Mitel MiContact Center Enterprise

MEDIA INTEGRATION – OPERATION INSTRUCTIONS

Release 9.2

NOTICE

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). 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.

TRADEMARKS

Media Integration – Operation Instructions
Release 9.2 – November 2016

INTRODUCTION

This document describes various media options for your MiCC Enterprise Contact Center.

PERSONAL GREETING CONFIGURATION

For inbound and outbound calls, a recorded greeting per service group or agent can be played. The greeting must be configured in Configuration Manager.

Note: The personal greeting file must be recorded in the following format: 8 kHz, 16 bit PCM Mono.

At Agent start-up the configured file(s) will be downloaded from the MiCC Enterprise server. If downloading fails for any of the greeting messages, the greeting will not be available, and a message will be written to the application log file.

This feature is only available for agents running with the Softphone option.

SERVICE GROUPS

Each service group of type Campaign, Voice, or Voice – Manual Routing has a configuration item for a greeting option. A single greeting for the entire service group can be played by specifying the name of the greeting file. This is done from the **Personal Greeting** tab of the Service Group Properties dialog box.

Separate greetings can be defined for incoming and outgoing calls. The greeting will be applied to all agents handling calls for the service group, unless the agent has a personal greeting defined.

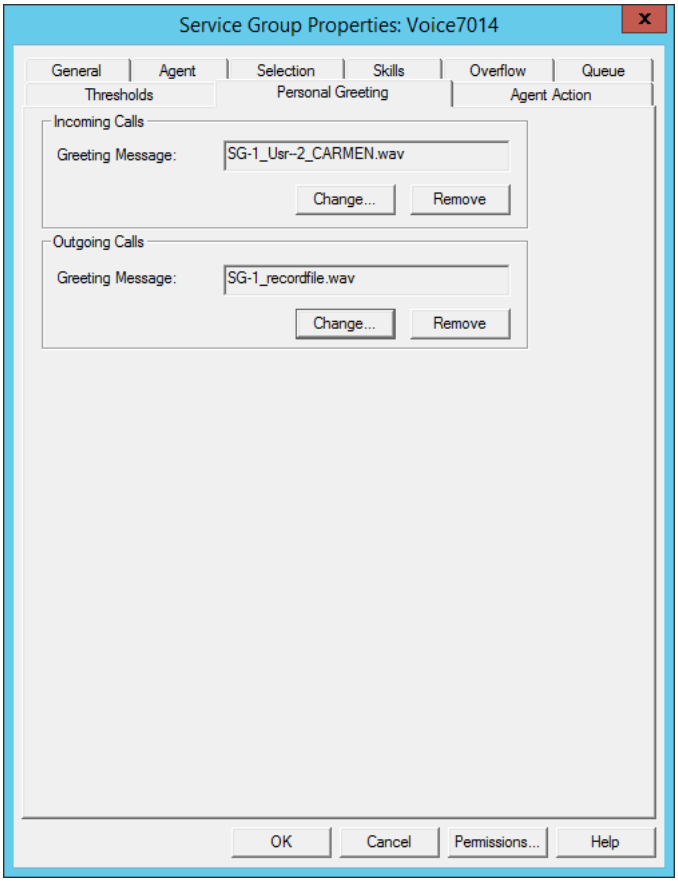


Figure 1: Personal Greeting tab of Service Group Properties

To select a file to use as personal greeting for incoming or outgoing calls:

1. Press **Change**. A dialog box with existing wave files opens.
2. Select the file to use as the greeting file and press **OK**. The file name will now show in the non-editable field for Greeting Message file names.
3. Press **OK** to upload the file to the MiCC Enterprise server. When a greeting file has been configured for a service group and uploaded to the server, the file name will be stored together with other settings for this service group. When an agent logs on to Agent, the configured greeting files for the service groups that the agent serves will be available.

Note: To make it possible to use the same greeting for several service groups, the greeting file name will be slightly modified when uploaded to the MiCC Enterprise Server. The new name is

the name shown in the Greeting Message field. For example, a file named `personal_greeting.wav` will be renamed to `SG-2_personal_greeting.wav`. The name indicates that the uploaded file is meant for a service group with ID = 2.

