Oracle[®] Public Cloud Machine

Using the Oracle Mapper Release 16.3.1 E67464-03

October 2016

This guide describes how to use the mapper to map source data structures to target data structures.

Oracle Public Cloud Machine Using the Oracle Mapper, Release 16.3.1

E67464-03

Copyright © 2015, 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Mark Kennedy

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Pr	eface	v
	Audience	v
	Related Resources	v
	Conventions	v
1	Getting Started with the Mapper	
	About Mappings	1-1
	About Mapping Data Between Applications	1-2
	Building Expressions with the Mapping Builder	1-3
2	Mapping Data	
	Creating Mappings	2-1
	Repeating a Target Element to Map to Different Sources	2-2
	Automatically Creating for-each Statements	2-4
	Deleting Individual Mappings	2-7
	Finding Data Fields	2-7
	Browsing and Filtering the Source or Target Data Structure	2-8
	Mapping Multiple Source Structures to a Target Structure	2-10
	Testing Your Mappings	2-11
3	Building Expressions	
	Building Mapping Statements	3-1
	Repeating Elements with the for-each Statement	3-2
	Creating Conditional Mappings	3-4
	Using XPath Functions	3-5
	Referencing Lookups	3-6
	Deleting Mapping Statements	3-8

Preface

Using the Oracle Mapper describes how to use the mapper to map source data structures to target data structures.

Topics:

- Audience
- Related Resources
- Conventions

Audience

Using the Oracle Mapper is intended for users who want to use the mapper to map source data structures to target data structures.

Related Resources

For more information, see these Oracle resources:

- Oracle Public Cloud Machine documentation in the Oracle Help Center: http://docs.oracle.com
- Using Oracle Integration Cloud Service

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with the Mapper

Review the following topics to learn how the mapper works to map source data structures to target data structures.

Topics

- About Mappings
- About Mapping Data Between Applications
- Building Expressions with the Mapping Builder

About Mappings

One of the key tasks to any integration is defining how data is transferred, or *mapped*, between two applications.

In most cases, the messages you want to transfer between the applications in an integration have different data structures. A visual mapper enables you to map fields between applications by dragging source fields onto target fields. When you open the mapper for a request or response message in an integration, the data structures are automatically populated with the information pulled from the source and target connections. You can expand and the load data structure levels on demand to display additional levels. There is no limit on the levels of display.

▲ Exit Mapper Map: processOrganization to Get										
View 🔻	Filter 🔄 📄 Detach	Map	ŧ.		View 🔻	Filter 📴	Detach			
Source	Find	٩		🛷 Mappings	Target		Find Q	Mapping		
.⊿ <> *pro	ocessOrganization									
\diamond	*changeOperation					🔤 id				
# 7 5	organizationParty					<> Looku	pName			
	PartyNumber		Ξ			<> Create	<> CreatedTime			
	<> Partyld			Drag and drop source to		<> Updat				
	<> PartyType	•		target to create a mapping.		♦ <> Addre	sses			
	PartyName			Click a checkmark on		er				
	LastUpdatedBy			source or target to see mappings.		► <> CRMN	Adules			
	ValidatedFlag					♦ <> Organ	izationCustomFields			
	LastUpdateLogin					<> Extern	alReference			
	CreationDate					FileAt	tachments			
	RequestId					Indust	лу			
	<> LastUpdateDate					<>> Name	dReadOnlyID extends Industry			
	CreatedBy					<> Login				
	OrigSystemReference					<> Name	l.			
	 HQBranchIndicator 					<> Name				
	OUNSNumberC					<> NewP	assword			

The maps you create are called transformation maps, and use the eXtensible Stylesheet Language (XSL) to describe the data mappings, which lets you perform complex data manipulation and transformation. A standard set of XPath functions are provided for you to define how data is modified when moving from one application to another. A specialized function is also provided for you to reference lookups directly from the mapper.

The mapper supports both qualified and unqualified schemas (that is, schemas without elementFormDefault="qualified"). Elements and attributes with and without namespace prefixes are also supported.

About Mapping Data Between Applications

Once you create an integration and have the source and target connections in place, you can define how data is mapped between the two data structures.

The mapper appears with the source data structure on the left and the target data structure on the right.

