



**Customer Journey Platform  
Business Rules Engine  
User Guide**

## **Notification**

The Broadsoft CC-One solution is now the Cisco Customer Journey Platform. From August 2018, you will see the Cisco name and company logo, along with the new product name on the software, documentation and packaging. During the transition, you may see both Broadsoft and Cisco brands and former product names.

## **Copyright Notice**

Copyright© 2018 Cisco Systems, Inc. All rights reserved.

# •••• Contents

<b>Chapter 1. Introduction</b> .....	5
<b>Chapter 2. Business Rules Engine Implementation</b> .....	7
Creating a Domain and a Set of Rules.....	8
Configuring a BRE Data Sync Instance.....	12
Uploading Records into the BRE Data Base.....	15
Creating the Call Control Script.....	16
<b>Appendix A. Business Reports Engine Sample Data</b> .....	19





# Introduction

The Cisco<sup>®</sup> Customer Journey Platform (CJP) Business Rules Engine (BRE) provides a means for tenants to incorporate their data into the CJP environment for custom routing as well as for general implementation.

For example, a tenant who wants to route calls to a specific group of agents based on the ANI dialed could simply upload a list of ANIs and their associated virtual team identifiers. If the ANI of the incoming call is on that list, it is routed to the group of agents specified. If the ANI is not on the list, it is routed straight through to the general queue.

A typical BRE implementation involves these major components:

- **Business Rules Engine utility**, which provides an interface for creating domains and rule sets. The BRE requires an incoming decision request to be associated with a *domain*. The domain contains a set of rules. Each rule is assigned a priority. The BRE tries to match the domain's highest priority rule with the decision request based on the conditions in the rules.
- **Call Flow Design Interface** in the Routing Strategy module of the CJP dashboard, where you can create a control script that invokes the BRE.
- **BRE Data Sync** configuration utility, which provides an interface for defining the Data Sync instances to import data into the BRE database. After a Data Sync instance is defined, the tenant can upload a .csv file to a specified folder within their CJP FTP folder. The folder is periodically monitored based on parameters specified for that BRE Data Sync instance. The uploaded .csv data is converted into records in the BRE database.
- **Data Sync FTP location** where the tenant uploads records in a .csv file to the specified FTP folder. If the .csv file is processed successfully, the records are imported into the BRE database.





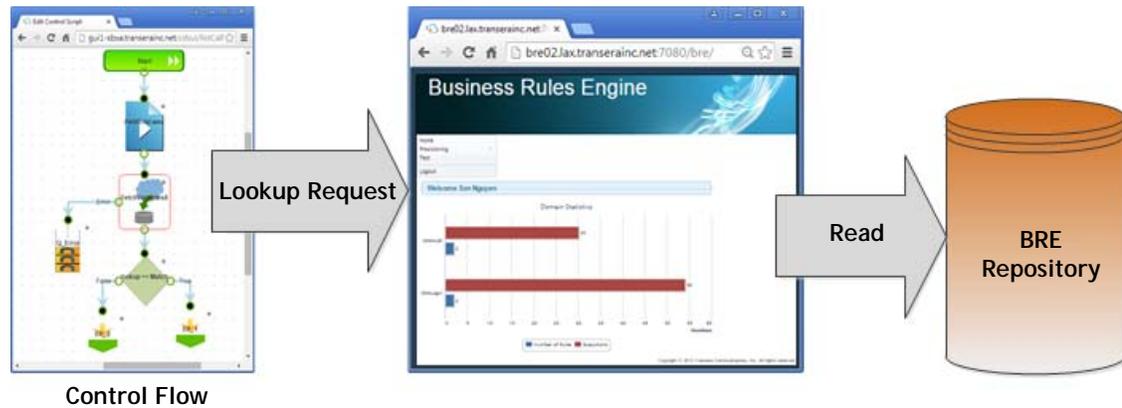
## Business Rules Engine Implementation

BRE consists of a real-time rules engine that gets invoked by the call flow scripts associated with an entry point. Typically the ANI of the incoming call along with other information associated with the call (such as an account number) are used by the rules engine to assess whether any special handling is needed for that call. This rules engine typically consults with a data repository to assess the desired call handling. The BRE configuration requires:

- [Creating a Domain and a Set of Rules](#)
- [Configuring a BRE Data Sync Instance](#)
- [Uploading Records into the BRE Data Base](#)
- [Creating the Call Control Script](#)

## Creating a Domain and a Set of Rules

The Business Rule Engine utility is typically invoked by the CFB script when a new voice (or chat or email) request is presented to the ACD. This section explains how rules can be set up so that the BRE utility can assist the ACD to route the incoming request.