AGENTS

It is possible to configure a personal greeting also for agents. In this case, the .wav file configured as the agent's greeting will be played. Specifying the personal greeting file for agents is done from the **Personal Greeting** tab of the User Properties dialog box.

Separate greetings can be defined for incoming and outgoing calls. The greeting will be applied to all service group calls handled by the agent.

The name of the greeting file will be stored with other settings (for example password, last name, personal number and so on) for this user. The greeting file will be available when a user logs on to Agent.

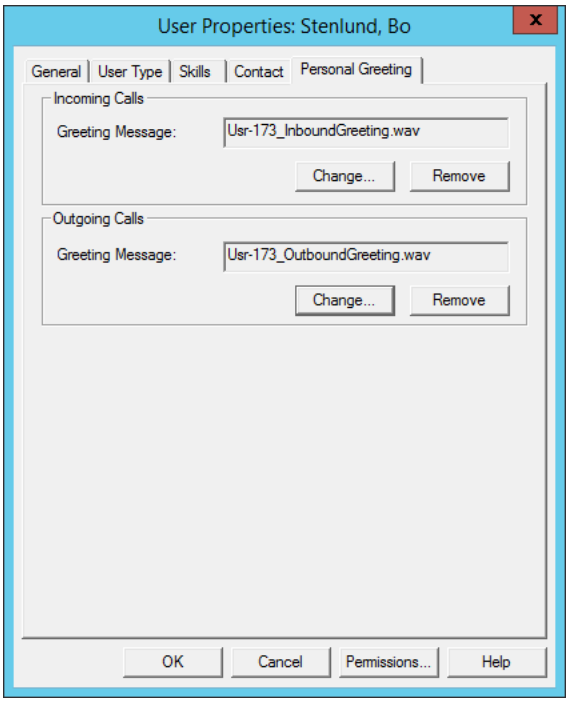


Figure 2: Personal Greeting tab of User Properties dialog box

Note: The greeting file name will be modified when uploaded to the MiCC Enterprise server; the new name is the name shown in the Greeting Message field. For example, a file named `personal_greeting.wav` will be renamed `Usr-1_personal_greeting.wav`. This is to enable different users to upload greeting files with the same name. The name indicates that the file uploaded is meant for a user with ID = 1.

USING FLASH WITH THE AGENT USER DEFINED TABS

The Agent User Defined Tabs use a Chrome-based control to display web pages. The built in Flash player used by Chrome is not supported, so the Adobe Flash player needs to be installed through Google Chrome.

To view a page that uses Flash, please perform the following steps:

- Stop Agent if it is running.
- Install Google Chrome if not already installed.
- Visit http://get.adobe.com/flashplayer/ with Google Chrome browser.
- Download and install the Flash Player.
- Through Google Chrome, check the settings and verify that the Flash Player is enabled.

WEB CALLBACK SETUP

Web Callback can be enabled through MiCC Enterprise, which causes the MiCC Enterprise Router service to poll the SQL database for web callbacks. To enable this option, the following steps must be executed:

