
ORACLE®

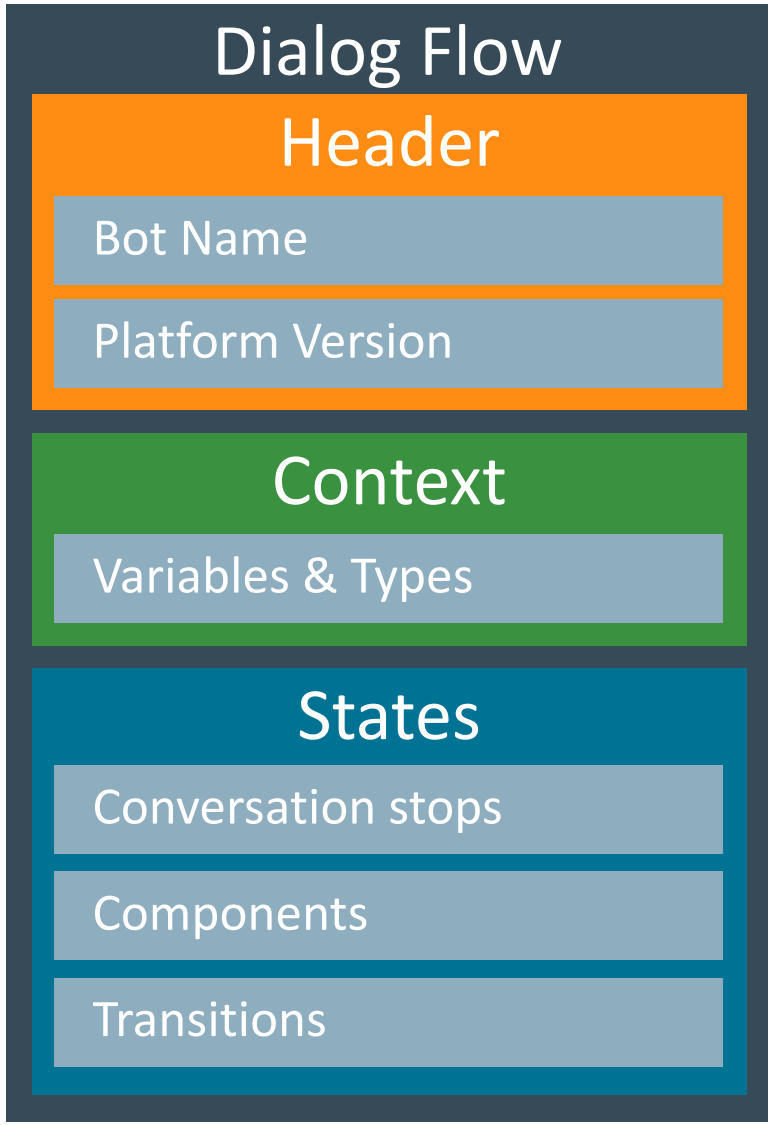
Oracle Intelligent Bots Advanced Training

Advanced Dialog Flow Topics

ORACLE

Safe Harbor Statement

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



Dialog Flow

- Defines user-bot conversations as a sequence of state navigation
  - Invokes Components
    - Executes Logic
    - Receives User Input
    - Returns Bot Responses
  - Determines navigation
- Three sections
  - Header, Context, States
- Developed in BotML

Topic Agenda

1. Variables
2. Navigation
3. Entity Slotting
4. Apache FreeMarker
5. Hints & Tips

Dialog Flow

Variables

Variable Types

- Context Variable
  - Used for temporarily saving data
- User Variable
  - Defined at runtime using a name prefix of "user."
- Profile Variable
- System Variable
  - Set by system components
    - Manage internal state
    - Flag conditions

System Variables

- system.errorState
  - Name of state that caused an error
- system.invalidUserInput
  - Boolean flag set when users failed providing a valid user input
- system.processedUserMessage
  - Boolean flag whether the CR component processes the user message
- system.entityToResolve
  - Used with composite bag entity to track the current entity to resolve
- system.unexpectedAction / system.actualState*
  - Name of out-of-order action executed by a user
- System.expectedState*

