Mitel MiContact Center Enterprise

ADVANCED CONFIGURATIONS – OPERATING INSTRUCTIONS

Release 9.5 SP2

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

MiContact Center Enterprise Advanced Configurations
Operating Instructions
Release 9.5 SP2 – October 2021

Table of Contents

INTRODUCTION 1

ACCESSING THE WINDOWS REGISTRY EDITOR 1

REGISTRY KEYS 1

REDUNDANCY 3

HOW TO FORWARD MICC-E EVENTS AS TRAP TO SNMP MANAGERS 3

Installing and configuring Windows SNMP Agent 3

AGENT CONFIGURATION 4

CHECK WINDOWS SNMP AGENT WITH SNMP QUERY TOOLS 9

WINDOWS EVENT CONFIGURATION 9

MICC ENTERPRISE EVENT MIB 10

Using MiCC Enterprise Event MIB with HP OpenView 10

APPLICATION COMMAND LINE PARAMETERS 13

General Parameters 14
Agent 15
Configuration Manager 16
Report Manager 16
Tenant Client Configuration Utility 17

CONFIGURATION OF PERSONAL CALL ROUTING 17

Personal Call Routing Without Dispatch 18

Configuring Personal Call Routing 18

INTEGRATING WITH MIVOICE CALL RECORDING 19

MiCC Enterprise Support DLLs 22

Using Call Recording in Agent 22

USING CUSTOMER AUTHENTICATION 22

Overview 22

Configuration 22
Agent Operation 23
Using Customer Authentication with Phone Agents 24
E-MAIL, CHAT AND SMS RESPONSE FILES 24
Format 24
REPLACEABLE IDENTIFIERS IN RESPONSE FILES AND KB RESPONSES 27
CUSTOMER CONFIGURATIONS FOR WEB INSTALLATION AND CLIENT UPDATES 29
Setup.config Format 29
Example 33
Custom Customer Views 35
HOTFIX/INSTALLATION UPDATES 35
New Installation Packages 36
HotFix Updates 36
MiCCEHotFixInstaller 37
Command Line Parameters 37
Log File 37
PASSWORD MANAGEMENT 37
SERVICE SECURITY 37
WSDL Suppression 38
Web Services Running Under IIS 38
WCF Services 38
CALLER ID FOR OUTGOING CALLS 39
Open Application Server 39
Telephony Application Service 39
CALL MANAGER LOAD BALANCING 40
Load Balancing for Phone Agents 40

AGENT DATA ACCESS 40

PHONE AGENT AUTOMATIC LOGON 42

Database Handling 42

OBJECT TAGS 43

AGENT ADVANCED CONFIGURATION 47

TRANSFER OF SERVICE GROUP VOICE CALLS 48

MICROSOFT AZURE INITIAL CONFIGURATION FOR USER SYNCHRONIZATION AND SINGLE SIGN-ON 49

USER SYNCHRONIZATION 49

Configuring Microsoft Azure AD 49

Configuring MiCC Enterprise 51

Synchronization Process 52

Audit Log 54

Execution and Scheduling 55

SINGLE SIGN-ON 56

Configuring Microsoft Azure AD 56

Configuration ADFS (Active Directory Federated Services) 57

Configuring MiCC Enterprise 58

Single Sign-On Process 60

Single Sign-On Using SAML 2.0 in Web Manager 62

INTRODUCTION

This document describes advanced configurations of MiCC Enterprise features. This includes Windows registry keys that can be modified by users.

Note: Caution should be taken when modifying registry keys, as changes made will directly affect the operation of MiCC Enterprise applications and services.

Many registry keys can be set using the MiCC Enterprise Registry Configuration application (SeCCfg.exe), located in the NextCC Setup directory when MiCC-E is installed. This document describes details for keys not configured using SecCfg.exe.

Viewing or changing Windows registry keys associated with MiCC Enterprise are done using the Registry Editor.