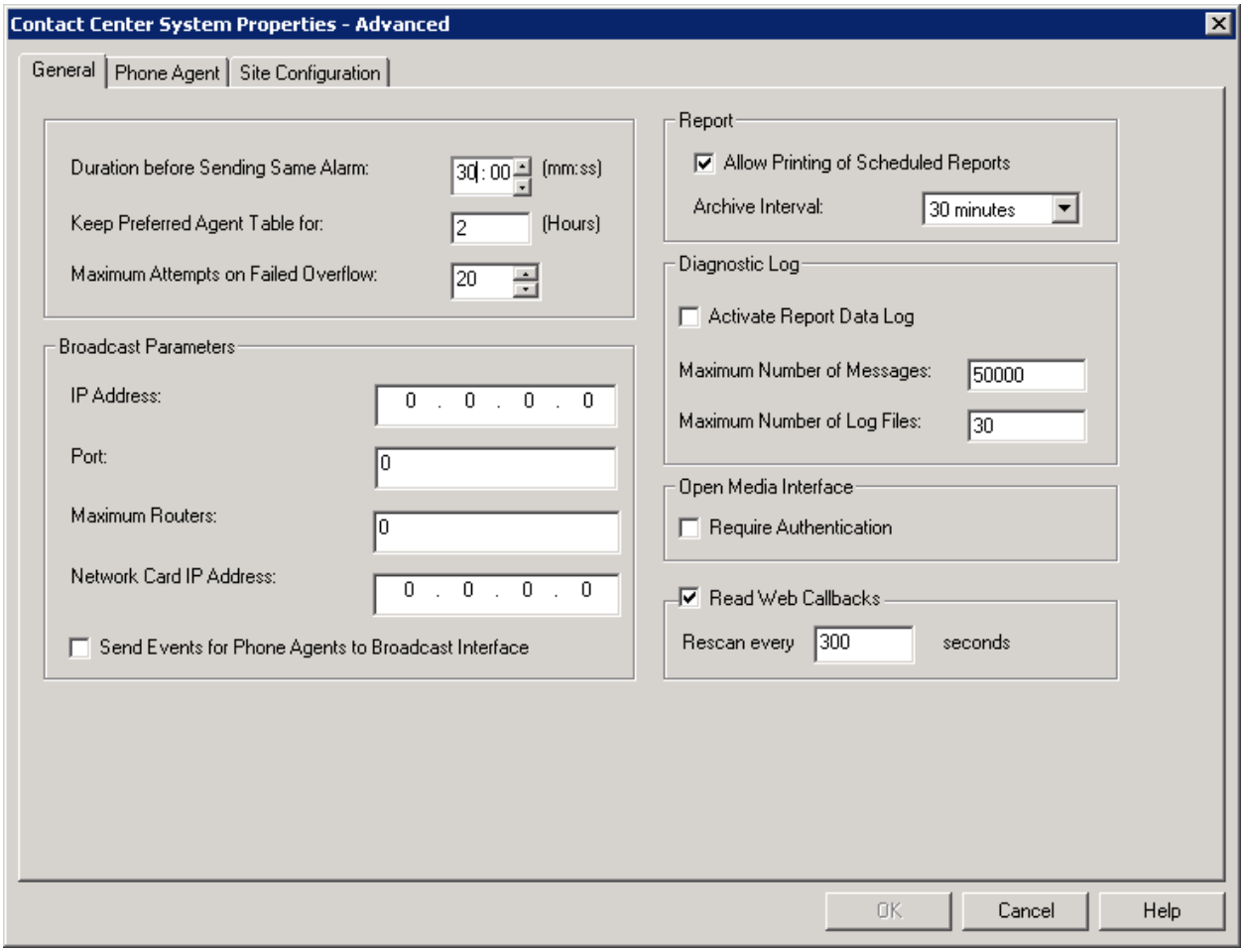


Figure 3: Contact Center System Properties - Advanced

1. From Configuration Manager, on the Contact Center System Properties – Advanced dialog, check the **Read Web Callbacks** check box to enable Web Callbacks. Enter the polling period. The default value is 300 seconds (5 minutes).
2. Make sure that an HTML page where customers can fill in information for call back is available. By default, a sample Callback HTML page is stored on the MiCC Enterprise server under the <installation folder>/Services/Web/WebCallback folder.
3. Determine to which Service Group the web callback is going to be allocated. By default the web callbacks are allocated to the service group that is configured as “Default Service Group for Web Callback” (see “Tenant” configuration in Configuration Manager online help for how to configure a default service group for web callback). Make sure that the service group is configured as a Voice or Voice – Manual Routing type service group. Furthermore, in order for the agents of that service group to receive Web Callback calls, they must be assigned the *Answer Service Calls* privilege and be Ready for voice calls.

4. In order to allocate the web callbacks to a specific service group perform the following steps:

   Determine the identifier of the desired service group by querying the nextccdb SQL database using for example the SQL query analyzer: SELECT id FROM service_grp WHERE name='<name of desired service group>'

   If the sample Callback pages are used: Use the id as input in the URL that calls the CallBack.asp page. A sample page, test.htm, is included in the WebCallBack folder. It is a Java script that will call the CallBack.asp with a parameter:

   call-back.asp?sgid=<service group id>

   The CallBack.asp page calls the CallbackSrv.asp page and passes along the sgid parameter in the form. The CallbackSrv.asp page will write the web callback request to the WebCallback table in the nextccdb database. The desired Service group id will be stored in the service_group_id column in the table.