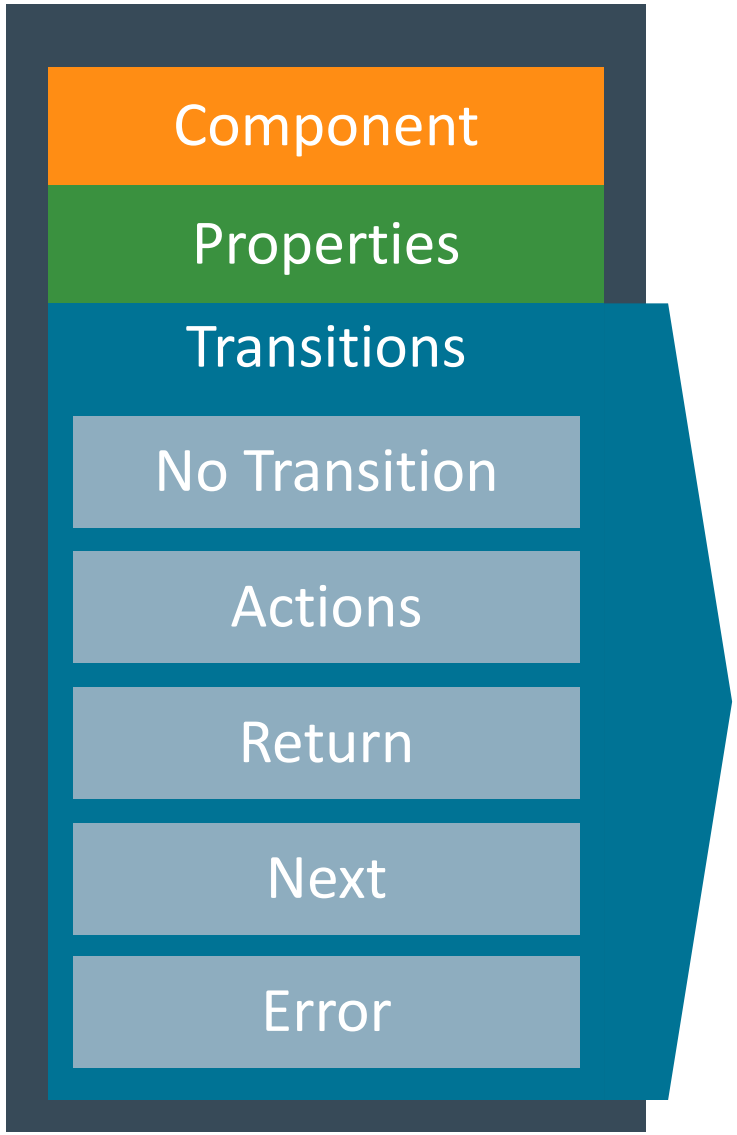
*version 18.3.4+

Dialog Flow

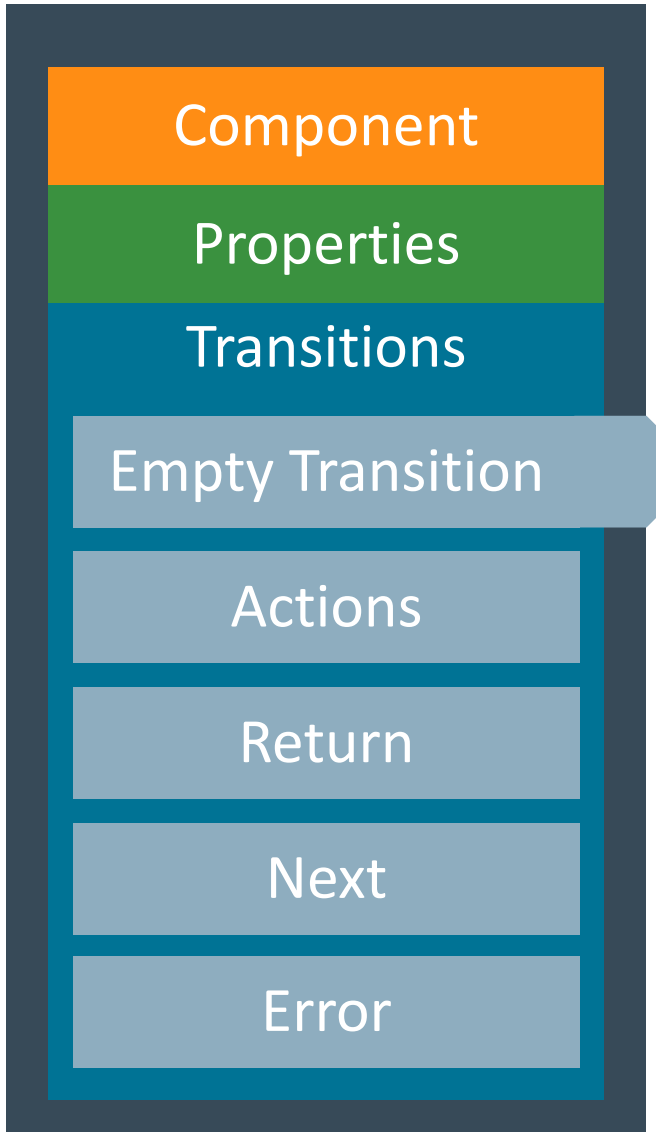
Navigation

Navigation may be hinted by the component but is **defined on the dialog flow state**

Image courtesy of Farbentek at FreeDigitalPhotos.net

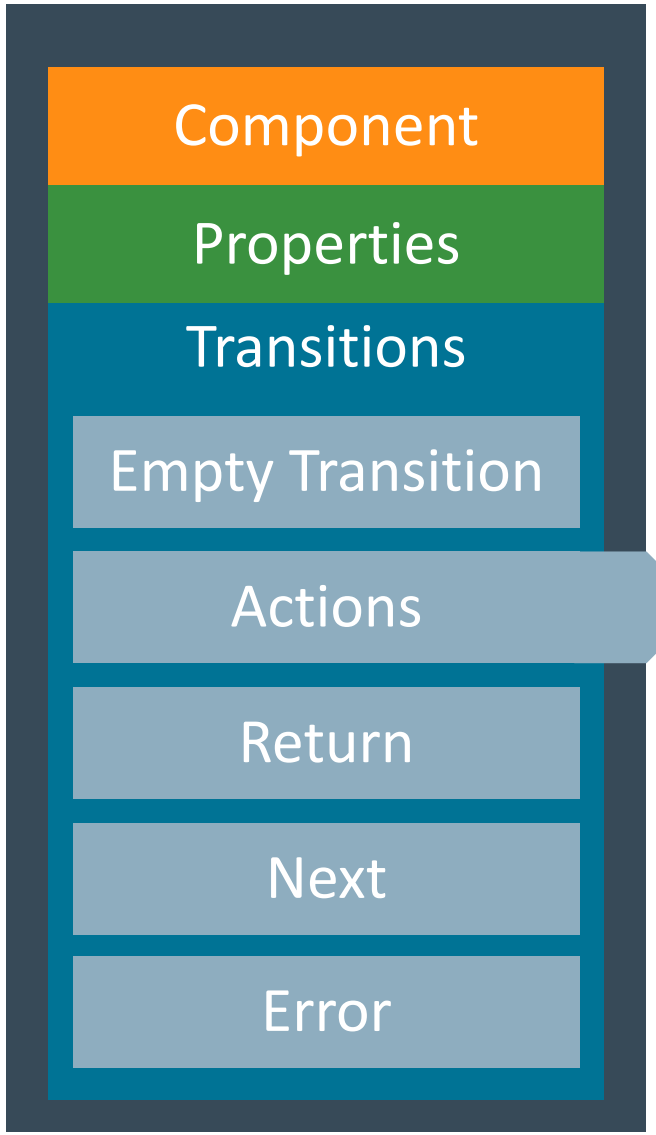


Every state may define zero or more directives to determine how navigation to a next state should happen.