The BRE requires an incoming decision request to be associated with a domain and a set of rules. The BRE tries to match the highest priority rule with the decision request based on conditions in the rules.

Be sure to create a rule set to cover all cases. For example, you should create rules for both *Match Found* and *Match Not Found* conditions. Or you could set up rules for multiple conditions. For example, *ANI Match* or *ANI No Match*, then *Gold* or *Silver*. In this case, you would need to create a rule for each possibility. For example:

ANI Match and Gold  
 ANI Match and Silver  
 ANI No Match and Gold  
 ANI No Match and Silver.

To create a domain and set of rules:

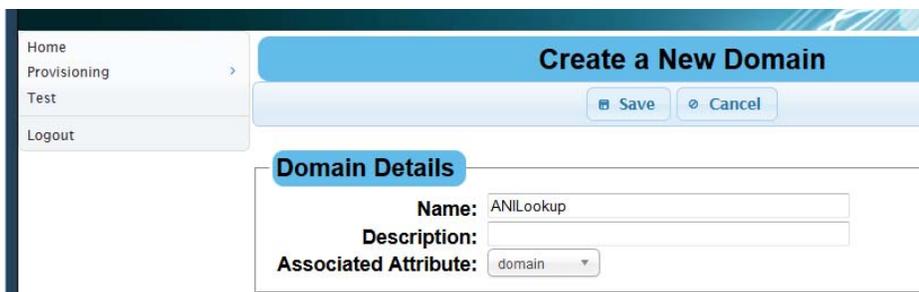
1. Log in to the Business Rules Engine utility using the URL and login credentials provided by Operations.
2. Create an attribute to associate with the domain if an attribute has not already been created. This attribute will carry the name of the targeted domain that the request to the BRE must be associated with in a call control script:
  - a. Select **Provisioning > Attributes** and click **New** on the page that appears.



- b. On the **Create a New Attribute** page, enter a text string in the **Name** field—for example, *domain* or *context*—and then click **Save**.



3. Select **Provisioning > Domains** and click **New** on the page that appears.
4. On the **Create a New Domain** page:
  - a. Enter a name and optional description for the domain.
  - b. If more than one attribute has been created, select the attribute you want to associate with this domain from the **Associated Attribute** drop-down list.
  - c. Click **Save**.



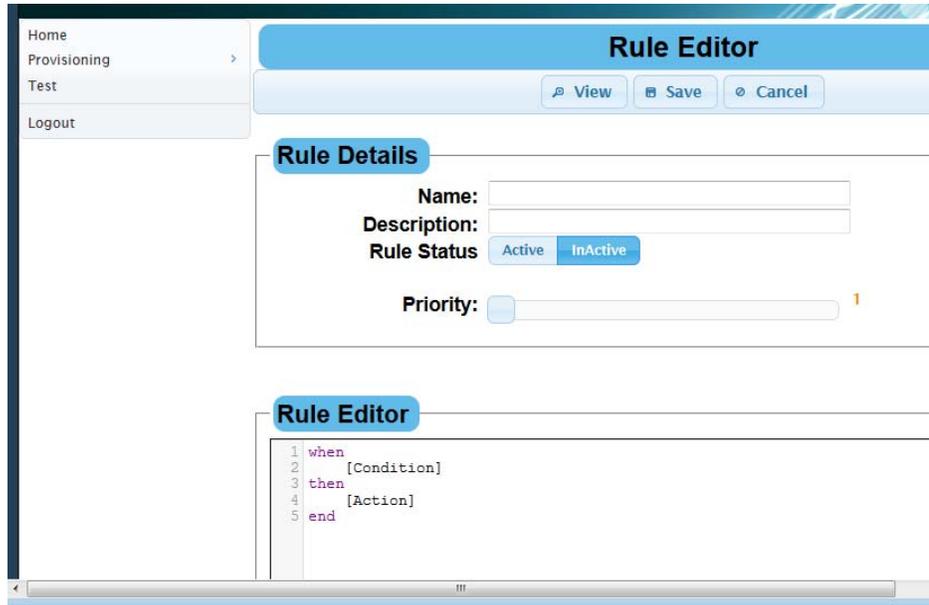
5. On the **Domains in the System** page, select the domain to create a rule set.