ACCESSING THE WINDOWS REGISTRY EDITOR

1. Log on to Windows.
2. Click **Start**, and then select **Run**.
3. Type **REGEDIT**.
4. Click **OK**.

REGISTRY KEYS

The following registry keys can be located in the HKEY_LOCAL_MACHINE window under the `\SOFTWARE\Wow6432Node\Mitel\SEC\Common\Parameters\Services\` subkey.

REGISTRY KEY	VALUE	DESCRIPTION
SeCLogonWS	LogLevel (DWORD)	Determines the level of logging for the Logon Web Service log file. Default is 2. Valid values are: 0= Log errors only 1= Log errors and warnings 2= Log errors, warning and status 3= Log all events
	LogonWSTimeout (DWORD)	This value indicates the number of milliseconds to wait for the Web Service to initialize before failing the logon request. Default value is 30000 ms (30 seconds).

REGISTRY KEY	VALUE	DESCRIPTION
SeCReportWS	LogLevel (DWORD)	Determines the level of logging for the Report Web Service log file. Default is 2. Valid values are: 0= Log errors only 1= Log errors and warnings 2= Log errors, warning and status 3= Log all events

The following registry keys can be located in the HKEY_LOCAL_MACHINE window under the `\SOFTWARE\Wow6432Node\Mitel\SocketManager\` subkey.

All communication between services and applications are using a common component called SocketManager. To isolate problems with the TCP/IP communication between MiCC Enterprise components, enable the following log for the target component.

VALUE	DESCRIPTION
Trace (DWORD)	This value indicates whether the SocketManager logging will be enabled. Default is 0.
<name of the executable> DWORD	Adding a value corresponding to the name of the executable file sets the SocketManager log for the executable. This overrides the Trace settings for the specified executable. For example, adding a registry value called cs.exe, with type DWORD set to value = 9, enables the SocketManager log for the configuration service to level 9.

REDUNDANCY

The recommended and supported solution for warm and hot stand-by is VMWare with High Availability and Fault Tolerance.

For additional information regarding VMware for MiCC Enterprise, see the document *Virtualization Description*.

HOW TO FORWARD MICC-E EVENTS AS TRAP TO SNMP MANAGERS

Windows systems log most system-level events on their own by default without any further administrative action required. This section describes how to reuse SNMP technology already bundled into Windows to generate lightweight alerts against pre-selected events, thus providing the basis for a flexible and scalable notification system that can work with existing network management tools. Network administrators can use the built-in alert system and an SNMP management station to trap critical events and automatically respond to them as soon as they happen.

The Windows SNMP agent has the ability to generate explicit SNMP trap messages from any of the discrete Windows event messages that can be logged. However, the component pieces to enable this functionality are not visible by default.

INSTALLING AND CONFIGURING WINDOWS SNMP AGENT

To install the Microsoft Windows SNMP agent on a Windows 2008 R2 or Windows 2012 R2 Server do the following:

1. Open the Control Panel and select **Programs**.
2. Under **Programs and Features** select **Turn Windows features on or off.**
3. From the list on the left pane, right click on **Features** and select **Add Features.**
4. Select **SNMP Services** from the list and click on the **Next** button.
5. Click on the **Install** button.

Your server may require a reboot after installing the SNMP Services. Once they are installed, proceed to configure the SNMP Agent.

Note: For Windows 2012 R2 Server systems, it may be necessary to add the SNMP Tools feature after installing the SNMP Services. To do this, from Server Manager, select **Manager** then **Add Roles and Features**. Under **Feature** select **Remote Server Administration Tools**

-> Feature Administration Tools -> SNMP Tools. Restart the SNMP Service after installing the SNMP Tools.

