

# MiCollab Advanced Messaging 9.4 UCConnect® UCROUTE Sample Script Reference Guide

## Developer Resources Document

For version 9.4 and above

## Notice

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). Mitel makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

## Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at [legal@mitel.com](mailto:legal@mitel.com) for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

© Copyright 2022, Mitel Networks Corporation

All rights reserved

# Contents

<b>Overview</b>	<b>4</b>
<b>Operating Modes</b>	<b>5</b>
Prompt for Input	5
Calling Party	5
Caller Language	5
<b>Database Support</b>	<b>6</b>
Routing Table Data Fields	6
Routing Log Table Data Fields	7
Wildcard Characters	7
Call Processor Parameter Combinations	8
<b>Application Phrases</b>	<b>9</b>
Phrase Listing	9
Phrase Implementation	10
<b>Application Configuration</b>	<b>11</b>
Configurable Settings	11
<b>Application File Listing</b>	<b>12</b>
Folder: UCConnect\Data	12
Folder: UCConnect\Script	12
Folder: UCConnect\Speech\UCRoute	12
Folder: UCConnect\Source\Samples\UCRoute	13
<b>Implementation Steps</b>	<b>14</b>

# Overview

The UConnect Call Routing sample script, UCROUTE, is a sample script included with the UConnect software module. In addition to being a robust sample application, it provides functionality that is useful “out of the box.”

The UConnect software module must be properly installed and working on the MiCollab Advanced Messaging (MiCollab AM) system in order for the application to function.

The UCROUTE application allows calls to be routed based on calling party data received through the telephone switch integration as well as caller provided DTMF input and the language that is in use within the MiCollab AM call processor at the time the application is launched. The application can be called at any point within a call processor tree to facilitate custom routing at the required point in the call flow.

Regardless of the data value used to direct the routing, the application queries a database table looking for a match for the data value. Pattern matching for single digits and digit strings of any length is supported in the database. If a match is found, a customer-configured string of digits is returned to the next call processor to direct further call processing. If a match is not found, a customer-configured “no match” string of one or more digits is returned.

A local Microsoft Access database file is used to store application data, including routing data and call log data. The database table formats are fixed. End-users are responsible for populating the routing table and accessing the call log table in a manner of their choosing; the application does not provide any tools for this purpose.

The Visual Basic.NET source code for the application is provided. Developers familiar with this programming language can customize the application as needed for particular implementations. In order to customize the application, the developer must have Microsoft Visual Studio 2008 or higher and the UConnect Developer software version 5.0 SP2 or higher installed on their workstation.

# Operating Modes

Three different operating modes are supported by the application. The operating mode to use at a particular point is specified by passing a parameter from the launching call processor mailbox.

## Prompt for Input

*Call Processor Parameter:* "P"

In this operating mode, the caller is prompted for input and responds by pressing keys on their telephone keypad (i.e. DTMF). The caller entry is used to identify the routing digits to be returned to the next call processor.

The application includes several pre-recorded phrases for common input types (see below).

Callers can be prompted to confirm their entries, and can be given multiple attempts to supply an entry that matches a routing record in the database. The number of attempts is configurable.

## Calling Party

*Call Processor Parameter:* "N"

In this operating mode, the complete MiCollab AM integration-provided calling party data is used to identify the routing digits to be returned to the next call processor.

The ability to specify partial matches in the database using wildcard characters allows routing to be accomplished using only a part of the calling party data, such as the area code.

## Caller Language

*Call Processor Parameter:* "L"

In this operating mode, the MiCollab AM language in effect when the script is executed is used to identify the routing digits to be returned to the next call processor.

The language index number used for the lookup is the same as the language number that would be specified when executing the "Language" action in a call processor mailbox. That is, the first language in the MiCollab AM "selected languages" list has an index number of 0, the second language in the list has an index number of 1, and so on, incrementing the index number by one for each entry in the list.