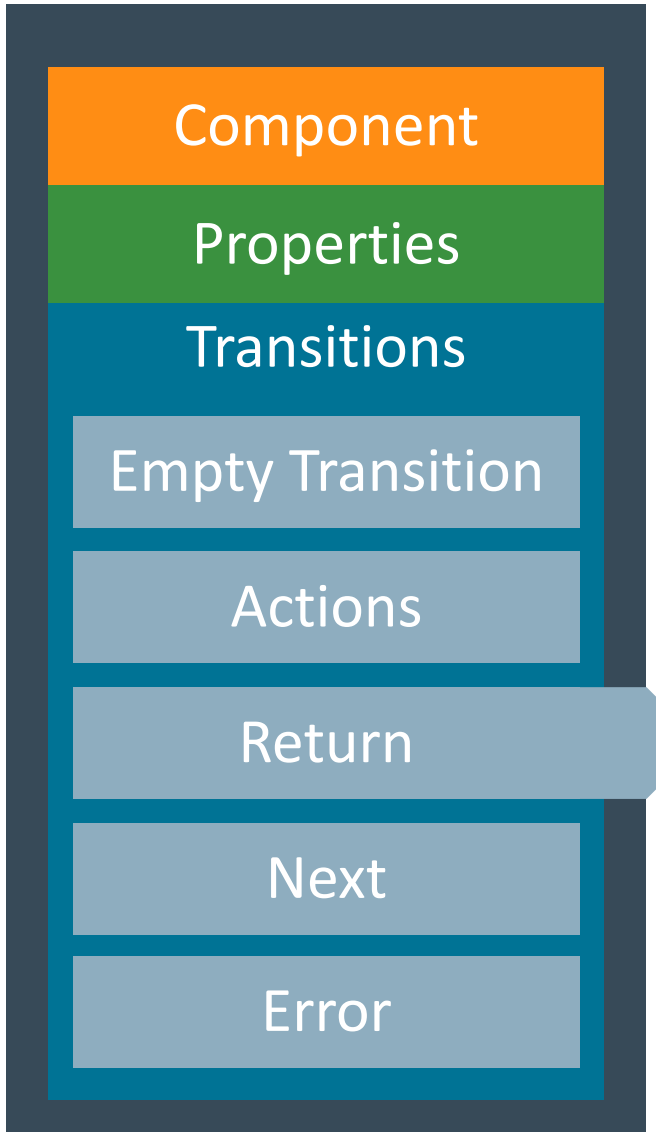

```
sayHello:
  component: "System.Output"
    properties:
       text: " ... "
     transitions: {}
```

- Performs default navigation
  - Top-to-bottom
  - Use for sequential navigation



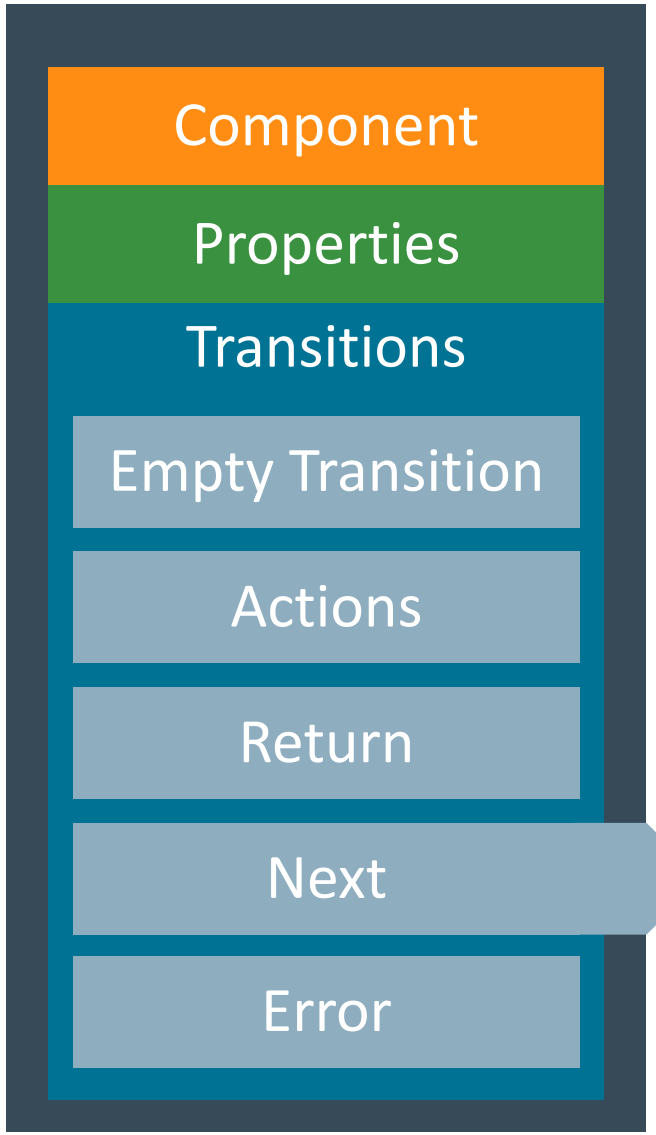

```
validateUserEntry:
  component: "System.MatchEntity"
    properties:
      …
    transitions:
      actions:
        match: "handleValidEntry"
        nomatch: "handleInvalidEntry"
```

- Actions are
  - Component outcome strings
  - Mapped to a state in the dialog flow