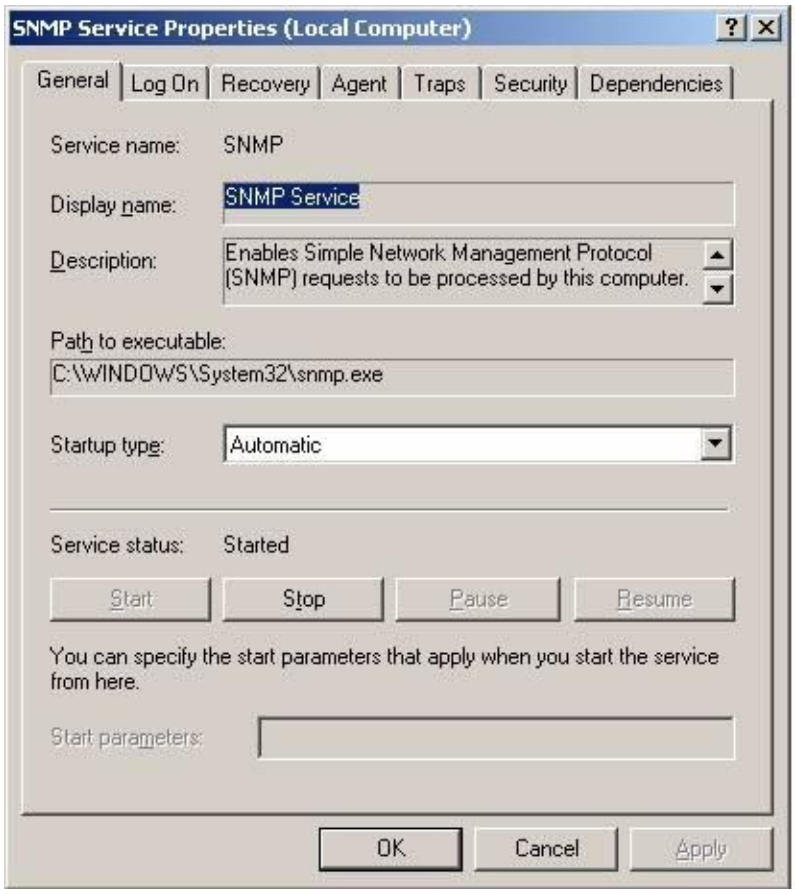
AGENT CONFIGURATION

The configuration of the SNMP service is performed through the Service properties option. To access the Service properties option, do the following:

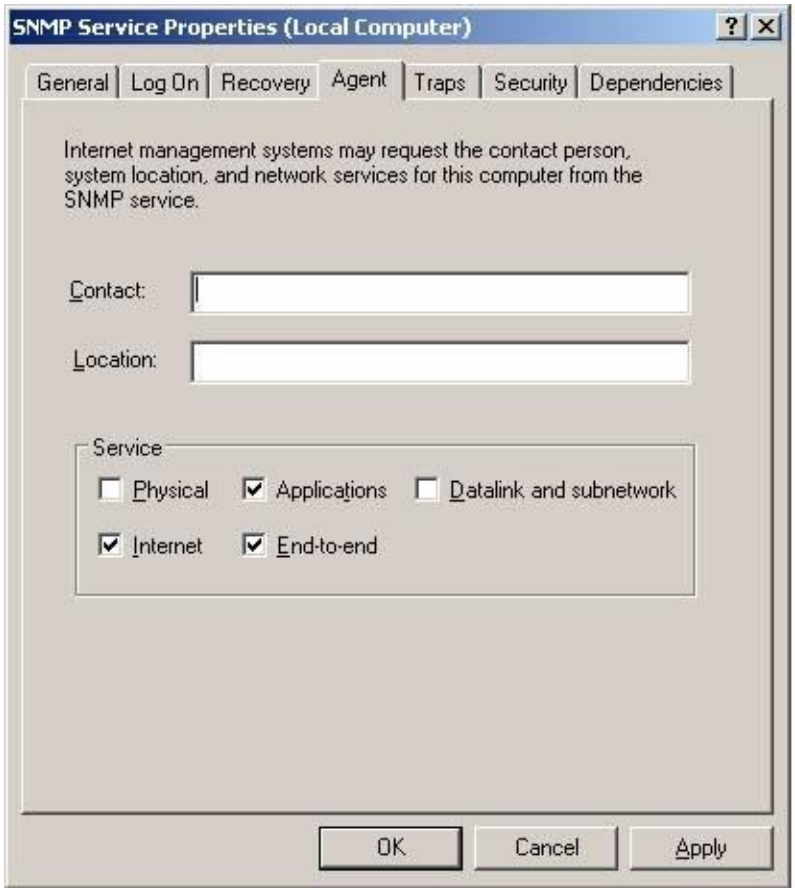
1. Open the Control Panel and select **System and Security**.
2. Select **Administrative Tools**
3. Select **Services** and then double-click the SNMP Service in the Service List.

