FortiVoice - IVR Technical Note

Version 6.0.0

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO GUIDE
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://fortiguard.com/

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

October 4, 2019
FortiVoice 6.0.0 IVR Technical Note
26-600-584187-20191004

TABLE OF CONTENTS

Change Log

Date	Change Description
2019-10-04	Initial release.

Introduction

FortiVoice Interactive Voice Response (IVR) is an advanced version of the auto attendant. It allows interaction with callers through the use of voice and DTMF tones input via the keypad. Callers proceed according to the IVR audio instructions to reach the callees or get the information they need.

To set up IVR you must configure how data is collected, define matching conditions based on that data, and then define the action for these matching conditions. Based on the information collected from callers and by interacting with the backend database, FortiVoice IVR can prioritize the calls using call queues and present callers' information to the agents.

FortiVoice IVR also interfaces with RESTful web service for querying caller information from the database. For more information, see FortiVoice and FortiCare RESTful integration.

IVR is only fully configurable in the GUI.

To use IVR efficiently and to its maximum potential, it is important to understand the concepts involved before configuring IVR, including the various data collectors.

Data collectors

There are three data collector types:

- **SIP header collector:** Collects and stores the SIP header fields from the SIP INVITE.
- **Digits collector:** Collects and stores the digit-input from the caller's phone keypad.
- **RESTful collector:** Collects and stores information from the RESTful service.

IVR call handling is processed based on the information gathered from the collector configurations.

For example, FortiVoice IVR can collect customer information based on their FortiCare subscription through the RESTful collector. This information can be used by the agent to correctly dispatch their call to different call queues accordingly. For more information, see the configuration example FortiVoice and FortiCare RESTful integration.

Note that IVR can be configured to make use of one of, some of, or all collectors, depending upon the requirement.

SIP header collector

The SIP INVITE is fundamental to every SIP phone call, as it is the initial request message sent out by the caller inviting the intended recipient for a session. These invites are made up of SIP headers used to identify the source and destination, among other various forms of information.

However, the various values of these SIP headers, such as user names, IP and email addresses, is not what is relevant. FortiVoice, through the use of the SIP header collector, simply parses the names of the SIP headers themselves in order to pass along identifying information of a caller for inter-PBX communication.

For example, one PBX may be located in Ottawa, and another PBX located in Vancouver, both responsible for your company's support services. A customer calls into the Ottawa call center, and their identifying information is retrieved by the Ottawa PBX. If the call must then be transferred over to the Vancouver PBX, that identifying information is transmitted to the Vancouver PBX thanks to the SIP collector. This helps the customer avoid having to provide their identifying information again.

The SIP collector can furthermore make use of SIP header variables, allowing certain information to be appended to the agent console, the SIP header of the call being transferred between PBX systems, to remote CDR databases, and to IVR reports.

Digits collector

The digits collector operates similarly to the SIP header collector in that it gathers the information it needs. However, the SIP header collector is somewhat passive in its method, in that there is little to no effort from the caller's perspective. The digits collector on the other hand requires the active input of the caller to successfully route the call to the next phase of IVR.

Callers are routed through the auto attendant depending upon the number, or key, they press. The configuration of these keys and the actions taken upon pressing those keys are configured in the auto attendant. For more information,

see Assign IVR to the auto attendant on page 9. The digits collector of an IVR can only operate when the action of an auto attendant dial plan is set to route the call to an IVR profile.

Part of configuring the digits collector is assigning an audio prompt. This recording indicates to the caller that they must press the digit that corresponds to their need, or enter multiple digits in cases where they must enter a unique identifying number.

The other part to configuring the collector is to set the minimum and maximum number of digits allowed for the input, the maximum number of invalid attempts allowed, and timeout settings.

RESTful collector

FortiVoice IVR can interface with the RESTful web service for querying caller information from a database.

Both SIP header and digit collectors are referenced in RESTful collectors. Upon successful querying of the database, IVR handling is then processed. For more information, see Configure IVR handling on page 14.

The IVR RESTful interface can be configured to use HTTPS and authentication credentials to send the request. The RESTful interface supports HTTP versions 1.0 and 1.1, GET and POST request methods. The service uses either password-based or OAUTH authentication in order to send the requests.

Fields can be defined in order to retrieve the appropriate data for successful call routing. A standard format for these fields is XPath, an XML syntax that uses specific path expressions to select fields in an XML document for successful customer retrieval and identification. For more information, see Configure the RESTful collector on page 12.

FortiVoice and FortiCare RESTful integration

This example shows the FortiVoice IVR RESTful integration with FortiCare, however it can be implemented with other systems, such as SalesForce. When the RESTful service is set up and a caller dials in, the FortiVoice unit sends the caller information inquiry to the RESTful web service which sends back the information to the agent who processes the call.

In this example, customer information is retrieved by utilizing the Express Routing Code (ERC), a unique customer ID.