Kexit Mapper Map: processOrganization to Get												
View 🔻	Filter 🐺 📄 Detach	Map 📩		View 🔻	Filter 🕞	Detach						
Source	Find	Q	🛷 Mappings	Target		Find Q	Mapping					
.⊿ <> *pro	cessOrganization	•										
\diamond	*changeOperation	•			id	l.						
	organizationParty				<> Looku	pName						
	<> PartyNumber	_ E	E		<> Create	edTime						
	Partyld PartyType PartyName		Drag and drop source to target to create a mapping. Click a checkmark on		<> Updat	tedTime						
					♦ <> Addre	sses						
					I ← Banne	er						
	LastUpdatedBy		source or target to see mappings.		► <> CRMM	Modules						
	ValidatedFlag				I <> Organ	nizationCustomFields						
	LastUpdateLogin				<> Extern	nalReference						
	CreationDate				I ← FileAt	ttachments						
	<> RequestId				Indust	try						
	↔ LastUpdateDate				Is In the second se	dReadOnlyID extends Industry						
					<> Login							
	OrigSystemReference				<> Name	1						
	HQBranchIndicator				<> Name							
♦ DUNSNumberC					<> NewP	assword						

1. To map fields directly, click a field in the source and drag it to the corresponding field in the target.

The name of the source field appears in the target **Mapping** column, and a green check mark icon appears next to both fields. The most recently mapped fields are connected by a green line. Click the green check mark of other sources and targets to see their current mappings.

Exit Ma	pper	I	Map: processOrganizatio	n to Get				
View 🔻	Filter 🐺 📄 Detach	Мар 🟥		View 🔻	Filter 🕞	Detach		
Source	Find	٩	🛷 Mappings	Target		Find	Q	Mapping
	oessOrganization	· · ·						
\diamond	*changeOperation			0	🔤 id			Partyld
4 \ \\	organizationParty				<> Looku	pName		
	<> PartyNumber	— Е			<> Create	dTime		
	<> Partyld			<> Updat	edTime			
	<> PartyType			♦ <> Addres	sses			
	ArtyName				I ← Banne	er		
	<> LastUpdatedBy			•	♦ <> CRMN	lodules		

- **2.** To use XPath functions in your mapping, see Using XPath Functions for instructions.
- 3. To use lookups in your mapping, see Referencing Lookups for instructions.
- 4. When you are done mapping data, click Save, then click Exit Mapper.
- 5. On the toolbar, click Save.
- **6.** Click the map icon between the response operations, and repeat the above steps to map the response. Make sure to save the integration when you are done.

Building Expressions with the Mapping Builder

Use the Mapping Builder to create and modify advanced mappings. For example, you can loop through repeating elements, create conditional statements, use XPath functions, and reference lookups. The Mapping Builder is displayed when you click a field in the target data structure of the mapper.

Mapping	Builder							×
Source				Mapping	Help	🕍 Save	Revert	T Actions
View 🔻	Filter 🕎	Detach	Map 📩	Target Element: //Get/Organization/NumberOfEmployees				
		Find	Q	Statement				
				⊿ <> <mo_v1_2:numberofemployees></mo_v1_2:numberofemployees>				
_al <> pr	ocess		- Â	Drag and Drop or Type here				
A 7	S Organization							
	▶ <> ID							
	<> Lookup!	Name	🥥 😑					
	<> Created	Time						
	<> Updated	Time						
	Address	es						
	Sanner							
	CRMMo	dules						
	> organiza	ationCustomField	Is					
	FileAttac	hments						
	Industry							
	<> Login							
	<> Name		-					
Mapp	ing Compor	ents						
▶ Variat	oles							
								Close

For more information, see Building Expressions.

2

Mapping Data

Use the intelligent data mapper to drag fields from the source structure to the target structure to map elements between the two.

Topics

- Creating Mappings
- Repeating a Target Element to Map to Different Sources
- Automatically Creating for-each Statements
- Deleting Individual Mappings
- Finding Data Fields
- Browsing and Filtering the Source or Target Data Structure
- Mapping Multiple Source Structures to a Target Structure
- Testing Your Mappings

Creating Mappings

You can map fields directly from the source data structure to the target data structure in the mapper.

To create mappings:

- 1. In the middle of the integration, click the **Mapper** icon for the request, response, or fault map to edit.
- 2. Click Create.

Click	Below
to Ci	reate
Ma	ap
	رام Create

- **3.** To map fields directly, perform one of the following steps:
 - **a.** Click a field in the source and drag it to the corresponding field in the target.

b. Click the source and the target fields, and then click **Map+**.



The name of the source field appears in the target Mapping column, and a green check mark icon appears next to both fields. The most recently mapped fields are connected by a green line. Click the green check mark of other sources and targets to see their current mappings. You cannot drag and drop onto a target that is already mapped.