   The SNMP Trap Service is only used to receive trap. If there is no trap receiver application on this system don't start it.

4. The SNMP Service Properties window is displayed.



5. Choose the **Agent** tab for specification of agent's properties.



The standard mib2 value **syscontact** and **syslocation** can be used.

Contact: Name and contact information of the administrator

Location: Location of the device. Here you can enter address, number of building, floor, room, rack number, and so on.

Services: Select the Agent's advanced properties.

Physical: Computer manages physical devices, hard disk partition.

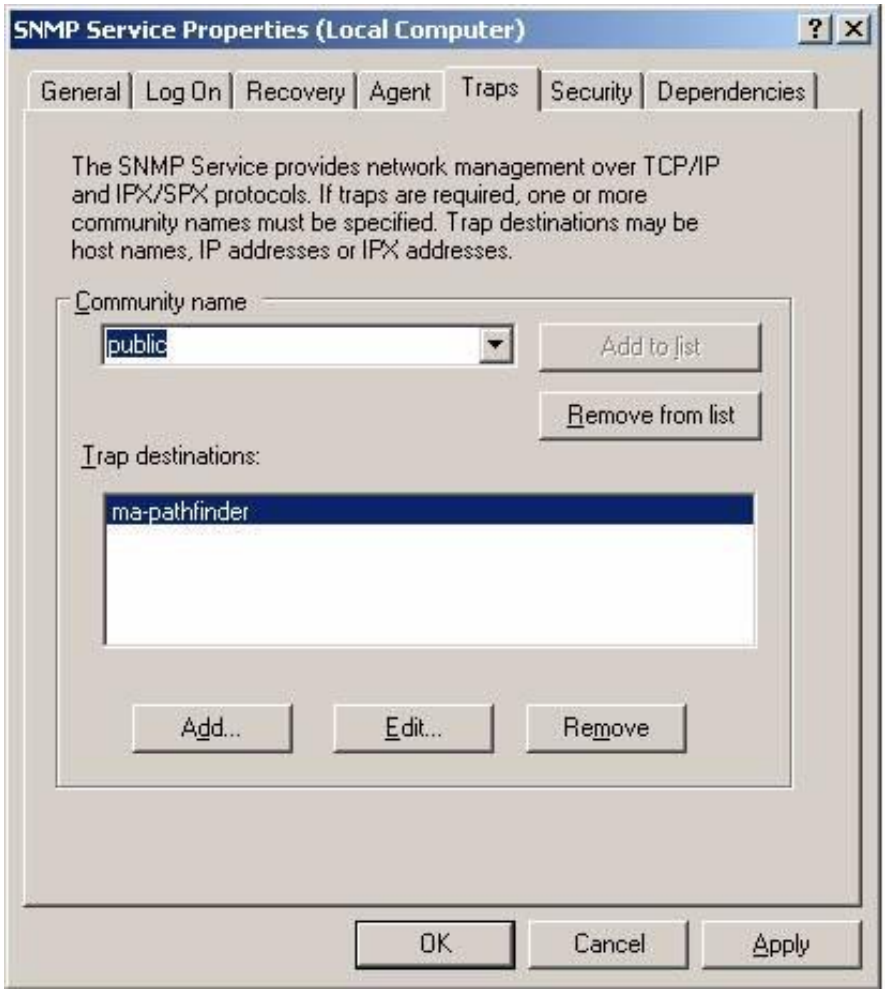
Applications: Computer uses applications which send data by help of TCP/IP protocols. This service should always be enabled.

Datalink and subnetwork: Computer manages bridges.

Internet: Computer works as an IP router.

End-to-end: computer works as an IP host. This service should always be enabled.