Call routing using ERC

Each customer has identifying data stored in FortiCare. The data is stored in the following format:

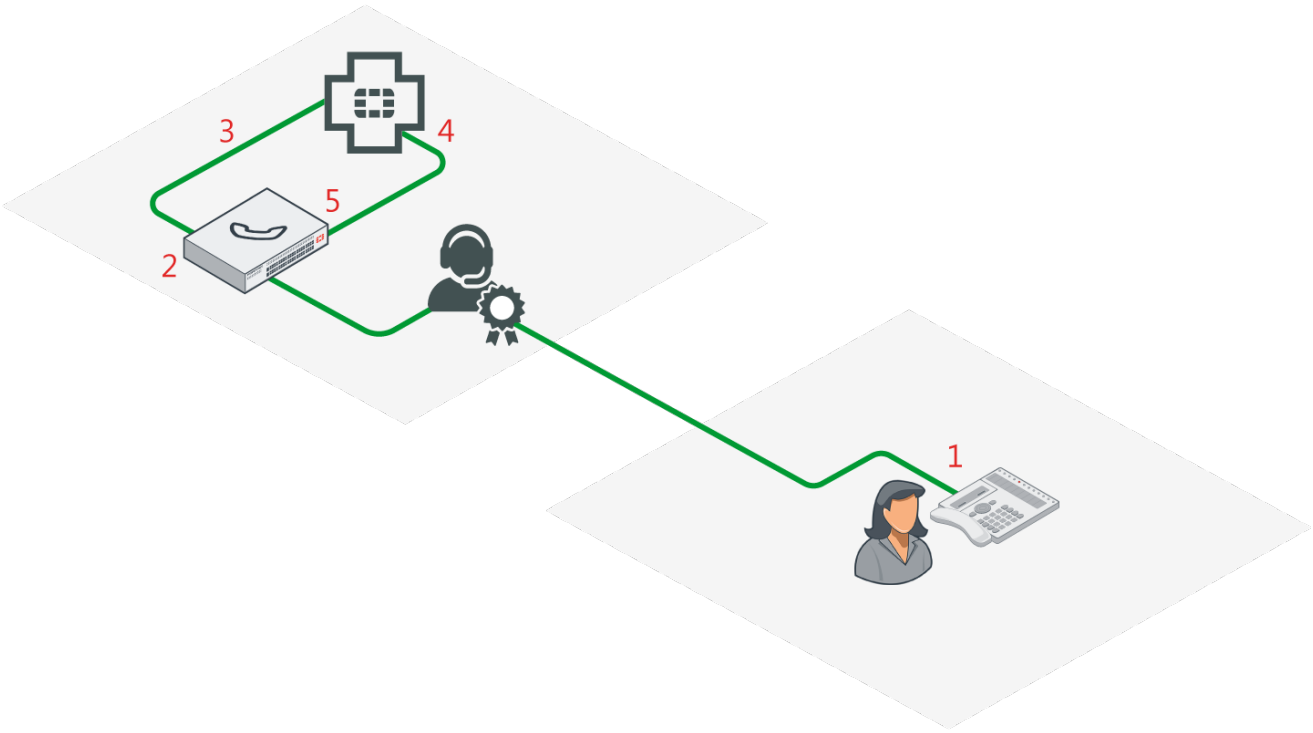
`Company, ProfileType, ERCCode, FirstName, LastName, PhoneGroup, Region, Email`

In this example, FortiVoice uses the ERC (`ERCCode`) as a customer ID to query the customer information from FortiCare. Upon a successful match, the call is then routed by `PhoneGroup` to the call queue.

Note that for this example, an IVR called *ERC* has already been created. In addition, custom audio greetings/prompts, the auto attendant, and a call queue have already been created.

Workflow:

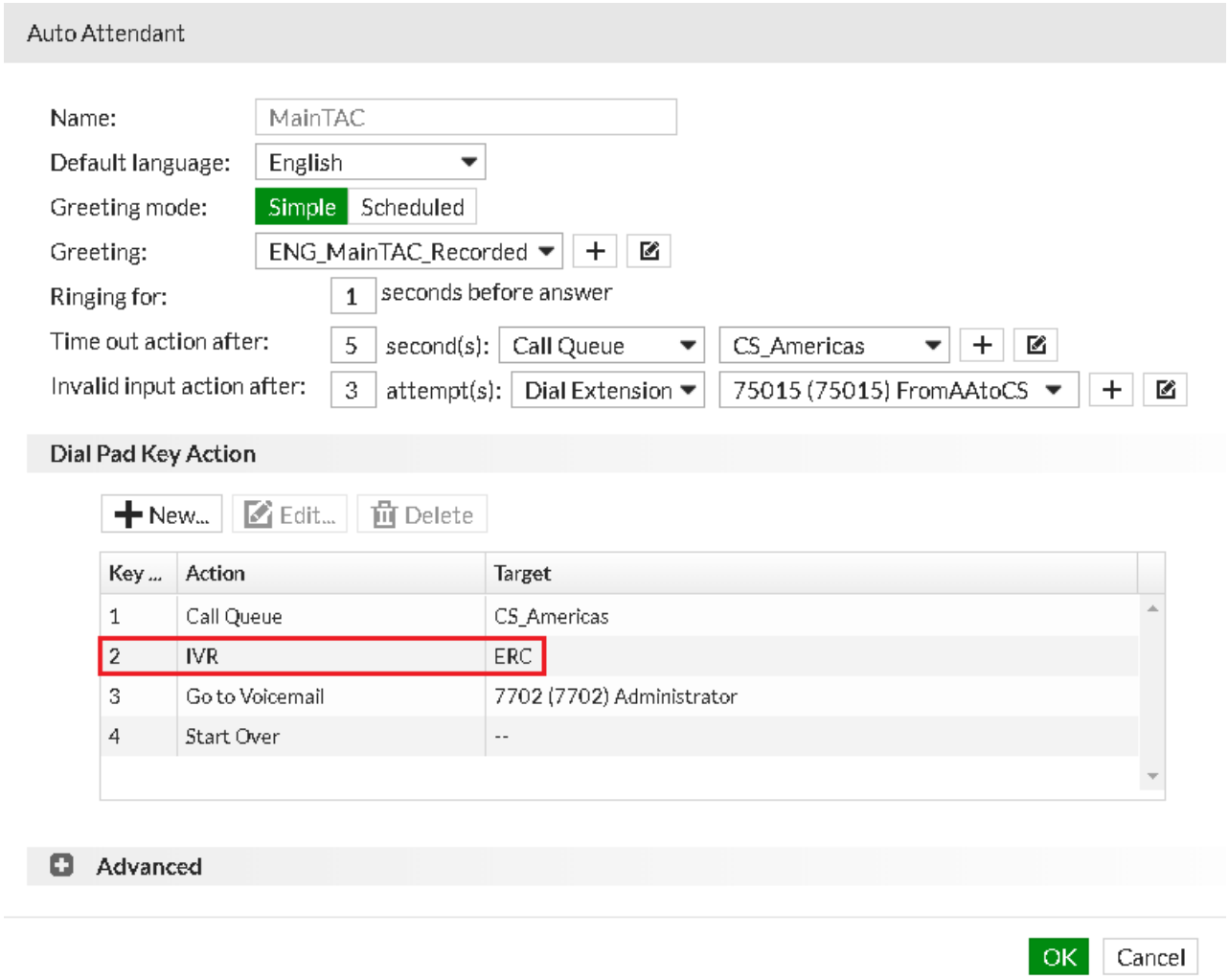
1. Customer calls in to Fortinet call center.
2. FortiVoice collects customer ID through IVR system.
3. According to the configuration, FortiVoice sends a RESTful request to FortiCare along with the customer ID.
4. FortiCare receives the request, searches for the customer by customer ID, and responds with a search result.
5. FortiVoice parses the response from FortiCare and utilizes the customer information (if any) according to the configuration, e.g. dispatch the call to a corresponding call queue.



Assign IVR to the auto attendant

1. Go to *Call Feature* > *Auto Attendant* > *Auto Attendant* and edit the appropriate auto attendant (in this example, *MainTAC*).
2. This auto attendant is configured for customer services operating in the Americas. Make sure that your IVR is assigned to a dial pad key action. In this example, customer's will dial 2 in order to initiate the identification query

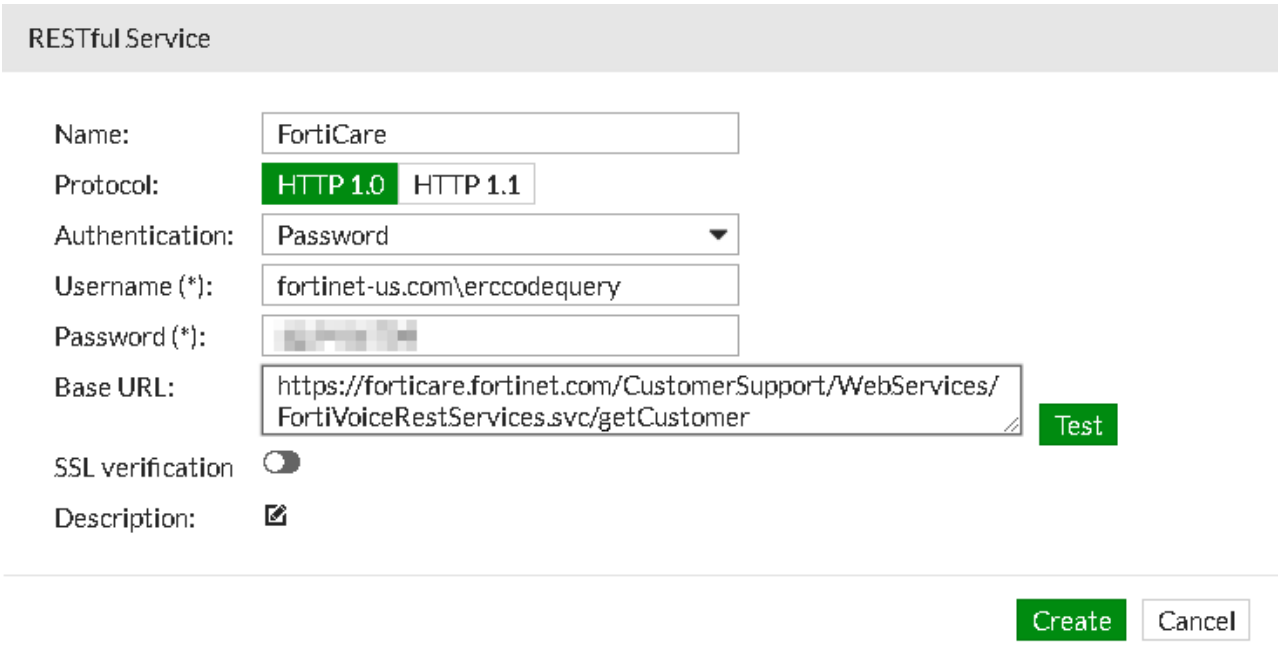
process.