K Exit Mapper			lap:	process To Get							
View 🔻	Filter 🖳	Detach	Map :	5		View 🔻	Filter 🖙	Detach			
Source		Find	٩,		📌 Mappings	Target			Find	Q,	Mapping
⊿ <> process			^			Get					
	Organization	n					₹> Organiza	tion			
)	> <> ID						▶ <> ID				
<> LookupName 🔗						- O	upName			LookupName	
	<> Created	dTime	0	-		0	<> Crea	itedTime			CreatedTime

Note: In the mapping summary, the full names of extended elements are not displayed.

Repeating a Target Element to Map to Different Sources

You can repeat a target element in the mapper. This enables you to map different sources to the same target element. Elements defined in the target schema with the maxOccurs attribute set to a value greater than one can be repeated.

To repeat a target element to map to different sources:

1. In the target data structure, right-click the element to repeat, and select **Repeat Element**. Elements that can be repeated are identified by a special icon with two bars to the left of the name. When you place your cursor over these elements, the words **Repeating Element** are displayed.



The element is repeated and displayed below the existing element. Elements that are repeated show the count (for example, **(1 of 2)** for the existing element and **(2 of 2)** for the repeated element. You can repeat an element multiple times.

2. Expand the existing and repeated elements to see that the attributes in each element are repeated.

Target	Find
A <> Process	SalesOrderFulfillmentEBM
# <> EBN	AHeader
> > o	EBMID
● → 5	EBMName (1 of 2)
● ► 5	EBMName (2 of 2)
4 2	EBOName (1 of 2)
	I languageCode
	InguageLocaleCode
4 8	EBOName (2 of 2)
	EE languageCode
	I languageLocaleCode

- **3.** Click the child attribute of the existing element (for this example, named **languageCode** under **EBOName (1 of 2)** to access the Mapping Builder.
- **4.** Replace **Drag and Drop or Type here** with a value (for this example, en is entered for English).



- 5. Click Save, then click Close.
- **6.** Click the child attribute of the repeated element (for this example, named **languageCode** under **EBOName (2 of 2)** to access the Mapping Builder.
- **7.** Replace **Drag and Drop or Type here** with a value (for this example, fr is entered for French).
- 8. Click Save, then click Close.

The assigned values are displayed in the Mapper column.

A TS EBOName (1 of 3)	
III languageCode	en
IanguageLocaleCode	
A 75 EBOName (2 of 3)	
E languageCode	fr
IanguageLocaleCode	

Note: If you create a repeatable element in which you do not do any mapping, then click **Save** and **Exit Mapper**, the empty element is not saved.

For more information about repeatable elements, see Automatically Creating for-each Statements.

Automatically Creating for-each Statements

You can automatically create for-each statements when mapping between repeatable source and target elements in the mapper.

To automatically create for-each statements:

 In the Source panel, identify the repeatable source and target elements to which to map. A repeatable element is identified by a special icon with two bars to the left of the name. When you place your cursor over these elements, the words Repeating Element are displayed. For this example, there are parent and child book and title repeatable elements in the Source panel and parent and child item and name repeatable elements in the Target panel.

Source	Find	Q		👉 Mappings	Target	
⊿ <> books			-			<> items
a 🖏 book						⊯ ₹5 item
► ₹5 ti	tie					► ₹5 name
> authors					6	> models
<> p	ublicationDate					<> madeln

2. In the **Source** panel, map the *child* repeatable element to the *child* target repeatable element (for this example, the source repeatable element **title** is mapped to the target repeatable element **name**).

Source	Find	9	📌 Mappings	Target	Find	9	Mapping
		<u> </u>		a <> items			
a 🖏 book				🔵 🔺 🐯 item			
► ₹5 title		0			1e		for-each(title), title
► <> auth	nors			●	dels		200
<> pub	licationDate			mad	deln		

In the **Mapping** column of the **Target** panel, a **for-each** statement is automatically created with a selectable value (for this example, named **title**).

3. Click the **for-each** statement to access the Mapping Builder.

A **for-each** statement with an absolute path is displayed.



- 4. Click Close to exit the Mapping Builder.
- **5.** In the **Source** panel, identify the *parent* repeatable source and target elements to which to map (for this example, **book** and **item**).
- **6.** In the source data structure, map the *parent* repeatable element to the *parent* target repeatable element (for this example, the source repeatable element **book** is mapped to the target repeatable element **item**).