5. In order to allocate the web callback to a preferred agent in the service group, perform the following steps:

   Determine the identifier of the desired agent by querying the nextccdb SQL database using for example the SQL query analyzer: SELECT id FROM cc_user WHERE last_name='<last name of desired agent>'

   If the sample Callback pages are used: Use the id as input in the URL that calls the CallBack.asp page. A sample page, test.htm, is included in the WebCallBack folder. It is a Java script that will call the CallBack.asp with a parameter:

   call-back.asp?agid=<agent id>

   The CallBack.asp page calls the CallbackSrv.asp page and passes along the agid parameter in the form. The CallbackSrv.asp page will write the web callback request to the WebCallback table in the nextccdb database. The desired Agent id will be stored in the preferred_agent_id column in the table.

   Example: if the Web Callback is targeted for the Service Group with ID 2 and the preferred agent has ID 14 then the call to the Callback.asp page could look like this:

   win=window.open(".\\Call-Back.asp?sgid=2&agid=14","WIN","scroll-bars=yes,resizable=yes,status=no,toolbar=no,location=no,menu=no,width=536,height=600");

WEB CALLBACK FIELDS

Following is a description of the fields in the web_callback SQL database table that can be used with the Web Callback feature.

FIELD	DESCRIPTION
name	Indicates the name of the customer to be called. This will be displayed to the agent.
phone_num	Indicates the phone number to be called by the agent. If configured, number translation can be applied to the number when it is presented to the agent.

FIELD	DESCRIPTION
service_grp_id	Indicates the record ID of the service group to which the web callback will be queued. This must be a Voice, or Voice Manual Routing service group. This field can be kept as NULL to use the default web callback service group, as defined in Configuration Manager.
preferred_agent_id	Indicates the record ID of the agent to which the web callback should be routed. If the agent is not logged on or not ready to receive voice calls, the callback will not be queued for the preferred agent. If the agent is ready to receive voice calls but busy, the callback will be queued for the agent up to the amount of time defined for the service group in Configuration Manager. This field can be kept as NULL to not specify a preferred agent.
contact_asap	Set to 1 to contact the customer immediately. Otherwise, if the value is 0 or NULL, the contact date/time should be specified in the callback_datetime field.
callback_datetime	If contact_asap is NULL or 0, specify the date and time, in UTC, when the customer should be called back. The web callback will wait until the specified time arrives, and then it will be added to the service group queue so it can be presented to an agent.
comments	Free form text field that will be displayed to the agent when the web callback is presented. Up to 255 characters can be entered.
ivr_label/ivr_data	Up to 3 IVR Label and Data fields can be specified, with up to 100 characters of data in each field. These fields will be displayed to the agent when the web callback is accepted.
processed	Internal field indicating that the MiCC Enterprise Router Service has read and processed the web callback. This field should not be modified.
time_stamp	Indicates the date and time in UTC that the web callback was written to the table. This field should be set to the current date and time in UTC when adding an entry.
priority	If call priority is configured to be used for the MiCC Enterprise system, the priority of the web callback can be specified in this field. Values may range from 1 (highest priority) to 100. Modifying the priority value allows the web callback to be placed in the service group queue ahead of other waiting calls.
call_id	Setting this field indicates that the web callback is a continuation of a previous service group call. See Using Script Manager for Callbacks for more details.

Note: If a web callback already exists in the service group queue for the same caller number, another web callback with the same caller number will not be added to the queue, to avoid calling back a customer multiple times.

WEB CALLBACK COMPLETION

When a callback completes, it will automatically be removed from the web_callback table. To prevent this from happening, run the MiCC Enterprise Registry Configuration application, SeCCfg.exe. From the Router Service tab, check the option **Keep Web Callbacks in Database**. The Router Service must be restarted after changing this option.