# Database Support

The database provided with the application is a local Microsoft Access (2002-2003) format database file. Two tables within the database are utilized by the application. Customers are responsible for interacting with the database tables using their preferred method; an import or administration capability is not provided with the sample application.

The database table containing routing data is designed to support multiple data sets, or segments. Segments are used to isolate different types of data for different lookups, for example; telephone numbers, ZIP Codes, account numbers, etc. By passing a segment number to the application in the call processor parameter, lookups are limited to a single segment.

In addition, a database table to contain call log data is included. The call log table contains a record for each lookup attempt. In most cases, this will result in a single record. However, multiple records may be generated during a single call in cases where callers are prompted for input and multiple attempts are allowed.

## Routing Table Data Fields

Table 1. Routing Table Data Fields

Name	Type	Description
Segment	Integer	Data set segment number.
DataValue	Text	Digit string matched against the lookup value, optionally including wildcards.
ReturnDigits	Text	Digit string returned to the next call processor.

## Routing Log Table Data Fields

Table 2. Routing Log Table Data Fields

Name	Type	Description
LogDateTime	Date/Time	Date and time of log record.
Line	Integer	MiCollab AM line number on which call occurred.
Mode	Text	Script operating mode.
Segment	Integer	Data set segment number.
InputValue	Text	Input data value used to query database.
DataValue	Text	Data value identified as match to InputValue.
ReturnDigits	Text	Digit string returned to the next call processor.
Result	Integer	Query result; 0=No match, 1=Match, 2=Error

## Wildcard Characters

The application supports the inclusion of two wildcard characters in the routing data table field used to find a match for the lookup value (DataValue field). Placement of the wildcard characters is only valid at the end of the data string following any numeric digits. The characters are:

- ?** Match any single digit. Can be included one or more times at the end of the digit string.
- \*** Match any number of digits (including none). Can be included one time at the end of the digit string.

For example:

- 123?** Matches input of any four digit number starting with 123.
- 123??** Matches input of any five digit number starting with 123.
- 206\*** Matches input of any digit string starting with 206 and followed by zero or more digits.
- ???** Matches input of any three digit number.
- \*** Matches input of any digit string of any length.

Note that the \* wildcard match can be used to override the ReturnDigitsNoMatch configuration setting (see below) for cases where a matching data value is not found.

## Call Processor Parameter Combinations

Two call processor parameters are supported. The parameters are separated by a single space character. The first parameter is the operating mode, and the second is the database segment to use for the lookup.

For example:   Open Script    "UCROUTE N 1"

If an invalid parameter is supplied an error message will be spoken to the caller and an error description will be written to the Windows Application event log.

If the database segment parameter is omitted the application will search segment "0".

If the database segment specified does not exist, the application will not find a match and the customer-configured "no match" digit string will be returned.



# Application Phrases

The application includes several pre-recorded phrases that can be prompt callers for input. End-users can re-record these phrases, or implement other phrases, as desired.

## Phrase Listing

Table 3. Phrase Listing

Phrase Name	Phrase Content
ConfirmPrompt	"If this is correct, press 1. If this is not correct, press 2."
ConfirmPrompt0	"If this is correct, press 1. If this is not correct, press 2." (Database segment 0).
EnterAccount	"Please enter your account number, followed by the pound sign."
EnterAreaCode	"Please enter your telephone number area code, followed by the pound sign."
EnterID	"Please enter your ID number, followed by the pound sign."
EnterTelephone	"Please enter your 10-digit telephone number, followed by the pound sign."
EnterTelephoneAlt	"Please enter your 7-digit telephone number, followed by the pound sign."
EnterTracking	"Please enter your tracking number, followed by the pound sign."
EnterZIPCode	"Please enter your 5-digit ZIP Code, followed by the pound sign."
ErrorGeneric	"We're sorry. The automated system is unable to process your request at this time."
InputPrompt0	"Please enter your 10-digit telephone number, followed by the pound sign." (Database segment 0).
InvalidEntry	"That is not a valid entry."
InvalidEntry0	"That is not a valid entry." (Database segment 0).
ThankYou	"Thank you."
TryAgain	"Please try again."