Source	Find	9		📌 Mappings	Target	Find	9,	Mapping
⊿ <> books			<u>^</u>		🦳 л <> items			
a 🖏 book		0-			-S item			for each(book)
► ₹3 title		0			📀 🕨 🏹 name			for-each(title), title

The mapper creates a second **for-each** statement to loop through the **book** element and place the mapping into the **item** element. This statement does not include a value to select because parent elements do not typically contain attributes to map. **7.** Click the **for-each** function to access the Mapping Builder. The **for-each** statement is displayed. Note that the absolute path of the mapping is converted to a relative path.

Mapping	Builder			
Source				Mapping
View 🔻	Filter 🕎	Detach	Мар 📥	Target Element: /items/item/name
		-	0	Statement
		Find	4	🔺 🥒 🐼 <xsl:for-each select="ns0:title"></xsl:for-each>
.⊿ <> bo	oks		4	stns:name>
A 3	book		0	
1	title		0	
I	<> authors			-
	<> publicat	ionDate		

- 8. Click Close to exit the Mapping Builder.
- **9.** Map the source attribute to the target attribute (for this example, **lang** is mapped to **lang**).

Source	Find	Q,	🖨 Mappings	Target	Find Q	Mapping
⊿ <> books						
a 🖏 book		0		🤡 🔺 🖏 item		for-each(book)
⊿ ₹5 title		0		📀 🔺 🖏 name		for-each(title), title
	lang	O			na	@lang
<>> auti	hors			> woodel	10	

- **10.** In the **Mapping** column, click the attribute to access the Mapping Builder.
- **11.** In the upper right corner, select **Actions** > **Show Content** to display the execution code.

The first **for-each** selects the books and the second **for-each** selects the book titles.

<xsl:for-each select = '/ns0:books/ns0:book'>

<xsl:for-each select = 'ns0:title'>

12. Click **Close** to exit the Mapping Builder.

13. Click Test.

14. Enter the source payload and click **Execute**. For this example, the payload includes two book titles. Each title is displayed in both English and Spanish.

The titles in English and Spanish for both books are displayed in the Output panel..

Test Map	Help Generate Inputs Clear Exercise
Input: books Sparam_lib Sparam_people	Output: items
<pre><?xmi version = '1.0' encoding = 'UTF-8'?> <book s.xmins="http://ww.books.org'> <book lang="engr"> <title lang="spir>Cinderella</title> <ti>title lang=" spir="">Cencienta</title> <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta <pre>cencienta</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></book></pre>	<pre><hr/> <hr/> <</pre>

For more information about repeatable elements, see Repeating a Target Element to Map to Different Sources.

Deleting Individual Mappings

You can delete individual source-to-target mappings with the mapper.

To delete individual mappings:

Note: You cannot delete a mapping by right-clicking the line between the source and target.

1. To remove a mapping, right-click the row of the target to which to remove and select **Delete Mapping**.

View View Filter	Detach I	Vlap 📩		Filter 🐺	Detach			
Source	Find	Q,	F Mappings Target		Target		Find	
▲ <> GetQuote				🔵 🔺 <> GetQuote				
<> symbol		O -		-0	<> symbol			
				X Delete Ma		Mapping		
						Reneat F	lement	

2. Click Yes when prompted to confirm.

If you attempt to delete a parent mapping that includes child mappings, you are prompted to confirm that you want to delete all the child mappings inside the parent mapping.

Finding Data Fields

The mapper displays the source data structure on the left and the target data structure on the right. You can search for specific field names in either structure.

To find data fields:

1. Type the full or partial name in the source or target data structure, and click the **Find** icon.



The tree scrolls to the first match.

2. Click the Find icon again to scroll to the next match.

Browsing and Filtering the Source or Target Data Structure

The mapper displays the source data structure on the left and the target data structure on the right. You can filter the display of either structure.

To browse and filter the source or target data structure:

- 1. To expand or collapse parent nodes one level at a time:
 - **a.** Right-click the parent node and select **Expand** or **Collapse**.
 - **b.** Click the **Expand** or **Collapse** icon to the left of the parent node.
- **2.** To expand or collapse all nodes under a parent, right-click a row and select **Expand** or **Collapse All Below**.
- **3.** To promote a node to the top, right-click a row and select **Show as Top**. To return to the original view, select the **Up** arrow in the first row, then select the data structure hierarchy to access.