ONHOOK CALLBACKS

When an onhook callback is registered in the MiCC Enterprise system, it is maintained by the Router service, as well as written to the MiCC Enterprise database, in the callbacks table. The purpose of this table is to maintain information about pending onhook callbacks in case the Router Service is restarted.

When the Router Service restarts, it will read the information from the callbacks table and add all entries that were registered within the last 24 hours. This will allow the customer to be called back by logged on agents.

Note that if a callback was currently in progress when the Router Service restarted, that callback will be added back into the MiCC Enterprise system, since the callback was never completed. This may result in a customer being called twice.

If a callback already exists in the service group queue for the same caller number, and a new callback attempts to register, the new callback will not be added to the queue, to avoid calling back a customer multiple times.

USING SCRIPT MANAGER FOR CALLBACKS

It is possible to use Script Manager to register a callback or to continue a voice call as a web callback. This allows voice calls entering the Service Group queue to be converted from a live, voice call to an onhook callback, or to a web callback.

REGISTERING A WEB CALLBACK

To modify a voice call into a web callback, the MiCC Enterprise system must be configured to handle Web Callbacks (see section Web Callback Setup above for details). A web callback can be registered by a Script Manager script used for incoming call handling, or for repeat queue handling.

The Script Manager SetCallResult block 

is used to set a call as a web callback.

From the Settings tab in the SetCallResult block, select the option **Continue as Web Callback**. When the call disconnects, the MiCC Enterprise system will maintain the call ID. A Service Group event will not be generated for the original voice call until after the web callback is processed. This allows a single service group event and CDR sequence to be generated for the call and the web callback together.

SAMPLE SCRIPT FOR WEB CALLBACK

A sample script for this functionality is as follows:

- OnCallDelivered block – Receive the incoming voice call

- AllocateResource block – Allocate a Player and Tone Detector resource and answer the call.

- Play block – Play a message asking the caller if he would like to be called back later

- GetDigits block – Retrieve the caller's response

- Condition block – Branch if the caller requests a callback

- SetCallResult block – Set the option **Continue as Web Callback**

- Play block – Play a message confirming that the customer will be called back

- ClearCall block – Clear the original voice call

Next, write the information to the web_callback table in the MiCC Enterprise database. It is important that the call_id field in the web_callback table matches the call ID for the original voice call. This can be retrieved in the script using the $$MediaLib.CallID system variable.

SERVICE GROUP EVENTS

When the Router reads the web callback, if the service group specified in the web callback request is different from the service group associated with the original voice call, a Service Group Overflowed event will be generated for the original call, indicating that the call overflowed out of the group. In addition, the service group associated with the web callback will indicate that the call overflowed in.

If the service group requested to handle the web callback is the same as the original service group, a single event will be generated for the service group at the completion of the web callback. This event will either be a Callback Failed event or a Service Group Call Completed event. In either case, the queue duration will be incremented by the amount of time that the original voice call waited in the queue.

WEB CALLBACK TIMEOUT

If a web callback is not registered with a call ID that matches the original voice call ID within the defined timeout value of 3 minutes, the CDR sequence will be closed. If the web callback is received after the timeout occurs, it will be handled as a new, incoming web callback and a new CDR sequence will be generated.

In addition, a Service Group Call Completed or a Service Group Call Abandoned event will be generated for the original call.

It is possible to modify the default 3 minute timeout by setting the following Registry value to the number of seconds to wait:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\CCRouter\Parameters\CDRDelayForSMCallbacks.

REGISTERING AN ONHOOK CALLBACK

An incoming voice call can be converted to an onhook callback by a Script Manager script, returned to the original service group queue and maintain its current position in the queue. It is also possible to send the call to a different service group.

An onhook callback can be registered by a Script Manager script used for incoming call handling, or for repeat queue handling.

The Script Manager SetCallResult block 

is used to set a call as an onhook callback.

From the Settings tab in the SetCallResult block, select the option **Continue as OnHook Callback**. When the call disconnects, it will be eligible to be routed immediately to an available agent as a callback. A single service group event and CDR sequence will be generated for the call and the callback together.