Configure the RESTful service

1. Go to *Call Center > IVR > RESTful Service* and select *New*.
2. Enter a *Name*, set *Protocol* to *HTTP 1.0*, set *Authentication* to *Password*, and complete the remaining fields as required.

The *Username* and *Base URL* are provided to you by FortiCare, used to retrieve customer information.



Configure the IVR

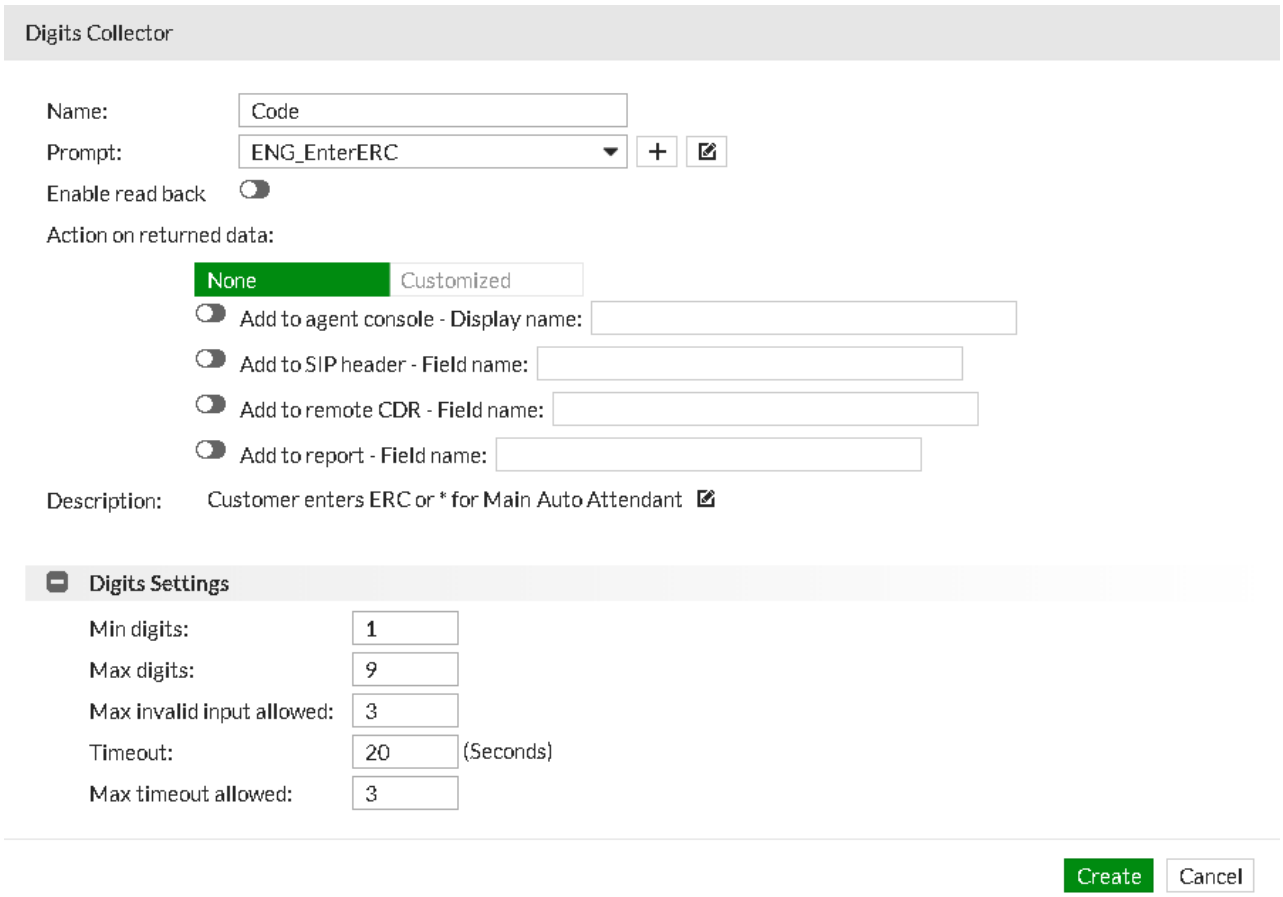
IVRs can be configured to use one or multiple data collectors, depending upon the requirement. This example incorporates the use of the digits and RESTful collector.

Configuring the IVR is made up of the following procedures:

- Configure the digits collector on page 11
- Configure the RESTful collector on page 12
- Configure IVR handling on page 14
- Configure exception handling on page 15