## Phrase Implementation

The phrases listed above are placed in the script speech folder during installation.

End-users can replace any or all of the phrases by creating a new phrase file with the same file name and audio format (G.711 u-law), and copying the new file to the "UCConnect\Incoming\Speech\UCRoute" folder while the UCConnect service is running.

The "Prompt for Input" mode requires the implementation of phrases for each specific database segment utilized. Example prompts for database segment 0 (zero) are included in the installation.

The phrases required for each database segment in use are listed below. In this listing, the "X" character represents the positioning of the database segment number within the phrase file name.

- ConfirmPromptX
- InputPromptX
- InvalidEntryX

For example, the phrases required when using database segment five (5) are:

- ConfirmPrompt5
- InputPrompt5
- InvalidEntry5

End-users can implement the required phrases by copying and renaming existing phrase files, or by recording new phrases and copying the new phrase files to the incoming location specified above.

Note that if the ConfirmPromptX phrase is not present, the input confirmation will not take place.

For additional information on recording and implementing UCConnect phrases, refer to the UCConnect Getting Started online book contained on the MiCollab AM DVD.

# Application Configuration

All configurable application settings are contained in a standard .NET application configuration file, named UCROUTE.exe.config. This file resides in the same folder as the script executable (\UCCconnect\Script). The file consists of XML statements and can be edited with any text editor.

## Configurable Settings

Table 4. Configurable Settings

Setting Name	Description (Default)
DBConnectionString	OleDb database connection string (default = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=..\Data\UCROUTE.mdb").
MaxInputRetries	Maximum number of input retries to allow in "Prompt for Input" mode (default = 2).
ReturnDigitsHangup	Digit string returned to call processor if caller disconnects (default = "B").
ReturnDigitsNoMatch	Digit string returned to call processor if no match is found in the database (default = "C").

# Application File Listing

The following list of files will be placed into the specified production folders during UCConnect software module installation.

## **Folder: UCConnect\Data**

UCROUTE.mdb

## **Folder: UCConnect\Script**

UCROUTE.exe

UCROUTE.exe.config

## **Folder: UCConnect\Speech\UCRoute**

ConfirmPrompt.wav

ConfirmPrompt0.wav

EnterAccount.wav

EnterAreaCode.wav

EnterID.wav

EnterTelephone.wav

EnterTelephoneAlt.wav

EnterTracking.wav

EnterZIPCode.wav

InputPrompt0.wav

InvalidEntry.wav

InvalidEntry0.wav

ThankYou.wav

TryAgain.wav

YouEntered.wav

## Folder: UCConnect\Source\Samples\UCRoute

app.config

AssemblyInfo.vb

UCROUTE.vbproj

UCScript.ico

UCScript.vb

UCScriptConfig.vb

UCScriptData.vb

UCScriptMain.vb

# Implementation Steps

To prepare to use the application for inbound callers, the following steps need to be completed.

Note that these steps will need to be performed for each different operating variation to be implemented.

## To implement an operation:

- 1 Create a call processor to launch the script with the appropriate parameters. Refer to the [Call Processor Parameter Combinations](#) section above.
- 2 Create a next call processor to handle the return digits from the script. The next call processor will need to handle any values returned from the database, as well as the ReturnDigitsHangup and ReturnDigitsNoMatch values documented in the [Application Configuration](#) section above.
- 3 Put data records in the RoutingData table in the application database. Take care to specify the appropriate Segment value. Implement values with wildcards as desired according to the [Wildcard Characters](#) section above.
- 4 If callers will be prompted for input, modify application prompts as appropriate. Refer to the [Application Phrases](#) section above for more information.