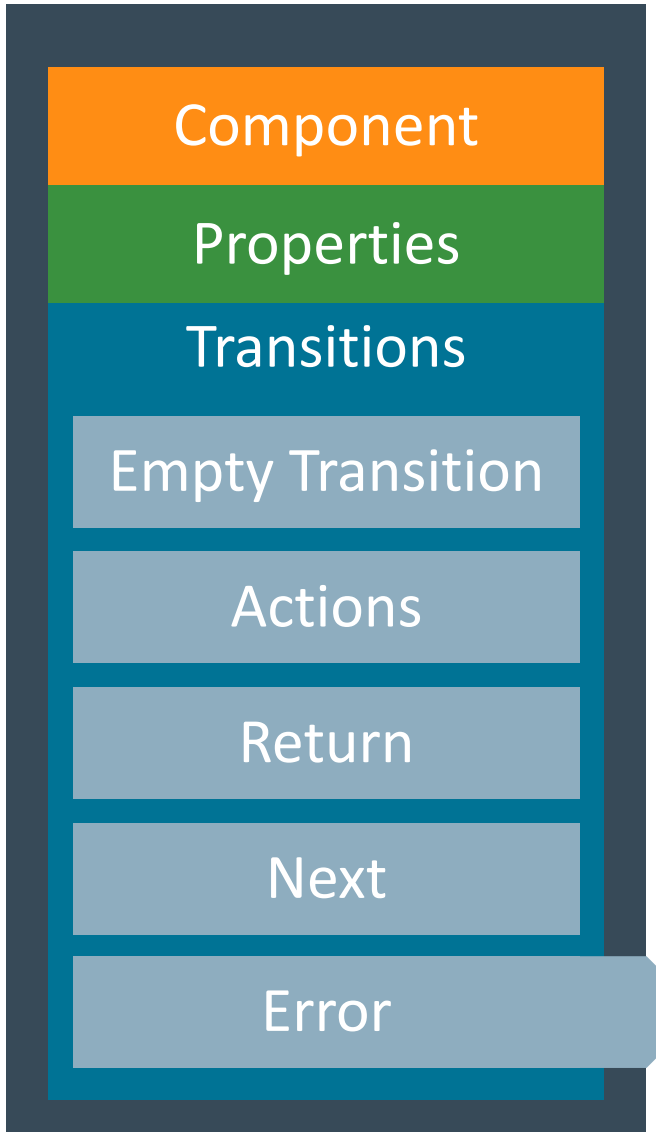

```
printBalance:
  component: "System.Output"
  properties:
    …
  transitions:
    return: "done"
```

- Exits the flow
- Reset all flow context variables
- Entry back into flow begins back at the top state




```
buyFlow:
  component: "System.SetVariable"
  properties:
    …
  transitions:
    next: "verifyCode"
```

- Proceeds to named state




```
type:
  component: "System.Intent"
  properties:
    …
transitions:
    error: "handleError"
```

- If an error occurs, proceeds to named state
  - Local error handling

Discussion

What if you use multiple transitions in a single configuration, e.g. using **actions** and **next**?

Discussion

What if no transition is configured
for a for a state?

Discussion

What if a component returns **three** action outcomes but only **two** are configured for a state?

Default Transitions

- Global transitions defined in dialog flow context
  - Optional (but useful)
- May be defined for
  - actions
  - *error* transition
    - No more "Uups …" messages
  - *next* transition
- Used as a fallback when local state does not handle a transition type

```
variables:
  …

defaultTransitions:
  error: "globalErrorHandler"
  next: "globalNextState"
  actions:
    unexpectedAction: "unhandledAction"
    ...

states:
  ...
```

Dialog Flow

Entity Slotting

Entity Slotting

Old 'SetVariable' Style

```
variables:
  orderId: "ORDER_ID"
  iResult: "nlpresult"
…
states:
  setOrderId:
    component: "System.SetVariable"
    properties:
      variable: "orderId"
      value: "${iResult.value.entityMatches['ORDER_ID'][0]}"
  handleOrder:
    component: "System.Text"
    properties:
      variable: "orderId"
```

Entity Slotting

New 'Component' Style

```
variables:
  orderId: "ORDER_ID"
  iResult: "nlpresult"
…
states:
  handleBooking:
    component: "System.Text"
    properties:
      prompt: "What is your Order Id?"
      variable: "orderId"
      nlpResultVariable: "iResult"
    transitions: {}
```

Discussion

In entity slotting, what is skipped
the state or the component?

And why does it matter?

Dialog Flow

Apache FreeMarker Expressions

About Apache FreeMarker

- Open source templating engine
- Used in BotML to access variables
  - ${variable_name.value}
- Built-in Expressions are appended to value expressions
  - ${variable_name.value?built-in-expression}
- Directive Expressions are scripts that conditionally assign values

Apache FreeMarker Built-In Reference

https://freemarker.apache.org/docs/ref_builtins.html

- Built-ins for strings
- Built-ins for numbers
- Built-ins for date/time/date-time
- Built-ins for boolean
- Built-ins for arrays

Working with Apache FreeMarker expressions **is like cooking**. A good cook can read any recipe. A master chef writes his own.

Image courtesy of Farbentek at FreeDigitalPhotos.net

Dialog Flow

Apache FreeMarker Expressions – String Built-Ins

String Built-In Examples

```
setStringValue:
  component: "System.SetVariable"
  properties:
    variable: "textVar"
    value: "Hello World"
```

FreeMarker Expression	Result
${textVar.value! 'empty'}	prints "empty" if no value set
${textVar.value?length}	11
${textVar.value?replace('World', 'Friends')}	Hello Friends

Apache FreeMarker expressions can be **chained** using the '?' character

String Built-In Examples

```
setStringValue:
  component: "System.SetVariable"
  properties:
    variable: "flight"
    value: "UA 1234"
```