6. On the page that appears, click **New [Editor]**.



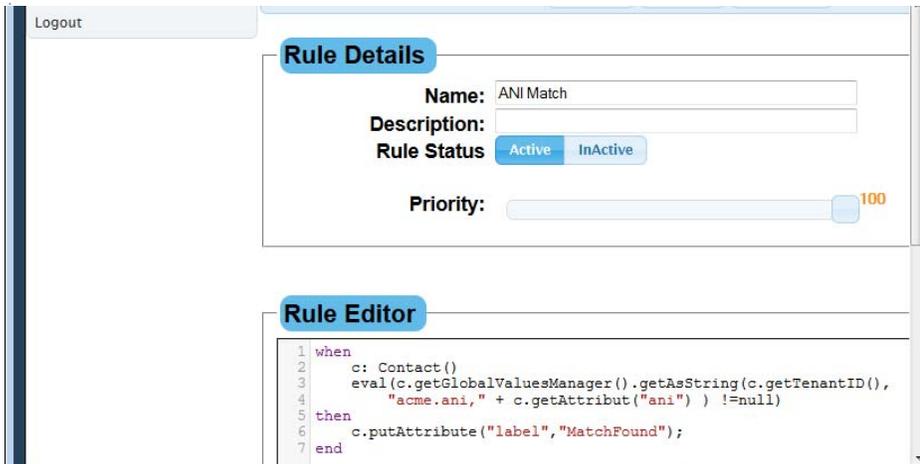
7. In the **Rule Details** section of the page, specify the settings described in the following table.



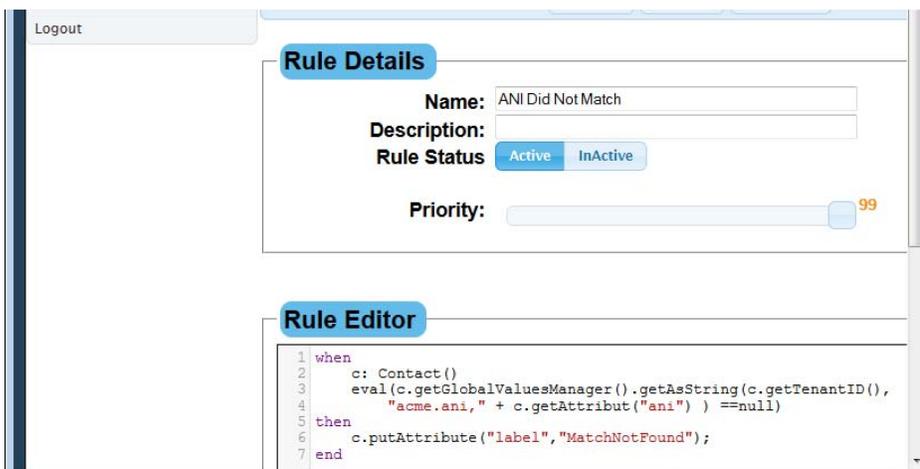
Field	Description
Name	The a name of the rule.
Description	An optional description of the rule.
Rule Status	Click a button to specify whether the rule is <b>Active</b> or <b>InActive</b> .
Priority	Drag the slider to assign a priority to the rule. The rules will be executed based on the assigned priority, from highest (100) to lowest.

- In the **Rule Editor** section of the page, enter the code for the rule and click **Save**.