Configure the digits collector

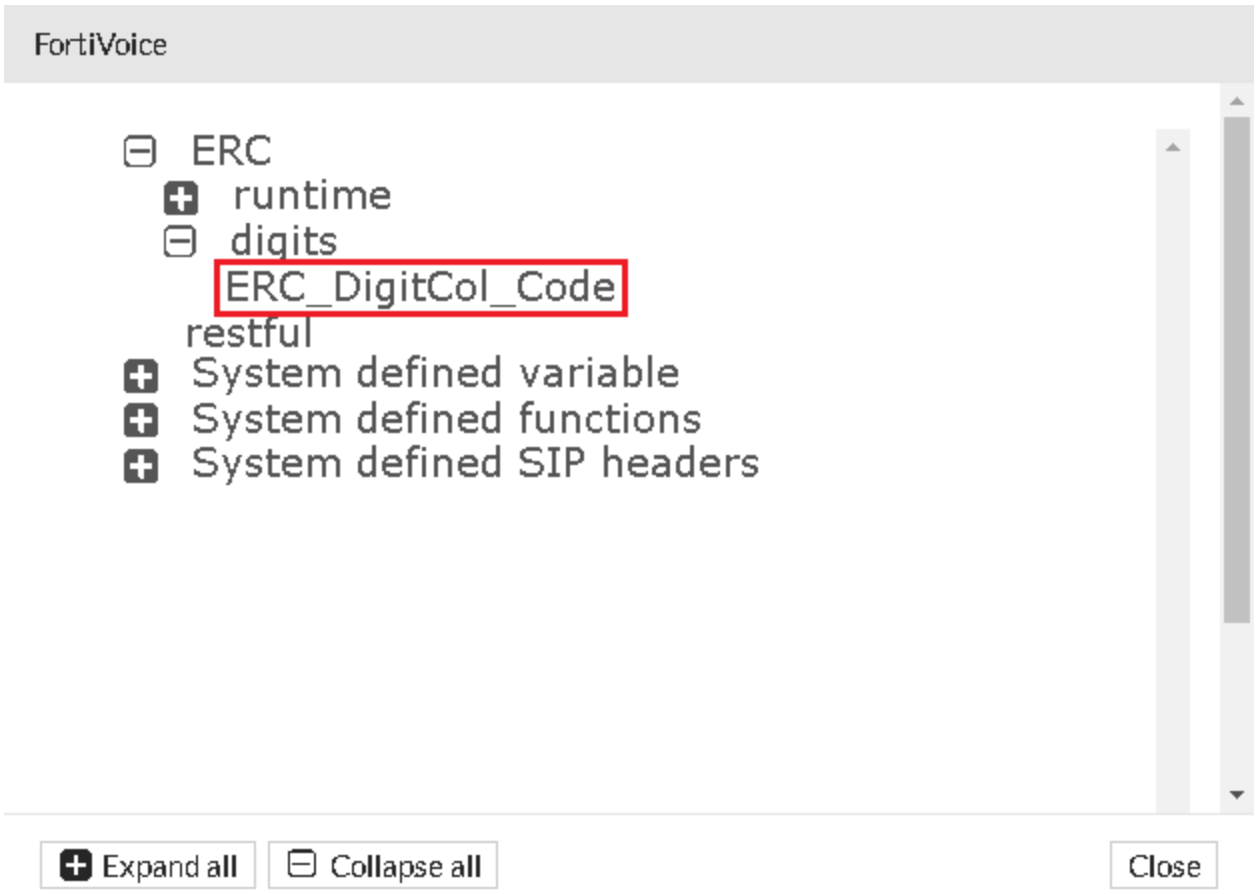
1. Go to *Call Center* > *IVR* > *IVR* and edit the ERC IVR. The *IVR Configuration* dialog opens.
2. Select *Add Digits Collector*.
3. Enter a *Name* (in the example, *Code*), assign an audio *Prompt* for the customer to be asked to enter their ERC, and configure the *Digits Settings* as necessary.



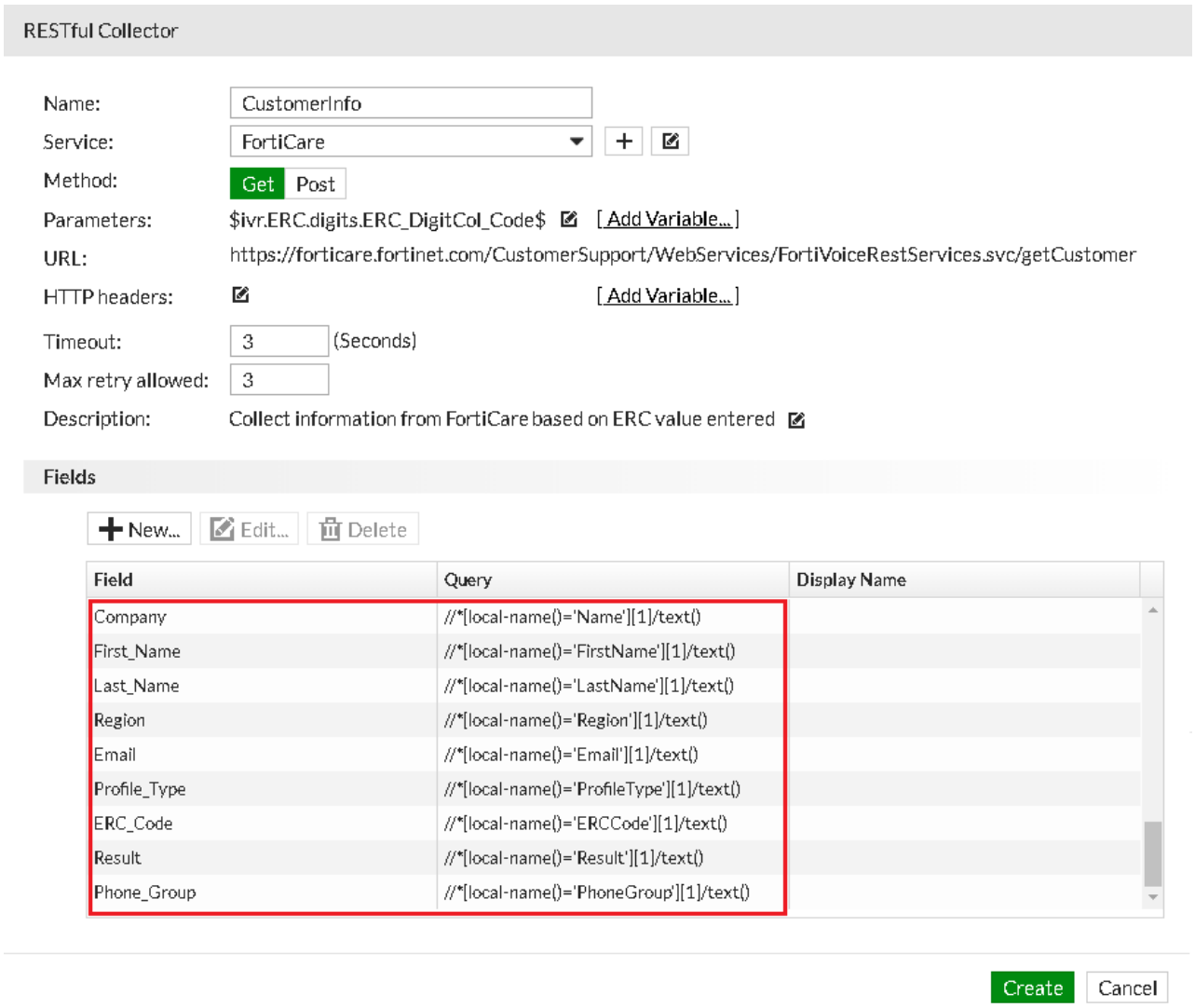
4. Select *Create* to return to the *IVR Configuration* dialog.