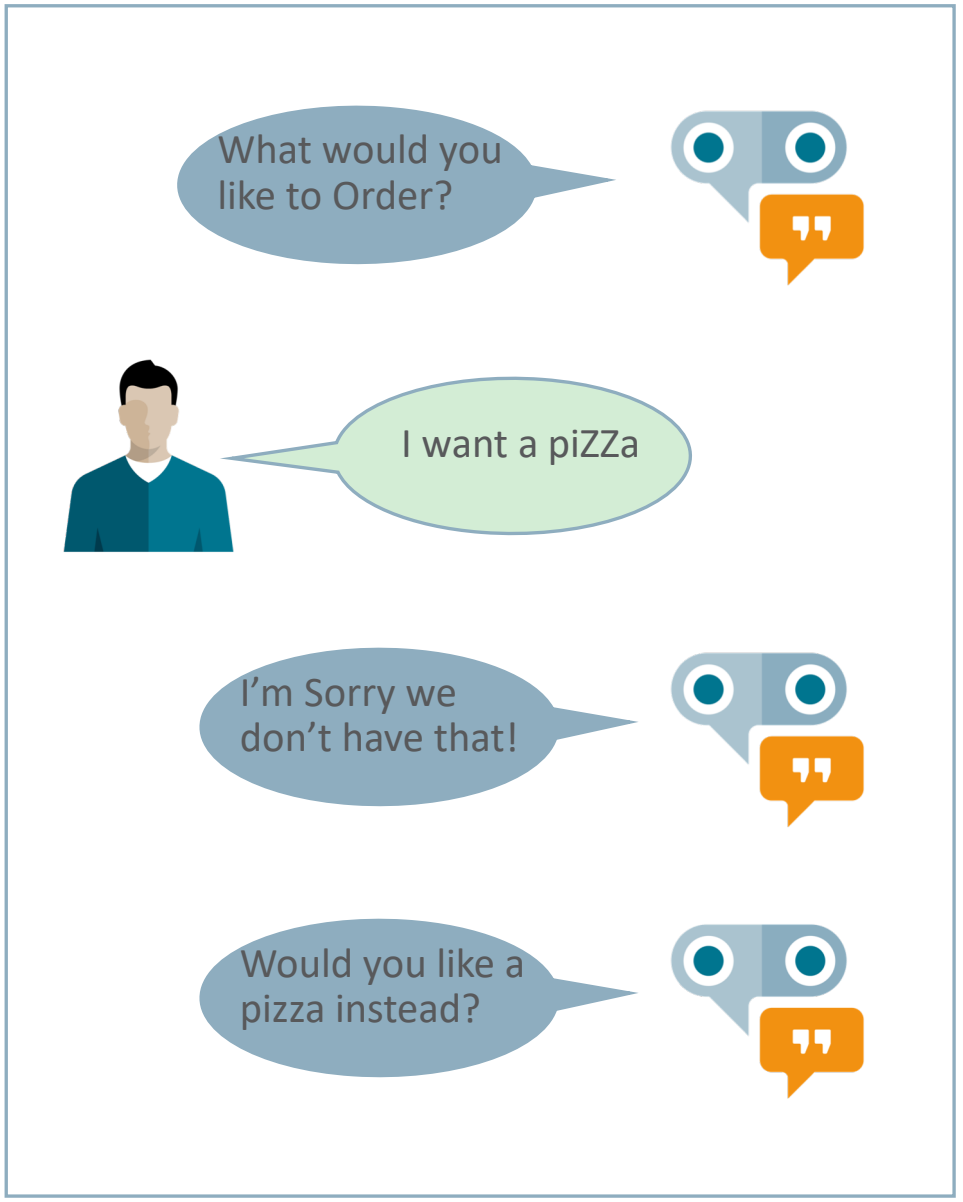
FreeMarker Expression	Result
${flight.value?contains('ua')?string('UA flight','other')}	UA flight
${flight.value?trim?lower_case?remove_beginning('ua ')}"	1234

Controlling User Input

Simple Example

```
component: "System.SetVariable"
properties:
  variable: "orderVar"
  value: "${userInputVar.value}"

 component: "System.Switch"
    properties:
      variable: "orderVar"
      values:
      - "PIZZA"
      - "PASTA"
   transitions:
      actions:
        PIZZA:  "OrderPizza"
        PASTA: "OrderPasta"
        NONE: "Sorry"
```

