First AJAX Rules Execution or Save](#)

---

## TRANSLATION STATUS

CPQ Cloud supports the consumption of both single and multi-byte character sets. Submit a service request on [My Oracle Support](#) to enable your site for a new language.

For the following languages, a translation of the CPQ Cloud user interface is available for both the platform and the reference application:

- Chinese (Simplified) [China]
- Chinese (Traditional) [Taiwan]
- Czech [Czech Republic]
- Danish [Denmark]
- Dutch [Netherlands]
- English
- Finnish [Finland]
- French
- French [Canada]
- German
- Hungarian [Hungary]
- Italian
- Japanese [Japan]
- Korean [South Korea]
- Norwegian (Bokmål) [Norway]
- Polish [Poland]
- Portuguese [Brazil]
- Romanian [Romania]
- Russian [Russia]
- Spanish (Worldwide)
- Swedish [Sweden]
- Turkish [Turkey]



## POST-UPGRADE CONSIDERATIONS

All test instances should be upgraded and tested on Oracle CPQ Cloud 2015 R2 before upgrading production instances.

---

### BROWSER SUPPORT

CPQ Cloud supports all browser versions that meet the criteria of the Oracle Software Web Browser Support Policy.

---

### SUPPORTED BROWSERS

#### Windows

- Major releases of **Google Chrome** upon general browser availability until Google no longer supports the version
- Major releases of **Mozilla Firefox** upon general browser availability until Mozilla no longer supports the version
- Major releases of **Internet Explorer/Microsoft Edge** within nine months of general browser availability until Microsoft no longer supports the version

#### Mac OS X

- Major releases of **Google Chrome** upon general browser availability until Google no longer supports the browser version
- Major releases of **Mozilla Firefox** upon general browser availability until Mozilla no longer supports the version
- Major releases of **Safari** within nine months of general browser availability until Apple no longer supports the version

#### Android

- Major releases of **Google Chrome** upon general browser availability until Google no longer supports the browser version

#### iOS

- Major releases of **Safari** within nine months of general browser availability until Apple no longer supports the browser version

If you experience issues using a supported browser version, open a ticket on [My Oracle Support](#) so that the issue can be resolved. If an issue does arise when using a supported browser, a certified browser version can be used until a fix is delivered. Certified browsers are selected based on current market share and are thoroughly tested to work with the current version's standard functionality.

## CERTIFIED BROWSERS

---

### Windows

- Google Chrome 47.x
- Mozilla Firefox 41.x
- Internet Explorer 11

### Mac OS X

- Google Chrome 47.x
- Mozilla Firefox 41.x

### Android

- *Operating System:* 4.4 Kit Kat
- *Browser:* Google Chrome 47.x
- *Screen resolution:* 2560 x 1600

### iOS

- *Operating System:* iOS 9.x
- *Browser:* Safari 9.x
- *Screen resolution:* 2048 x 1536

**Note:** Sites created with additional JavaScript, alternate CSS, or other custom functionality may no longer be compatible with our supported browsers. Open a ticket on [My Oracle Support](#) to determine if workarounds and minor fixes are available. Add-on work may be required to support customizations.

---

## SALESFORCE MANAGED PACKAGE SUPPORT

CPQ Cloud will no longer release updates to Salesforce integration Managed Packages prior to v5.0. With the release of 2015 R2, only Managed Packages v5.x and v6.x are officially supported.

Although Salesforce integrations that use a Managed Package prior to v5.0 are still expected to function, new issues that arise in these versions will not be addressed by CPQ Cloud.

---

## TLS SUPPORT FOR SALESFORCE INTEGRATIONS

As of Salesforce Winter '16, Salesforce supports only TLS 1.1 and TLS 1.2.

**Note:** Salesforce's deactivation of TLS 1.0 support is tentatively scheduled for the first or second quarter of 2016 (see the Supported Browsers for Salesforce Classic section of [Winter '16 Release Notes](#) for more information).

In light of this change, in addition to TLS 1.0, CPQ Cloud 2015 R2 and subsequent versions will support TLS 1.1 and TLS 1.2. Additionally, 2015 R1 Update 3 and 2014 R2 Update 8 will be released by January 1, 2016 and will enable TLS 1.1 and TLS 1.2 support for 2015 R1 and 2014 R2, respectively. Customers on

2015 R1 that require TLS 1.1 or 1.2 support should update to at least 2015 R1 Update 3. Customers on 2014 R2 that require TLS 1.1 or 1.2 support should update to at least 2014 R2 Update 8.

**Important:** If a customer with an existing integration with Salesforce fails to upgrade to a CPQ Cloud version that supports TLS 1.1 and TLS 1.2, CPQ Cloud will no longer be able to send/receive data to/from Salesforce upon Salesforce's deactivation of TLS 1.0 support.

**Important:** Updates to enable TLS 1.1 and TLS 1.2 support will not be released for CPQ Cloud versions prior to 2014 R2. Customers on CPQ Cloud versions prior to 2014 R2 must update to at least the most recent 2014 R2 update available on January 1, 2016 in order to support TLS 1.1 and TLS 1.2.

---

## TRAINING

Please refer to release documentation for all versions between your current version and the version that you are upgrading to in order to see all new functionality, resolved known issues, and functional known issues.

It is strongly recommended that users refer to the help documentation to become familiar with the new features introduced in Oracle CPQ Cloud 2015 R2. For additional help, see [My Oracle Support](#).

Any information not explicitly mentioned in this document as being supported by the software can be verified against the product help for Oracle CPQ Cloud 2015 R2 and/or with the Oracle CPQ Cloud Consulting team.

---

## ADDITIONAL INFORMATION

For more information on Oracle CPQ Cloud, visit the [Oracle CPQ Cloud documentation site](#).

---

## DISCLAIMER

The details in this document are provided for high-level informational purposes only and are not intended to function as a specification or to replace the Product Help and/or User Guide. Upgrading to a subsequent release may require Commerce Processes, Configuration, and/or global function settings to be re-deployed. Modifications to integration XSL files and/or APIs also may be required.

---

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.

**Integrated Cloud Applications & Platform Services**