Configure the RESTful collector

1. Select *Add RESTful Collector*.
2. Enter a *Name*, and select the FortiCare RESTful service configured earlier from the *Service* dropdown menu. This will automatically fill the *URL* field.
3. For the *Parameters* field, select *Add Variable*.
4. A tree of system defined and user defined options are available. Assign the digit collector configured earlier, as shown.



5. Under *Fields*, select *New*.
6. Set the *Field* name to *Company* and enter the following XPath query:
   `//*[local-name()='Name'][1]/text()`
7. Define the remaining fields as required. When complete, your *RESTful Collector* dialog should look similar to this:



These are examples of some typical fields. However, it is specifically the *Phone_Group* field that will be utilized in the IVR handling for successful call routing.

8. Select *Create* to return to the *IVR Configuration* dialog.

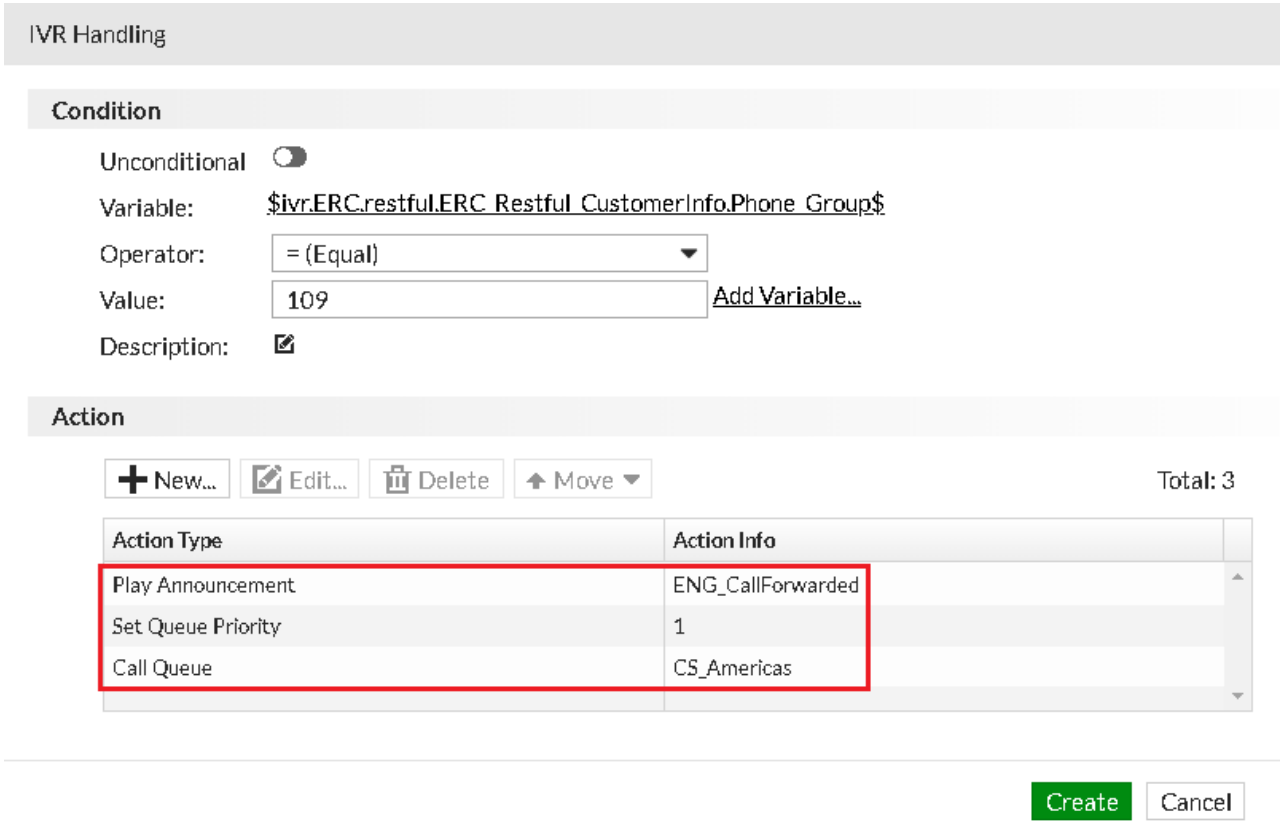
Configure IVR handling

All collector configurations can only take effect after IVR handling is set up. IVR handling can be configured to function unconditionally, in which case the system default conditions apply, and a predetermined action will take effect.

IVR handling can also be configured to function conditionally, where self or system defined variables must be met before an action can take effect. These variables offer varying degrees of control and functionality.