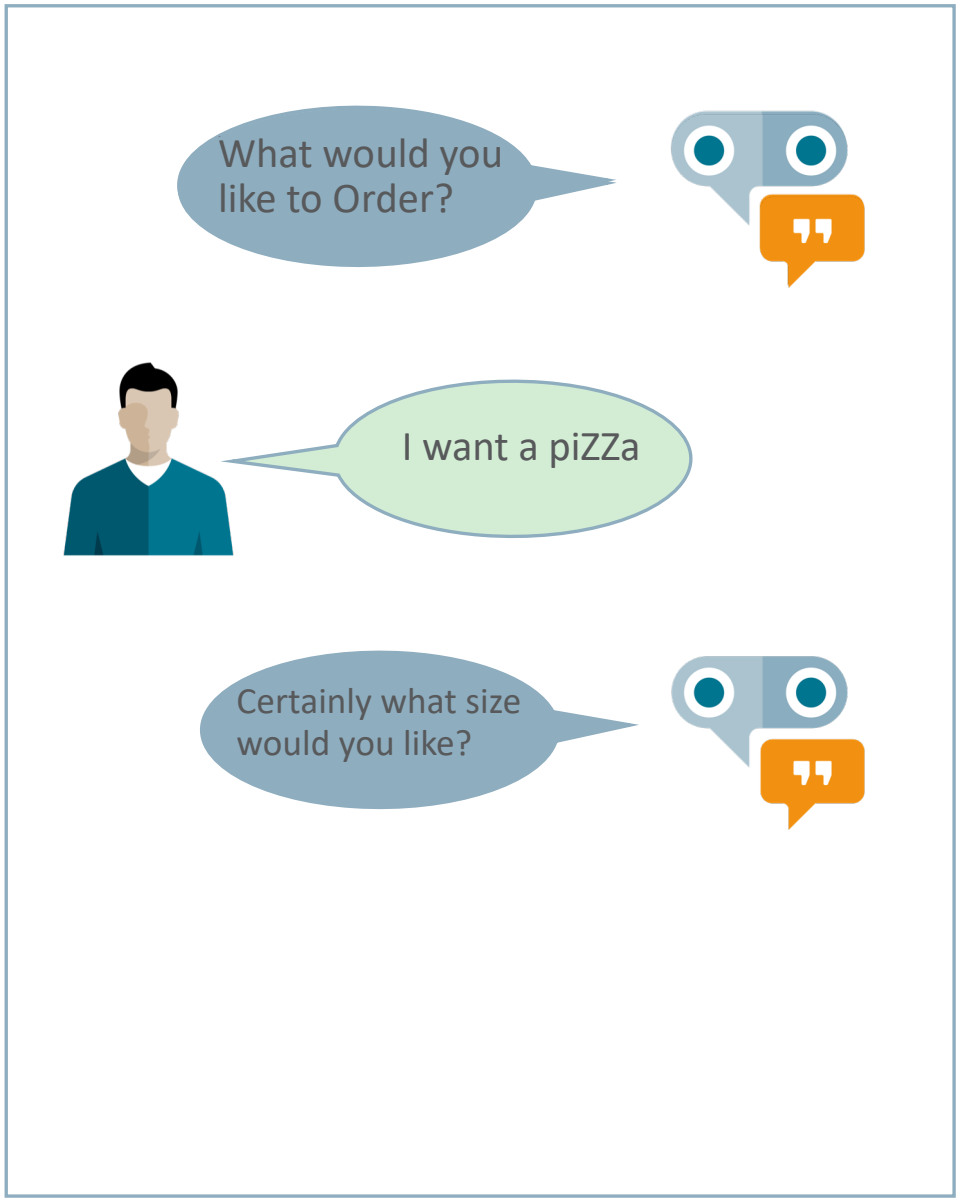


Controlling User Input

Simple Example

```
component: "System.SetVariable"
properties:
  variable: "orderVar"
  value: "${userInput?upper_case}"

 component: "System.Switch"
    properties:
      variable: "orderVar"
      values:
      - "PIZZA"
      - "PASTA"
   transitions:
      actions:
        PIZZA:  "OrderPizza"
        PASTA: "OrderPasta"
        NONE: "Sorry"
```



Text Built-In Overview

- boolean
- capitalize
- contains
- date, time, datetime
- ends_with
- ensure_ends_with
- ensure_starts_with
- index_of
- keep_after
- keep_after_last
- keep_before
- keep_before_last
- last_index_of
- left_pad
- length
- lower_case / upper_case
- number
- match
- replace
- right_pad
- remove_beginning
- remove_ending
- slice
- split
- starts_with
- string
- switch
- trim

Dialog Flow

Apache FreeMarker Expressions – Number Built-Ins

Number Built-In Examples

```
setNegativeValue:
    component: "System.SetVariable"
    properties:
      variable: "negativeValue"
      value: -2.512
```

FreeMarker Expression	Result
${negativeValue.value?abs}	2.512
${negativeValue.value?round}	- 2
${negativeValue.value?floor}	- 3
${negativeValue.value?string['###.##']}	- 2.51
${negativeValue.value?abs?string['###.##\u00A3']}	2.51 £

Number Built-In Overview

- abs
  - returns positive number
- round
  - rounds to the nearest whole number
  - ≥ n.5 rounded upwards
- floor
  - rounds the number downwards
- ceiling
  - rounds the number upwards

- lower_abc
  - converts 1, 2, 3, etc., to "a", "b", "c"
- upper_abc
  - converts 1, 2, 3, etc., to "A", "B", "C"

- string
  - converts a number to string
- <number>?string['###.##']
  - formatted string output
  - rounds to nearest neighbor
- Use utf-8 for currency
  - ?string['###.##\u00A4'] $
  - ?string['###.##\u20AC'] €
  - ?string['###.##\u00A3'] £

Dialog Flow

Apache FreeMarker Expressions – Date Built-Ins

Date Built-In Examples

```
variables:
  dateVar: "DATE"
…
getNLPDate:
  component: "System.SetVariable"
  properties:
    variable: "dateVar"
    value: "${iResult.value.entityMatches['DATE'][0]}"
```

FreeMarker Expression	Result
${dateVar.value.date}	Date in milliseconds UTC
${dateVar.value.date?long?number_to_date?string.short}	4/13/18
${dateVar.value.date?long?number_to_date?string.medium}	April 13, 1:29:54 PM
${dateVar.value.date?long?number_to_date?string.long}	April 13, 1:29:54 PM UTC

Date Built-In Examples

```
variables:
  dateVar: "DATE"
…
getNLPDate:
  component: "System.SetVariable"
  properties:
    variable: "dateVar"
    value: "${iResult.value.entityMatches['DATE'][0]}"
```

FreeMarker Expression	Result
${dateVar.value.date?string['dd.MM.yyyy, HH:mm']}	13.04.2018, 13:29
${dateVar.value.date?long?number_to_datetime?iso_local}	2018-04-13T13:29:54Z
${dateVar.value.date?time}	1:29:04

Setting a Default Value to a DATE Entity

```
variables:
  dateVar: "DATE"
  now: "string"
 …
  setNow:
    component: "System.SetVariable"
    properties:
      variable: "now"
      value: "${.now}"

  setDateVariable:
   component: "System.MatchEntity"
   properties:
     variable: "dateVar"
     sourceVariable: "now"

  print:
    component: "System.Output"
    properties:
      text: "${dateVar.value.date}"
    transitions:
      return: "done"
```

- Apache FreeMarker expression ${.now} sets "now" variable to current datetime
- System.MatchEntity
  - sets current date to dateVar context variable
- Entity "DATE" variable has "date" property
  - Milliseconds UTC

Dialog Flow

Apache FreeMarker Expressions – Array Built-Ins

Arrays in Oracle Intelligent Bots

- System arrays
  - ${iResult.value.entityMatches['name of entity']}
  - ${iResult.value.intentMatches.summary}
- Variable arrays
  - Context variable of type string
    - Custom Components
    - Apache FreeMarker Template Expression

Building Arrays with Apache FreeMarker

Array of Objects

```
variables:
  fruits: "string"
  …
states:

  fruitArray:
    component: "System.SetVariable"
    properties:
      variable: "fruits"
      value:
      - name: "Apple"
        color: "green"
      - name: "Banana"
        color: "yellow"
```

Simple Array

```
variables:
  colors: "string"
  …
states:
  colorsArray:
    component: "System.SetVariable"
    properties:
      variable: "colors"
      value:
      - "red"
      - "green"
      - "blue"
      - "yellow"
```

Array Built-In Examples

```
fruitArray:
    component: "System.SetVariable"
    properties:
      variable: "fruits"
      value:
      - name: "Apple"
        color: "green"
      - name: "Banana"
        color: "yellow"
```

FreeMarker Expression	Result
${fruits.value?size}	2
${fruits.value[0].name}	Apple
${fruits.value[0].name !'unknown'}	Same as above, but prints "unknown" if name is missing
${fruits.value?sort_by('color')?reverse[0].name}	Prints "Banana"