- **4.** To filter the source side, click **Filter** to specify map filtering options based on the following criteria, then click **Apply**. You can select one option from each section. When the **Filter** icon is displayed in blue, this indicates that the filter has been modified from its default settings and is active. To reset the filter option to its default values, click **Reset All**.
 - Current field status (show all fields, mapped fields (applies to target nodes to which a source, variable, or function has been mapped), or unmapped fields). If you select **Not Mapped**, it shows all parent nodes of the unmapped elements even if the parent nodes are only mapped.
 - Whether fields have special annotations (all fields, standard fields delivered out-of-the-box as part of a prebuilt Oracle integration, or custom fields you created in a prebuilt Oracle integration that was edited in customization mode).
 - The source data structures in the integration. This option is only displayed if there are multiple source data structures in this mapper.

Filter	Reset All Apply
Is it Mapped?	
Filter based on whether fields a	are mapped or not.
Mapped	
Not Mapped	
How are the Fields Def	fined?
Filter based on whether the field	d has special annotation.
Standard Fields	
Custom Fields	
Which Source Schema	1?
Filter based on one of the source All Source Schemas	ce schemas
process	
\$RequestEnrichmentAppl	licationObject

5. On the target side, click **Filter** to specify map filtering options based on the following criteria, then click **Apply**. You can select one option from each section. When the **Filter** icon is displayed in blue, this indicates that the filter has been modified from its default settings and is active. To reset the filter option to its default values, click **Reset All**

- Current field status (show all fields, mapped fields (applies to target nodes to which a source, variable, or function has been mapped), or unmapped fields). If you select **Not Mapped**, it shows all parent nodes of the unmapped elements even if the parent nodes are only mapped).
- Whether fields have special annotations (all fields, standard fields delivered out-of-the-box as part of a prebuilt Oracle integration, or custom fields you created in a prebuilt Oracle integration that was edited in customization mode).
- The origin of mappings (all, standard, or custom).
- Whether the mappings are valid.

Filter	Reset All Apply
Is it Mapped?	
Filter based on whether fields are mappe All Fields	d or not.
Mapped	
Not Mapped	
How are the Fields Defined?	
Filter based on whether the field has spe	cial annotation.
All Field Types	
Standard Fields	
Custom Fields	
Where did the Mappings Origi	inate?
Filter based on whether the mappings or	ginated standard or custom.
All Origins	
Standard Mappings	
Custom Mappings	
Are the Mappings Valid?	
Filter based on results of validation.	
All Validation Results	
Errors Found	
Warnings Found	
No Issues Found	

Mapping Multiple Source Structures to a Target Structure

You can map fields from multiple source structures to a single target structure in certain parts of integrations (for example, integrations in which message enrichment points have been added or integrations with a response mapping). This action applies to the creation of new maps.

To map multiple source structures to a target structure:

- 1. In the mapper, note that two source structures are displayed:
 - The initial request mapping source (process)

• The secondary request enrichment mapping source (**\$RequestEnrichmentApplicationObject**)

View View Filter			View 🔻	Filter 🕎	Detach			
Source Find	Q,	📌 Mappings	Target			Find	٩	Mapping
⊿ <> process				Get				
Organization				😽 Organiza	tion			
▲ <> \$RequestEnrichmentApplicationObject				▶ <> ID				
⊿ <> GetResponse				<> Look	kupName			
► ₹ Organization				<> Crea	itedTime			
		Drag and drop source to target to create a mapping.		<> Upda	atedTime			

2. Expand the initial source data structure and drag appropriate source fields to target fields.

K Exit Mapper		Ma	p: process To Get						
View 🔻 Filter 🖫	Detach	Map 📩		View 🔻	Filter 🕞	Detach			
Source	Find	٩	🛷 Mappings	Target			Find	٩	Mapping
▲ <> process					Get				
🔺 🐺 Organizatio	n			• 4	₹3 Organiza	ition			
					▶ <> ID				
id ≊⊠				0	<> Loo	kupName			LookupName
<> Lookup	Name	Ø =			<> Crea	atedTime			

Expand the secondary source data structure and drag appropriate source fields to target fields.

1										
√ ³ View ▼	Filter 🔄	Detach	Мар 📩		View v	Filter 🕞	Detach			
Source		Find	٩	📌 Mappings	Target			Find	Q,	Mapping
	<> Parent	,	^			▶ <> ID				
	SalesS	Settings			0	<> Look	kupName			LookupName
	<> Service	Settings			ø	<> Crea	atedTime			CreatedTime
	<> Source	1		/	•	<> Upd	atedTime			
	<> ValidNu	ullFields				► <> Addr	esses			
	<> Named	dReadOnlyID exte	ends Industry			Is a state	ner			
	tequestEnrich	nmentApplication	Object	• /		▶ <> CRN	Modules			
4.0	GetRespon	ise				► <> Orga	anizationCustom	Fields		
	🖌 🐺 Organi	zation	=			► <> File/	Attachments			
	▶ <> ID			/		▶ <> Indu	stry			
	<> Lo	okupName				<> Logi	n			
	<> Cr	eatedTime	•			<> Nam	ie			

- 3. To test the mappings, see Testing Your Mappings.
- 4. When complete, click Save, then click Exit Mapper.