1. Select *Add IVR Handling*.
2. Disable *Unconditional*, and select *Add* next to *Variable*.
3. Navigate through the IVR tree and select the *Phone_Group* field created earlier.
4. Leave *Operator* set to *= (Equal)* and enter a value as required (in this example, *109*). This value is the user's phone group that will be used to route the call to the appropriate call queue.

5. Under *Action*, select *New*, and create the following three actions as shown below:



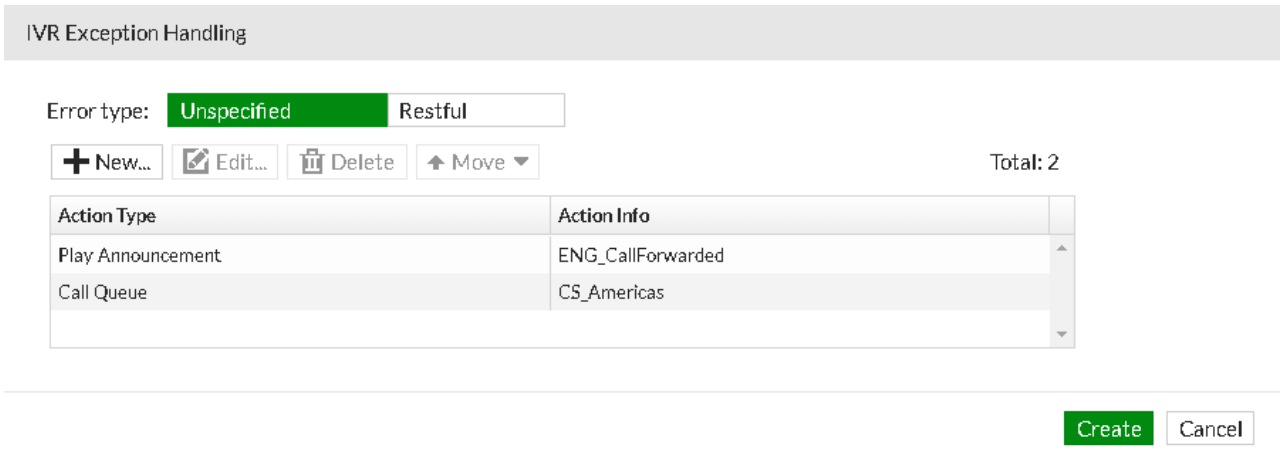
In this example, when a user is prompted to enter their ERC, FortiVoice sends a query to FortiCare to retrieve all information about that specific ERC. If the phone group value within that ERC matches the one specified (109), FortiVoice will sequentially execute the actions in the order they are listed: Play a custom announcement (in the example, a call forwarded announcement), set the call queue to the highest priority (1), and deliver the call to the appropriate call queue (regional customer services).

6. Select *Create* to return to the *IVR Configuration* dialog.

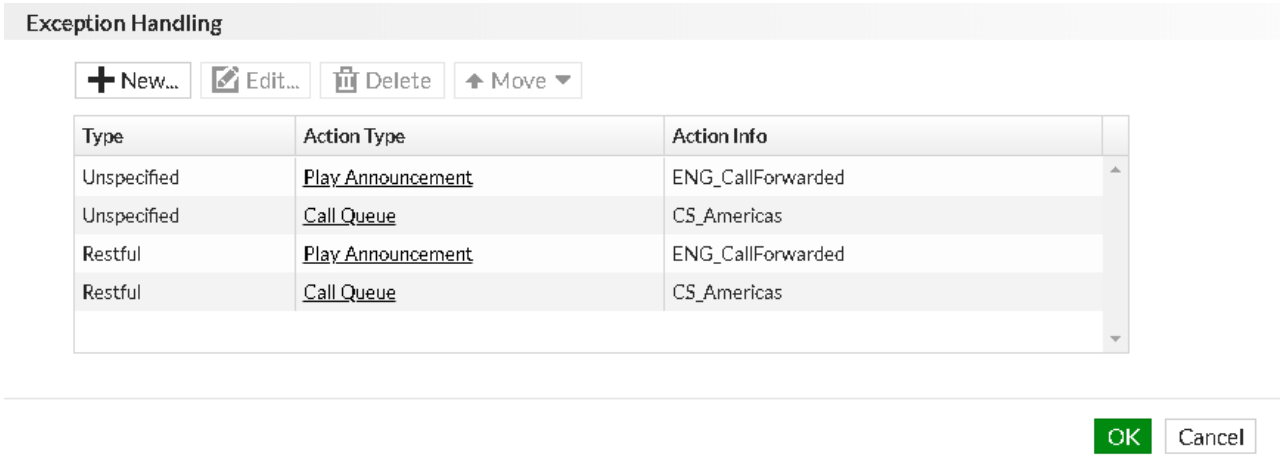
Configure exception handling

If the system encounters an unknown error from the digits collector or the RESTful service, exception handling can be implemented to ensure that calls can still be forwarded to the appropriate call queue.