SAMPLE SCRIPT FOR ONHOOK CALLBACK

A sample script for this functionality is as follows:

- OnCallDelivered block – Receive the incoming voice call

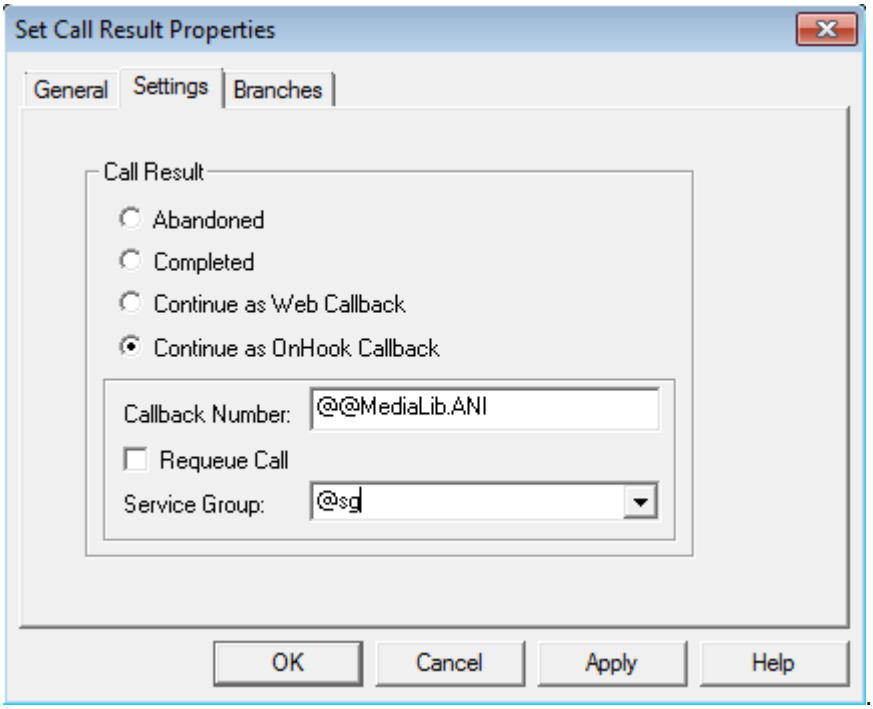
- AllocateResource block – Allocate a Player and Tone Detector resource and answer the call.

- Play block – Play a message asking the caller if he would like to be called back later

- GetDigits block – Retrieve the caller's response

- Condition block – Branch if the caller requests a callback

- SetCallResult block – Set the option **Continue as OnHook Callback**.



Set the **Callback Number** field to the number to be used to callback the customer. To use the customer's calling number, the system variable @@MediaLib.ANI can be entered, as shown above.

Set the **Service Group** field to the name of the Service Group to which the callback will be queued. If the script is used in Repeat Queue Handling, the **Requeue Call** checkbox may be checked indicating that the original service group will be used. Note that the service group must be a voice, non-Dispatch service group to handle the callback.

Note: Both the **Callback Number** and **Service Group** fields are mandatory. If either of these fields is not specified or is empty, the script will fail to compile or fail to execute and the call will not be registered as a callback.

- Play block – Play a message confirming that the customer will be called back

- ClearCall block – Clear the original voice call. It is recommended that the script clears the original call, since the callback cannot be routed to an agent until after the original voice call is cleared.

- ServiceGroup block – This block should be used to handle the *Failure* branch of the SetCallResult block. That will allow the call to continue to be queued at the service group if it fails to be registered as a callback.

SERVICE GROUP EVENTS

If the service group specified in the callback request is different from the service group associated with the original voice call, a Service Group Overflowed event will be generated for the original call, indicating that the call overflowed out of the group. In addition, the service group associated with the callback will indicate that the call overflowed in.

If the service group requested for the onhook callback is the same as the original service group, a single event will be generated for the service group at the completion of the onhook callback. This event will either be a Callback Failed event or a Service Group Call Completed event. In either case, the queue duration will be incremented by the amount of time that the original voice call waited in the queue.

CAMPAIGN CUSTOMER DATABASE IMPORT

MiCC Enterprise supports loading of campaign customers directly into the database. The following describes the fields for the customer related tables.