Testing Your Mappings

Once you complete designing your mappings, you can test it by entering sample content of the message to process in the mapping tester. When you execute the test, the output is generated from the sample content.

To test a mapping:

1. In the mapper toolbar, click Test.

The mapping tester appears. The names of the source and target data structures for your mapping are displayed at the top. If your mapping includes multiple source data structures, both names are displayed. The primary source is displayed first,

followed by the secondary source. Two instances are also generated and displayed for repeating nodes.

*					×
Test Map		Help	Generate Inputs	Clear	Execute
Input: process	Output: Create				
Please enter the input payload here or press the Generate Inputs button to automatically create all payloads necessary. Then press the Execute button to see results in the output panel.					

2. In the **Input** panel, enter the payload you want to test. You can manually enter the payload, copy and paste the payload, or click **Generate Inputs** to automatically generate the payload.

fest Map	-		Help Generate Input
Input: process	<pre><?ml\ursion = '1 0' encodion = 1 ITE.875</pre></pre>	Output: Create	
	The set of	2	
	The second secon		

If your mapping includes multiple source data structures, payloads for both sources can be generated and displayed by clicking the source name.

Test Map		
Input process	\$RequestEnrichmentApplicationObject	Output: Get
<br <n <</n 	xml version = '1.0' encoding = 'UTF-8'?> s0:GetResponse xmlns:ns1="un:base.ws.rightnow.com/v1_2" xmlns:ns2="un:objects.v rns0:Organization> <ns1:did='id345''> <ns1:lookupname>LookupName339</ns1:lookupname> <ns1:createdtime>2015-07-04T12:19:32.583 <ns2:addresses> <ns2:addresses> <ns2:country> <ns2:country> <ns2:country> <ns1:name>Name350</ns1:name> <ns2:tbitedofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns2:stateofprovince> <ns1:name>Name352</ns1:name> <ns1:name>Name352</ns1:name> <ns1:name>Name351</ns1:name> <ns1:name>Name361</ns1:name></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:stateofprovince></ns2:tbitedofprovince></ns2:country></ns2:country></ns2:country></ns2:addresses></ns2:addresses></ns1:createdtime></ns1:did='id345''>	
•	4	***

Note: If the payload is very large, it is not automatically generated and you receive the following error message:

Payload could not be generated for the ''\$SourceApplicationObject'' schema due to excessive size and a lack of system memory

3. Expand the display of the input payload by dragging the splitter in the middle of the page to the right. You can also click the splitter to toggle between the input and output payloads.

nput: proce	55		Output: Create
	<pre><?xml version = '1.0' encoding = 'UTF-8'?> <pre>vprocess xmlns="http://xmlns.oracle.com/cloud/adapter/osc/AccountCreatedService_REQUES' <account> <pre>vPartyld xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> <partylumber xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> <pre>vPartylumber xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> <pre>vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> <pre>vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> </pre>vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> </pre>vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> </pre>vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> </partylumber></pre>vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> </account></pre>vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> </pre> vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/"> vSourceSystem xmlns="http://xmlns.oracle.com/apps/crmCommon/salesParties/accountService/">	4	÷

- 4. Scroll through the input payload and note the following details:
 - Unbounded, repeating elements are displayed multiple times.
 - Schemas of up to 20 levels in depth can be displayed.

- Random values are automatically generated for payload elements. Based on the data type of the element, the correct values (for example, numerical or string values) are generated.
- You can manually edit the randomly-generated values, as necessary.
- 5. Click Execute to generate results in the Output panel.
- **6.** Review the results in the **Output** panel to ensure that your input payload was processed correctly.



- **7.** Test your mapping and, as necessary, return to the mapper to make mapping changes and to the Mapping Builder to make mapping logic changes, such as changing the XSL elements or functions used.
- 8. To clear the **Input** and **Output** panels, click **Clear**.
- **9.** When testing is complete, click **Close**.

Building Expressions

Use the Mapping Builder to create and modify advanced mappings. For example, you can loop through repeating elements, create conditional statements, use XPath functions, and reference lookups.

Topics

- Building Mapping Statements
- Repeating Elements with the for-each Statement
- Conditional Mappings
- Using XPath Functions
- Referencing Lookups
- Deleting Mapping Statements

Building Mapping Statements

This section describes how to build mapping statements.