1. Select *Add Error Handling*.
2. Set *Error type* to *Unspecified*, for unknown errors the system may encounter.
3. Select *New* and create two actions: a call forward announcement followed by an action to route the call to the customer service call queue, as shown below. Make sure to select *Create* when finished.



4. Select *New* under *Exception Handling* and this time set *Error type* to *Restful*.
5. Create the same two actions as before, and select *Create* when finished. You will have four exception handling actions, as shown below.

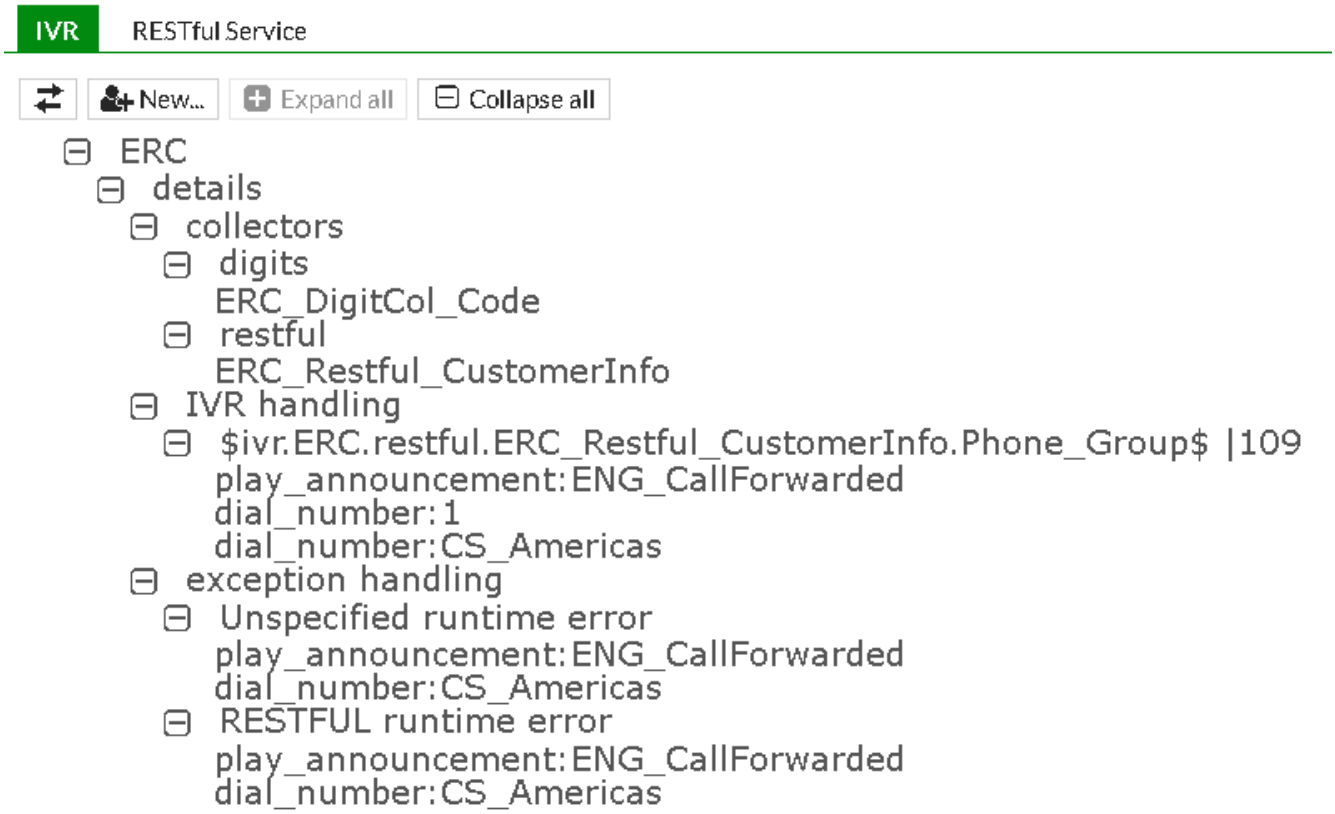


6. Select *OK*. The IVR configuration is complete.

Summary

By the end of this example configuration, several elements come together to perform a sequential pattern of actions, based on the data input by the caller and the conditions and actions applied within the IVR configuration.

Go to *Call Center* > *IVR* > *IVR* and select *Switch* (if necessary) to switch to the expandable/collapsable screen. Here you can view the IVR tree that has been created. When fully expanded, this is what the full IVR tree looks like:



You can select any child-element of the IVR tree to immediately open the options dialog for that particular element.