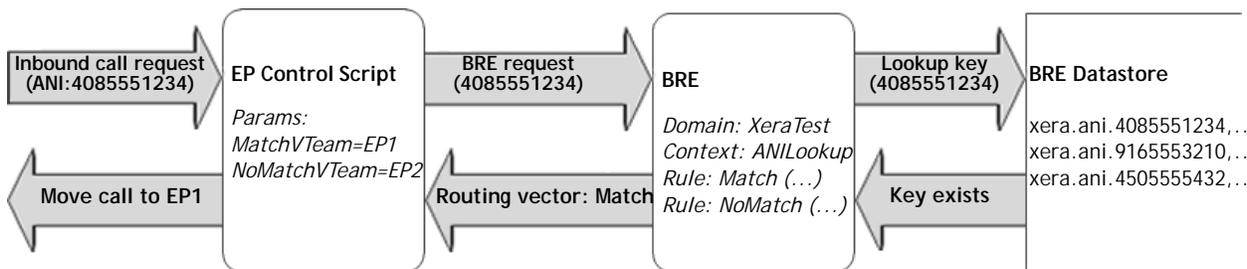
The following example code returns the value *MatchFound* for an attribute named *label* if the number the caller dialed from (ANI) matches an ANI on the list the tenant uploaded to the FTP server.



The following example code returns the value *MatchNotFound* if a match was not found.



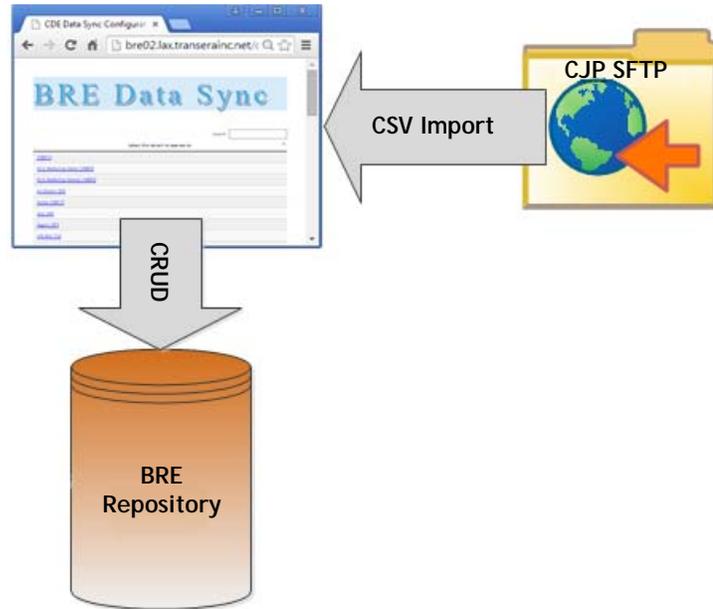
Example: Here is a rule set for routing calls based on ANI Match or No Match.



Simple ANI Match/No Match Example

## Configuring a BRE Data Sync Instance

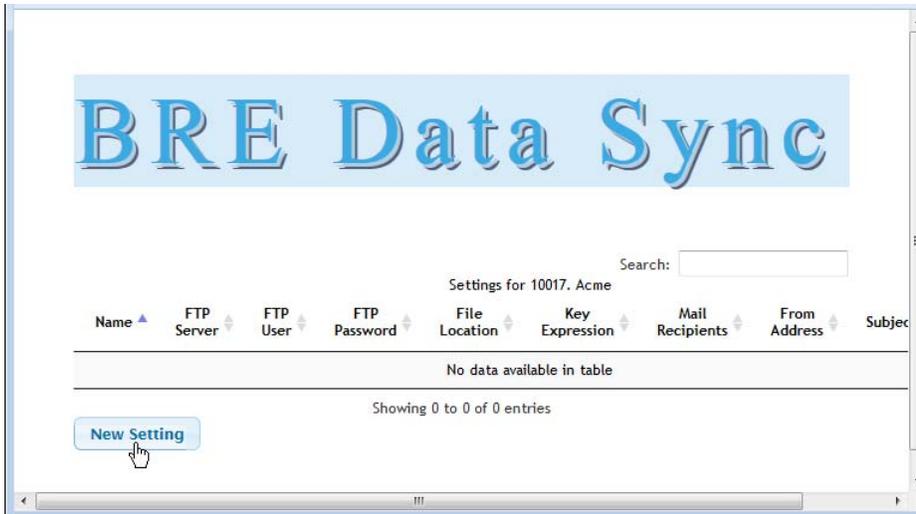
The BRE Utility accesses a database to make routing decisions. This database needs to be populated and periodically updated with the appropriate information. This section describes how the Data Sync utility can be configured to periodically update the database.