To build mapping statements:

- **1.** Drag a source to a target to create a mapping.
- 2. Click the target node to launch the Mapping Builder.

If the target is not mapped, you see two statements: the target tag and a new empty statement with the text **Drag and Drop or Type here...**.

3. Drag a source element onto the empty row.

Mapping Builder								
Source					Mapping			
View 🔻	Filter 🔄	Detach	Map 🖠	5	Target Element: /Get/Organization/ID			
		Find	0		Statement			
		11114			<rnb_v1_2:id></rnb_v1_2:id>			
lal <> pr	ocess				In Crag and Drop or Type here			
	Sorganization	ı			13			
	▶ <> I <mark>D</mark>							

The value-of mapping statement appears.

Mapping Builder										
Source					Mapping					
View 🔻	Filter 🔄	Detach	Мар	5	Target Element: /Get/Organization/ID					
		Find	0		Statement					
		T IIIG	~		✓ <rnb_v1_2:id></rnb_v1_2:id>					
_al <> pr	A <> process			🖉 🔮 <xsl:value-of select="/nsmpr2:process/nsmpr2:Organization/rnb_v1_2:ID"></xsl:value-of>						
∡ → Organization					2					
▶ <> ID										

4. Click the Edit icon, or right click and select Edit.

The statement expands to show the select and source statements.

5. Change the mapping as necessary by dragging a different source element onto the source statement.

Mapping Builder									
Mapping									
Target Element: /Get/Organization/ID									
Statement									
▲ <> <mb_v1_2:id></mb_v1_2:id>									
🔺 🖉 🖉 <xsl:value-of select="/nsmpr2:process/nsmpr2:Organization/mb_v1_2:ID"></xsl:value-of>									
⊿ ໝ⊒ select									
<> /nsmpr2:process/nsmpr2:Organization/rnb_v1_2:ID									

6. Click Save.

Repeating Elements with the for-each Statement

The for-each statement allows you to loop through repeating elements.

To repeat elements with the for-each statement:

- **1.** Drag a source to a target to create a simple mapping.
- 2. Click the target node to launch the Mapping Builder.
- 3. Click Mapping Components.

You can either find or browse for the function.

- To find the function, type for-each in the Find field and click Find.
- To browse for the function, expand the XSL Elements folder and locate foreach.

Mapping Builder						
Source Mapping Components					Mapping	
					Target Element: /Get/Organization/ID	
		Map 📩		State	ement	
	Find	Q		4	<> <rnb_v1_2:id> Drag and Drop or Type here</rnb_v1_2:id>	
Functions			1			
Operators						
🔺 🚞 XSL Elements						
attribute name="	xsi:nil'					
🕨 🐼 choose			4			
🕨 🐼 for-each			1			
🕨 💽 if 🛛 🔓						
otherwise						
► 🐼 text						
🕨 🐼 value-of						
🕨 🐼 when						

4. Drag the **for-each** function onto the target tag statement.

The **for-each** statement appears as a parent of the target tag. A new empty row is created with the text **Drag and Drop or Type here...**

Mapping Builder	
Source	Mapping
Mapping Components	Target Element: /Get/Organization/ID
Мар 📩	Statement
Find Q	<mb_v1_2:id> <p< th=""></p<></mb_v1_2:id>
Functions	⊿ ≊ select
Operators	Drag and Drop or Type here for 'select'.
A 💼 XSL Elements	
attribute name='xsi:nil'	
▶ 🔿 choose	
🕨 💿 for-each	

5. Click Source.

•

6. Drag a repeating source element onto the empty row.

This uses the source element as the basis for the loop.

7. Click Save.

Creating Conditional Mappings

The if and choose statements are two ways to create conditions. If statements allow you to specify a single condition. Choose/when/otherwise statements allow you to specify multiple conditions, similar to if/then/else.

To create conditional mapping:

- 1. Drag a source to a target to create a mapping.
- 2. Click the target node to launch the Mapping Builder.
- 3. Click Mapping Components.

You can either find or browse for the function.

- To find the function, type if or choose in the Find field, and click Find.
- To browse for the function, expand the **XSL Elements** folder and locate **if** or **choose**.