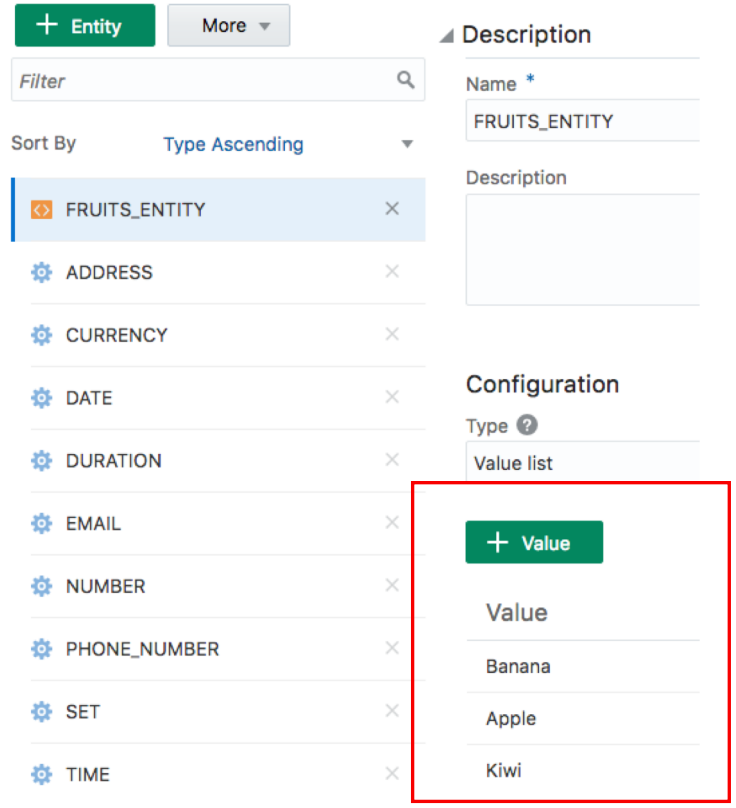
Array Built-Ins

- first.<attribute>, last.<attribute>
  - Returns first or last element's attribute
- seq_index_of(value)
  - Works with simple arrays
- seq_last_index_of(value)
  - Works with simple arrays
- join('delimiter_char')
  - Returns simple array as strings delimited by delimiter_char
- seq_contains(value)?string('yes','no')
  - returns yes or no string
- min, max
  - For simple arrays of type number or datetime
  - Returns min or max value
- sort
  - Sorts simple array
- sort_by(attribute)
  - Sorts object array by named attribute

Accessing Values from a "Value List" Entity

- Apache FreeMarker expressions to access values from a List entity
  - ${entityFruits.type.enumValues?split(',')?size}
    - Returns number of items = 3
  - ${entityFruits.type.enumValues?split(',')?sort[0]}
    - Returns first value in sorted array = Apple
  - ${entityFruits.type.enumValues?split(',')?seq_index_of('Kiwi')}
- Use cases
  - Access specific element in entity value list
  - Loop over entity value list
  - Detect if entity value list contains specific value

```
context:
  variables:
    entityFruits: "FRUITS_ENTITY"
```



Dialog Flow

Apache FreeMarker Directives

Apache FreeMarker directives are allow you to conditionally **assign content** to component properties

Example: <#if >, <#else>

```
<#if condition>
  ...
<#elseif condition2>
  ...
<#elseif condition3>
  ...
...
<#else>
  ...
</#if>
```

- Conditionally skip sections of a template
- Conditions must evaluate to boolean
- <#if> </#if> encloses <#elsif> and <#else>
  - The "if" tag can contain any number of "elseif" and optionally one "else"tag

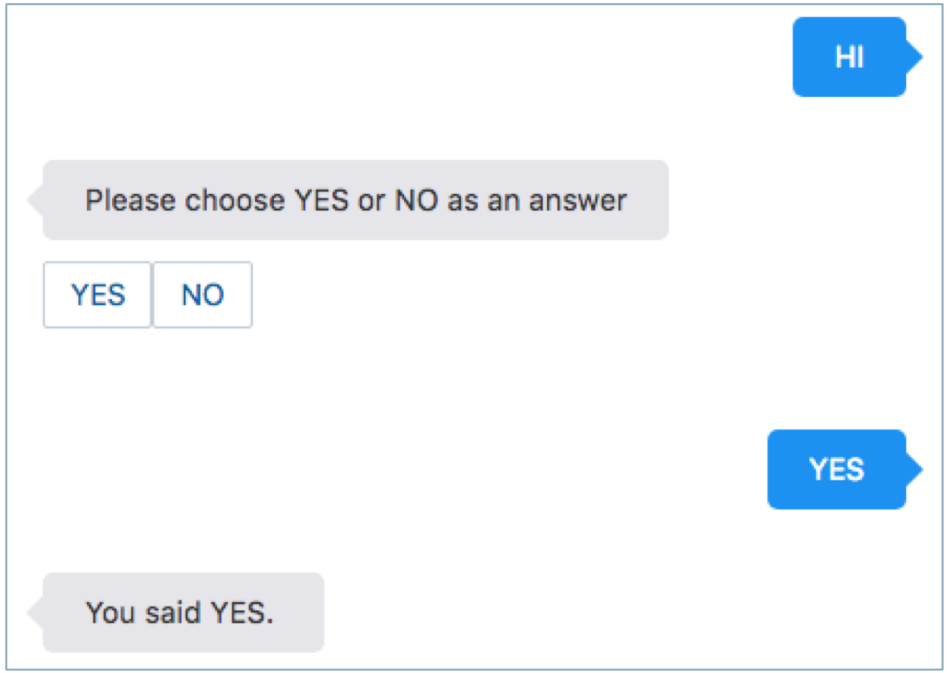
<#if >, <#else> Example 1

Conditional Messages

```
context:
  variables:
    yesNo: "string"

states:
  ...
  yesNoPrint:
    component: "System.Output"
    properties:
      text: "<#if yesNo.value?lower_case?matches('yes') >
             You said YES. <#else> You said NO. I am sorry. </#if>"
    transitions:
      return: "done"
```