You must create a BRE Data Sync instance for every dataset that the Rules Engine will be consulting during its decision-making process. Each instance sets the system up to monitor a specified folder on the CJP FTP server and to begin processing the data in the .csv file as soon as it is uploaded. The uploaded data is converted into records in the database.

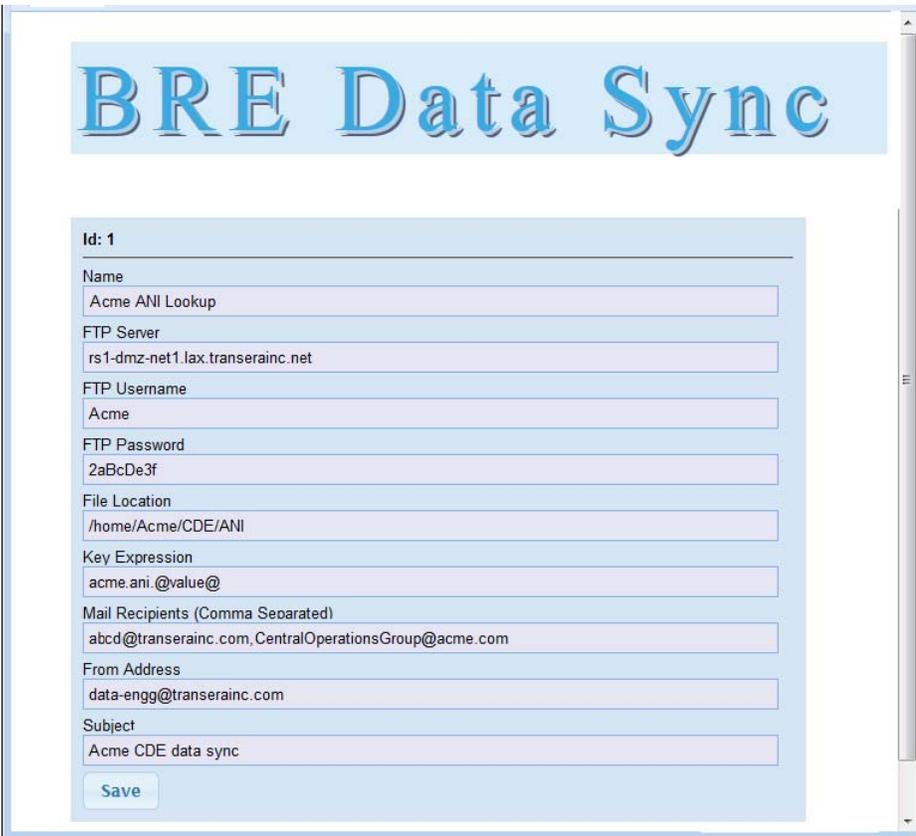
To create a new BRE Data Sync instance:

1. Navigate to the URL for the BRE Data Sync utility and select the name of the tenant.
2. On the page that appears, click **New Setting**.



3. Enter the appropriate value in each field and then click **Save**.

Settings for an example BRE Data Sync instance are shown in the next illustration, followed by a table describing each field.



Field	Description
Name	The name for the instance. This name is not used anywhere else.
FTP Server	The host name of the FTP server.
FTP Username	The tenant FTP login name.
FTP Password	The tenant FTP password.
File Location	The FTP location where the .csv file will be uploaded. For clarity and consistency, consider creating a subfolder beneath a folder named <i>BRE</i> in the tenant folder. For example: <i>home/Acme/BRE/ANI</i> .
Key Expression	This is the lookup key. The format must be followed exactly and consists of a key and a value. For example:  acme.ani.@value@  The key needs to be unique for every record. The value that is being imported will be appended to the key. The value can be anything, for example an ANI such as 4085551212.
Mail Recipients (Comma Separated)	The email addresses of the individuals who want to be notified of the status of the upload results. Every time a .csv file is uploaded and processed, the system sends an email to the mail recipients giving them the status of the run.
From Address	The <i>From</i> address that will appear in the email. For example, <i>data-engg@transerainc.com</i> .
Subject	The subject line of the email notification, for example <i>Acme ANI Lookup Datasync</i> .