Mapping Builder					
Source Mapping Components			Mapping Target Element: /Get/Organization/ID		
	Map 📩	Sta	atement		
Find	٩	4	<> <rnb_v1_2:id> Drag and Drop or Type here</rnb_v1_2:id>		
Functions					
Operators					
🔺 🚞 XSL Elements					
attribute name='xsi:nil'					
🕨 🗠 choose		•			
🕨 🐼 for-each					
🕨 🖸 if 🛛 😓					
otherwise					
🕨 🐼 text					
value-of					
🕨 🐼 when					

4. Drag the if or choose function onto the target tag statement.

The statement appears as a parent of the target tag. A new empty row is created with the text **Drag and Drop or Type here...**

- 5. Click Source. Then do one of the following:
 - Drag a source element onto the empty row to use a source element as the basis for the test condition.
 - Select an XPath function as the basis for the test condition.
 - Use a literal as the basis for the test condition by typing in the empty row.

- 6. For choose, you may specify additional when and otherwise conditions.
 - **a.** Right-click the **choose** statement and select **Insert Child**.

A new empty row is displayed with the text Drag and Drop or Type here...

- b. Drag and drop when or otherwise from the Mapping Components list.
- 7. Click Save.

Using XPath Functions

You can use a variety of XPath functions in your data mappings to transform the data you transfer between applications.

To use an XPath function:

- 1. Drag a source to a target to create a mapping.
- 2. Click the target node to launch the Mapping Builder.
- 3. Click Mapping Components.

You can either find or browse for the function:

- To find the function, type the full or partial name of the function in the **Find** field, and click the **Find** icon.
- To browse for the function, expand the **Functions** folder and subcategory folders to locate the function.

Mapping Builder							
Source					Mapping		
Mapping Components		Target Element: /Get/Organization/ID					
	Map 🛔				Statement		
	Find	٩		4	<> <rnb_v1_2:id> Drag and Drop or Type here</rnb_v1_2:id>		
🔺 🚞 Functions							
Advanced							
▶ f_X generate-id							
🕨 🖬 Boolean			=				
Conversion	Conversion						
Date							
F ICS							
🕨 🛅 Mathematical							
🔺 💼 Node-set							
f_{X} current							
<i>f</i> _X last							
) $f_{\!X}$ local-name	N		-				
	15						

4. Drag the function onto the mapping statement.

If the function has a single parameter, it automatically inserts the existing mapping expression into the parameter.

If the function has multiple parameters, it prompts you to select which parameter into which to insert the existing mapping expression.

- **5.** Select a parameter or **None** if you want to remove the existing mapping expression, and click **OK**.
- 6. Click Save and Close.

Referencing Lookups

A special lookup function in the mapper enables you to call a lookup from a mapping to determine the value to populate into a field when transferring data between applications.

To reference a lookup from a data mapping:

- 1. Drag the source to a target to create a simple mapping.
- **2.** Click the target node to launch the Mapping Builder.
- 3. In the Mapping Builder, click Mapping Components.
- 4. Type lookupValue in the Find field, and click Find.
- **5.** Drag the function onto the mapping statement.

Mapping Builder								
Source			Mapping					
Mapping Components				Target Element: /Get/Organization/ID				
		Map 📩		State	ment			
					<>> <rnb_v1_2:id></rnb_v1_2:id>			
	Find	Q			Drag and Drop or Type here.			
Functions		-	•					
Advanced								
🕨 🚞 Boolean								
Conversion								
Date								
		=						
▶ <i>f_x</i> lookupValue								
🕨 🛅 Mathematical 😽								

The mapper prompts you to select a previously created lookup.

/lapping Builder > Lookup Browser	r			×
Filter (type full or partial name)	Q,			🕜 Help
Lookup Tables		MyLookup	Preview	
MyLookup		Oracle Sales Cloud	Oracle R	ightNow
2		No Preview Available		
			Use	Cancel

6. Search by lookup name.

When you select a lookup in the **Lookup Tables** column, you see preview data for that lookup on the right based on the connections you selected as the source and target.

- 7. Select a lookup and click Use.
- **8.** The system automatically populates the following parameters in the lookup function:
 - dvmLocation: with the lookup name you selected
 - srcColumn: with the source application type
 - srcValue: with the existing mapping expression
 - targetColumn: with the target application type
- **9.** Enter a defaultValue parameter. This is the value sent to the target if the lookup function is unable to find a match for the value passed from the source.

10. Click Save and Close.

Deleting Mapping Statements

You can delete one or all mapping statements from the **Actions** menu in the Mapper Builder.

To delete mapping statements:

- **1.** Drag the source to a target to create a simple mapping.
- 2. Click the target node to launch the Mapping Builder.
- **3.** To delete a single statement, right-click a statement or click the **Actions** menu and select **Delete**.
- 4. To delete all statements, click the Actions menu and select Delete All.