Please note that loading of the campaign itself directly to the database is not supported.

CUSTOMER FIELDS

The following is a description of the fields in the campaign_customer_res database table that contain the customer information.

When the customer is being processed, the system will update the record and set the locked column to true. The 3rd column indicates which fields may be updated by an external system when the record is not locked.

FIELD	DESCRIPTION	ALLOWED TO UPDATE WHEN NOT LOCKED
Id	Identity column (automatically assigned primary key)	No
campaign_id	The id of the campaign to add the customer to. This can be found in the campaign_params table.	No
last_name	Customer's last name. Up to 30 characters. Only alpha numeric characters are allowed.	Yes
first_name	Customer's first name. Up to 30 characters. Only alpha numeric characters are allowed.	Yes
number	Primary phone number for the customer. Up to 20 characters. Numeric, parenthesis, dashes and spaces are allowed.	No
comment	Comment field visible to agents. Up to 255 characters.	Yes
agent_id	Must be set to 0	No
last_called_time	Must be set to 1970-01-02 00:00:00	No
num_attempts	Must be set to 0	No
call_status_cd	Must be set to 1	No
next_call_cd	Must be set to True	No
next_call_time	Must be set to 1970-01-02 00:00:00	No
answd_dur	Must be set to 0	No

FIELD	DESCRIPTION	ALLOWED TO UPDATE WHEN NOT LOCKED
first_attempt_number	Must be set to the same value as the 'number' column described above	Yes
first_attempt_number_type	Number Type – see table below for a description	Yes
second_attempt_number	Phone number to call for the second attempt to contact. Up to 20 characters. Numeric, parenthesis, dashes and spaces are allowed.	Yes
second_attempt_number_type	Number Type – see table below for a description	Yes
third_attempt_number	Phone number to call for the third attempt to contact. Up to 20 characters. Numeric, parenthesis, dashes and spaces are allowed.	Yes
third_attempt_number_type	Number Type – see table below for a description	Yes
fourth_attempt_number	Phone number to call for the fourth attempt to contact. Up to 20 characters. Numeric, parenthesis, dashes and spaces are allowed.	Yes
fourth_attempt_number_type	Number Type – see table below for a description	Yes
fifth_attempt_number	Phone number to call for the fifth attempt to contact. Up to 20 characters. Numeric, parenthesis, dashes and spaces are allowed.	Yes
fifth_attempt_number_type	Number Type – see table below for a description	Yes
locked	Must be set to False	No
preferred_agent_id	null if preferred agent is not to be set for the customer, otherwise the id of the preferred agent from the cc_user table.	Yes
preferred_agent_timestamp	Must be null	Yes

Number Type

The Number Type is used in the first_attempt_number_type, second_attempt_number_type, etc. fields above.

VALUE	DESCRIPTION
0	Home
1	Office
2	Mobile
3	Other

CUSTOMER CAMPAIGN FIELDS

If Customer Fields have been added to the campaign through Configuration Manager, the customer data for these fields can be imported into the campaign_customer_field table. The following is a description of the fields in campaign_customer_field.

Please note that the locked column from the campaign_customer_res table must be false before making changes to campaign_customer_field

FIELD	DESCRIPTION
campaign_field_id	The id of the campaign_field from the campaign_field table
campaign_customer_res_id	The id of the customer from the campaign_customer_res table
data	Update to 60 characters. Only alpha numeric,dash and spaces are allowed.

INTEGRATING WITH NICE IEX WORKFORCE MANAGEMENT

MiCC Enterprise supports integration with the NICE IEX Workforce Management system. To setup the integration, follow the steps below:

1. Run IEXInterfaceSetup.exe. It is located in the ThirdParty\IEXInterface directory of the MiCC Enterprise installation package.
2. When prompted for the Database Server, enter the name of the machine where the MiCC Enterprise SQL Server database is installed.
3. Configure the following items before running the NICE IEX Workforce Management interface:

   - **Report Interval**
     This item is stored in the Windows Registry under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\CCReport\Parameters\NiceIEXReportInterval.
     The value is configured during the IEXInterface installation with a default value of 900. A value of 900 indicates that the IEX output file will be generated by MiCC Enterprise every 15 minutes (900 seconds), and a value of 1800 indicates that the file will be generated every 30 minutes (1800 seconds). If this value is changed after installation, the MiCC Enterprise Report Service will need to be restarted for the new value to take effect.