## Uploading Records into the BRE Data Base

To load records into the BRE data base, the tenant must create a .csv file in a specific format and upload it to the folder on the FTP server specified in the BRE Data Sync instance, using the credentials supplied by CJP. If the .csv file is processed successfully, the records are imported into the BRE database.

The .csv format requires the following three fields separated by commas:

key, value, action

The key and value fields can have multiple parameters separated by a delimiter other than a comma, such as an underscore or pipe.

The action must be either ADD, UPDATE, or DELETE.

In the following example, the tenant wants to identify a VTeam ID based on the number the caller dialed (TFN) and the caller's area code, extracted from the ANI. This requires a compound key (TFN and area code). In this example, an underscore is used to separate the parameters. The key expression is **acme.tfn\_ac.@value@** where **acme** is the tenant name and **tfn.ac** represents the TFN&AreaCode combination key. These are some values that might be listed in the .csv file:

```
TFN_AreaCode,VTeamID,Action
8665551123_450,25766,ADD
8665551123_514,25766,ADD
8665551123_418,12742,ADD
8665559033_450,12695,ADD
8665559033_514,12695,UPDATE
```

Here is a simple example of just ANIs that may be sufficient for a rule with only two possibilities—match or no match—the value can be blank. For example:

```
ANI,Action
4155551234,ADD
4155558765,ADD
4085554321,ADD
4085559807,DELETE
```

Do not create duplicate records or update or delete a non-existent record. Doing so will cause the entire set of data in the.csv file to be rejected. For example, if the action is ADD and any of the keys already exist within the BRE repository, the entire .csv file will be discarded by the BRE application and the previous data in the database will continue to be used.

## Creating the Call Control Script

To create a call control script that interacts with the BRE, you will need the names and URL that are used in your implementation:

- name of the attribute associated with the targeted domain (for example, *domain*)
- name of the targeted domain (for example, *ANILlookup*).
- value of the `transera.rules.apiKey`—this value is manually inserted into the BRE database by Cisco CJP Professional Services and provides a way to map the request to the tenant.
- lookup key (for example *ani*, *vteam*, *pin*, *dnis*--whatever it is you want to look up on).
- the URL to use in the call flow for executing the BRE request. This URL is provided by Cisco CJP Operations in the following format:

```
http://<hostname>:<port>/bre/api/executeXML
```

You can create a call flow using the call flow design interface available in the CJP Management Portal only for a BRE implementation that involves two possibilities--match or no match. A more elaborate implementation, such as routing to a VTeam based on TFN and area code, requires a custom call control script.

To create a call control script within the CJP call flow design interface for a simple BRE implementation:

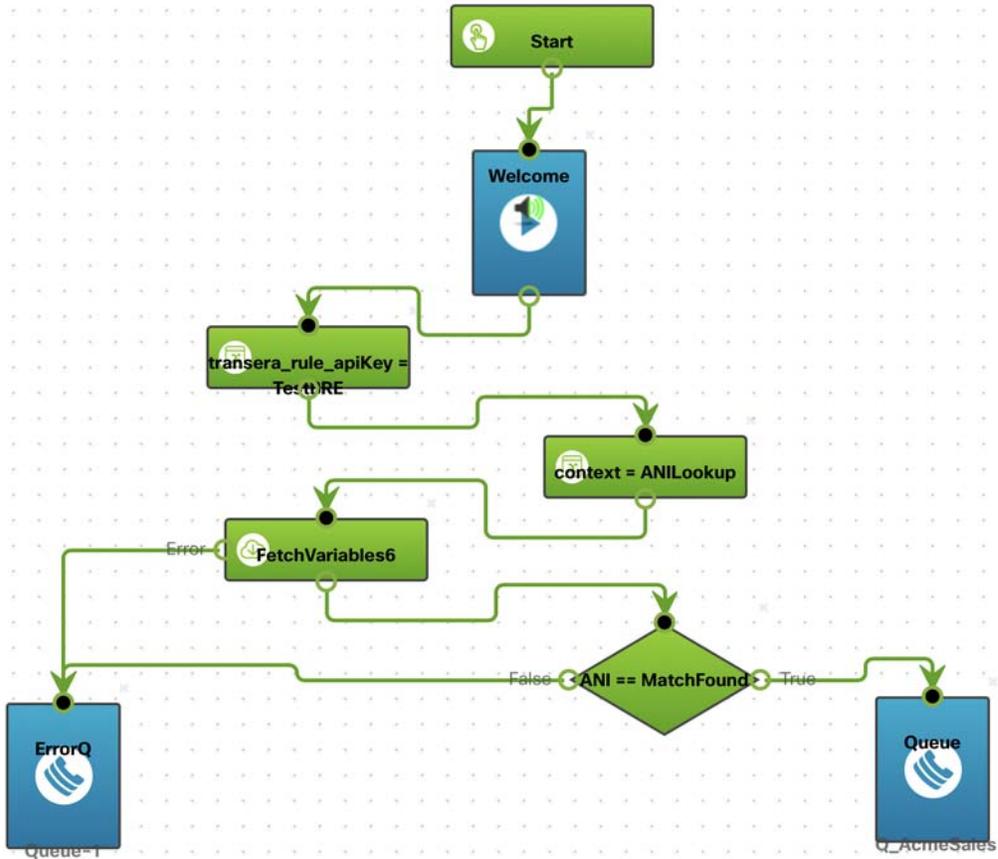
1. In the **Start** step's **Call Associated Data** field, add a CAD variable for the lookup key (for example *ani*).
2. Add a **Set Variable** step for `transera.rules.apiKey` set to the value provided by Cisco CJP Professional Services.
3. Add a **Set Variable** step for the domain attribute set to the name of the targeted domain. In the example domain described earlier in this chapter, the attribute name is *domain* and the domain name is *ANILlookup*.
4. Add a **Fetch Variable** step with the following parameters:

Parameter	Value
URL	Enter the URL for executing the BRE request provided by Cisco CJP Operations.
Request Variables	Select the names of the CAD variables to be sent on the request. These include the variables added to the Start step and the two Set Variable steps. For example, <i>transera.rules.apiKey</i> , <i>domain</i> , <i>api</i> .

Response Variables	Add the name of the variable that will hold the data returned by the BRE, as defined in the rule. For example, you would add a variable named <i>label</i> for the rule illustrated on page 11.
--------------------	---

5. Finish the control script based on whatever routing you want.

In the following example, a Condition step is configured to route the call to one of two queues based on whether or not the value returned for the *ANI* variable is *MatchFound*.







# Business Reports Engine Sample Data

This appendix contains sample data from the CJP FTP file. CJP uses this data to drive BRE execution.

Each row in the FTP Data file data contains three columns:

- The left column (**CallerPhoneNumber**) contains a phone-number that the BRE matches against the incoming caller's ANI.
- The center column (**CallerInfo**) specifies all the attributes for the ANI in the left column. Each attribute is a tuple consisting of a <name> and a <value> separated by a : (colon). The <name> should be alphanumeric and must not contain any spaces, : (colons), | (pipes), or other special characters. The pipe character concatenates multiple <name>:<value> pairs.
- The right column (**Action**) dictates if the row needs to be added, updated, or deleted from an existing BRE data table.

CallerPhoneNumber	CallerInfo	Action
4085551115	VTeamID:20001 PartnerID:1 PartnerName:Acme PartnerRegion:blueteam HDRegion:LT SiteID:823 VIP:False ContactID:123456	ADD
4085552222	VTeamID:20002 PartnerID:2 PartnerName:BestTechnologies PartnerRegion:redteam HDRegion:LT SiteID:114376 VIP:False ContactID:123458	ADD
4085551111	VTeamID:20003 PartnerID:1 PartnerName:ZenithDemoCenter PartnerRegion:blueteam HDRegion:LT SiteID:821 VIP:False ContactID:123451	UPDATE
4085551112	VTeamID:20003 PartnerID:1 PartnerName:WestServices PartnerRegion:greenteam HDRegion:LT SiteID:822 VIP:False ContactID:123452	UPDATE
4085551113	<Value field can be empty>	DELETE
4085551114	<Value field can be empty>	DELETE