Print conditional statement



Comparison Operator Encoding

Making Directives work in BotML

Operator	Encoding
>	>
<	<
>=	>=
<=	<=

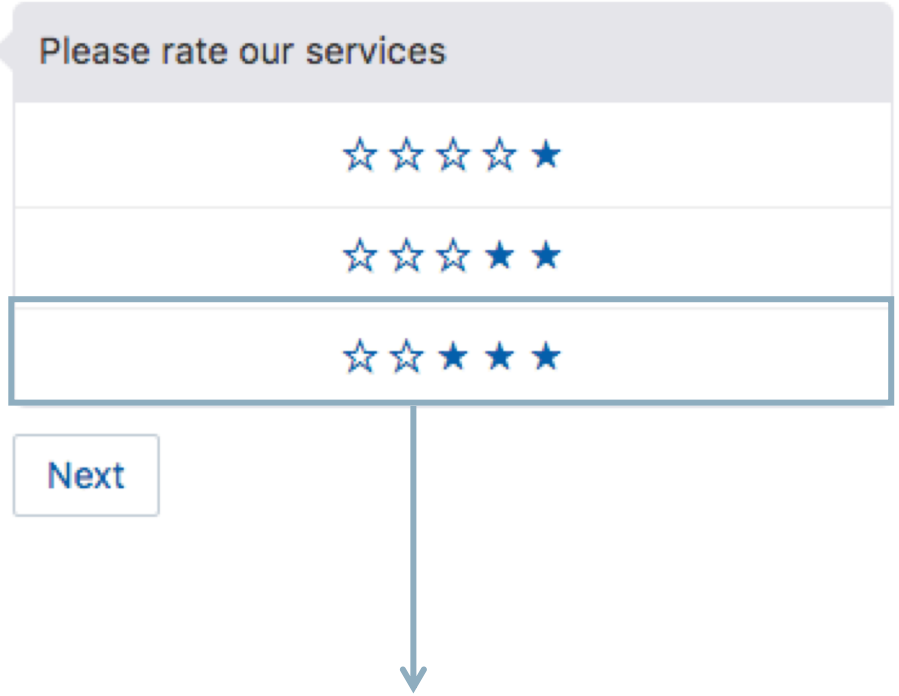
<#if >, <#else> Example 2

Pagination

```
component: "System.CommonResponse"
   properties:
     …
     metadata:
      responseItems:
      - type: "text"
        …
        actions:
        - label: "${starbars.label}"
          type: "postback"
```

rendered: "<#if starbars?index < (rangeStart?number + rangeSize?number)
&& starbars?index >= rangeStart?number>true<#else>false</#if>"

Conditional display rendering



<#if … > <#else></#if> expressions **can be nested**.

Yes, I know.

Dialog Flow

Apache FreeMarker <#list > Directive

<#list> … </#list> Example

```
states:
  setValues:
    component: "System.SetVariable"
    properties:
      variable: "data"
      value: {"FirstName":"John","LastName":"Doe","RegistrationId":"1234" }
```

```
print:
    component: "System.Output"
    properties:
      text: "<#list data.value?keys as k> \"${k}\" has a value of \"${data.value[k]}\"\n</#list>"
    transitions:
      return: "done"
```

"FirstName" has a value of "John"
"LastName" has a value of "Doe"
"RegistrationId" has a value of "1234"

Real Word Example

Find the value of an Attribute in an Array of Objects: E.g. the price for Audi

```
SetArray:
  component: "System.SetVariable"
  properties:
    variable: "listvar"
    value:
    - product: "BMW"
      price: 20000
    - product: "Audi"
      price: 21000
    - product: "Porsche"
      price: 22000


PrintPriceForAudi:
  component: "System.Output"
  properties:
    text: "<#list listvar.value as k> <#if k['product'] == 'Audi'> ${k['price']} <#break> </#if> </#list>"
  transitions:
    return: "done"
```

21000

Dialog Flow

Hints & Tips

How to Access User Input from Expressions in BotML

- There are two strategies
  - Use `System.Text` to save user input to context variable
    - Reference variable from System.Intent, System.TranslateInput, Custom Component etc.
    - `${userInputVar.value}`
  - Access user input from iResult (nlpresult type) variable
    - Requires `System.Intent` to handle user input
    - `${iResult.value.query}`

Accessing a List of Resolved Entities

Using iResult as the variable of type nlpresult

I like a pizza salami, a pizza cheese, a lemonade, a coke and a beer

- How many Entities?
  - ${iResult.value.entityMatches?size} → 2
- Name of first entity?
  - ${iResult.value.entityMatches**?keys**[0]} → PIZZA
- How many drinks?
  - ${iResult.value.entityMatches['DRINKS']?size} → 3
- Name of second drink?
  - ${iResult.value.entityMatches['DRINKS'][1]} → Coke

Accessing Intents

Using iResult as the variable of type nlpresult

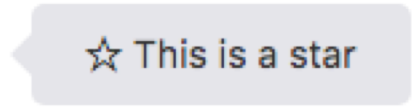
I like a pizza salami, a pizza cheese, a lemonade, a coke and a beer

- How many intents?
  - ${iResult.value.intentMatches.summary?size} → 1
- What is the intent name?
  - ${iResult.value.intentMatches.summary[0].intent} → OrderPizza
- What is the intent's score
  - ${iResult.value.intentMatches.summary[0].score} → 1 (= 100%)
- What is the name of the intent with the highest score?
  - ${iResult.value.intentMatches.summary**?sort_by('score')?reverse**[0].intent} → OrderPizza

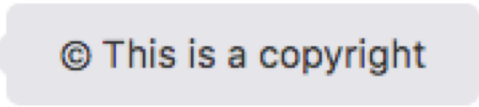
How-to Add Symbols to Bot Responses

Using Unicode Characters

```
printStar:
  component: "System.Output"
  properties:
    text: "\u2606 This is a star"
  transitions:
    return: "done"
```



```
printCopyright:
  component: "System.Output"
  properties:
    text: "\u00A9 This is a copyright"
  transitions:
    return: "done"
```



Advanced Bot Training Hands-On

Lab 3a: Critique Our Dialog Flow

Integrated Cloud

Applications & Platform Services


ORACLE®