   - **NiceIEXReportSharepoint**
     This item is stored in the Windows Registry under HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\CCReport\Parameters\NiceIEXReportSharepoint.
     The value is configured during the IEXInterface installation as the UNC path to where the MiCC Enterprise Report Service will place the report files for processing by the IEXInterface service (i.e. \\LOCALHOST\IEXInterfaceReports). If this value is changed after installation, the MiCC Enterprise Report Service will need to be restarted for the new value to take effect.

   - **IEXInterface.exe.config**
     The following items are configured in the XML file IEXInterface.exe.config. Open this file,

which is located at \Program Files\Mitel\IEXInterface, and edit the items to contain the correct values.

- **LogLevel:** Indicates the level of logging from 0 (lowest level) to 3 (highest level). The default is 3.
- **MaxLogFiles:** Indicates the maximum number of log files that will be generated before the files are overwritten. The default is 10.
- **LogDir:** Indicates the location where the IEXInterface.log file will be stored. To default to the configured MiCC Enterprise services log directory, leave this item empty.
- **ReportFileSharepoint:** Indicates the location where the MiCC Enterprise Report Service will place the generated reports for the IEXInterface service to process. Note that this value must indicate the same directory as is configured in the NiceIEXReportSharepoint Registry value for the MiCC Enterprise Report Service. The default value is \\LOCALHOST\IEXInterfaceReports.
- **FileCheckInterval:** Indicates how frequently the IEXInterface service will scan for new files in the ReportFileSharepoint to be processed. The default value is 60 (seconds).
- **FTPSite:** Indicates the name of the FTP site where the generated files should be placed by the IEXInterface service. The machine name should be prefaced with ftp://.
- **FTPUserID:** Indicates the user name for opening the connection to the FTP site. The default value is anonymous. For authenticated FTP, enter the user ID.
- **FTPPassword:** Indicates the password used for opening the connection to the FTP site. For anonymous FTP access, this value can be left empty.
- **SQLServerMachineName:** Indicates the name of the machine where the MiCC Enterprise SQL Server database resides.
- **Tenanted:** If the MiCC Enterprise system is installed with Tenanting, set this value to True. Otherwise, leave it set as False.
- **DeleteReportFiles:** After successfully sending the generated files via FTP, the IEXInterface service will delete the generated files. To move the files to a backup subdirectory in the ReportFileSharepoint directory instead of deleting them, set this value to False.

4. Configure Report Manager to generate the necessary reports.

- Logon to Report Manager with the Administrator user ID.
- Select **Daily** from the **Scheduled Reports** group.
- Double-click on **Admin.** The Scheduled Reports [IEX-AgentQueue], [IEX-AgentSysPerf] and [IEX-Queue] should be displayed.
- Double-click on the **[IEX-AgentQueue]** report. On the **Contents** tab, select the agents that you wish to report on and move them from the **Excluded Objects** list to the **Included Objects** list. Press **OK.**
- Double-click on the **[IEX-AgentSysPerf]** report. On the **Contents** tab, select the agents that you wish to report on and move them from the **Excluded Objects** list to the **Included Objects** list. Press **OK.**

- Double-click on the **[IEX-Queue]** report. On the **Contents** tab, select the service groups that you wish to report on and move them from the **Excluded Objects** list to the **Included Objects** list. Press **OK.**

Note: The MiCC Enterprise Report Service will generate dates in the locale configured for the account that the Report Service is running under. It is important that the IEXInterface service has the same settings. To achieve this, either run the IEXInterface service under the same account as the Report Service, or logon to Windows using the account that the Report Service is running under, set the locale, and select the option to copy the locale settings to the System Settings.

5. Restart the IEXInterface Service from the Services Control Panel applet.

For details regarding the calculation methods used to provide data to the NICE IEX Workforce Management application, consult the document Mitel-NICE IEX WFM Historical Design Specification.