6. Click the **Traps** tab.



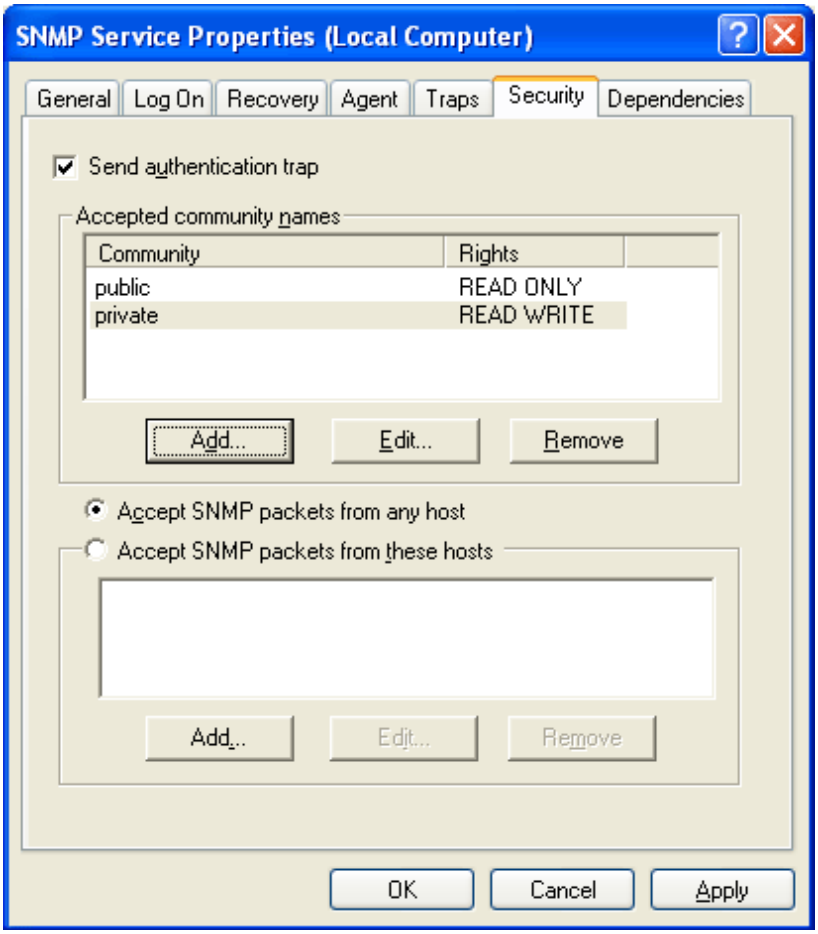
7. In the **Community name** box, type the case-sensitive community name to which this computer will send trap messages, and then click **Add to list.**

   Under **Trap destinations**, click **Add**.

   In the Host name, **IP or IPX address box**, type the name, **IP** or **IPX** address of the host, and then click **Add**.

   The host name or address appears in the **Trap destinations** list. Repeat to add all communities and trap destinations

8. To enable SNMP security, click the **Security** tab.



Accepted Community Names. The SNMP service requires the configuration of at least one default community name. The name Public is generally used as the community name because it is the common name that is universally accepted in all SNMP implemen- tations. You can delete or change the default community name or add multiple community names. If the SNMP agent receives a request from a community that is not on this list, it generates an authentication trap. If no community names are defined, the SNMP agent will deny all incoming SNMP requests.

Permissions. You can select permission levels that determine how an agent processes SNMP requests from the various communities. For example, you can configure the permission level to block the SNMP agent from processing any request from a specific community.

Accept SNMP Packets from Any Host. In this context, the source host and list of acceptable hosts refer to the source SNMP manage- ment system and the list of other acceptable management systems. When this option is enabled, no SNMP packets are rejected on the basis of the name or address of the source host or on the basis of the list of acceptable hosts. This option is enabled by default.

Only Accept SNMP Packets from These Hosts. Selecting this option provides limited security. When the option is enabled, only SNMP packets received from the hosts on a list of acceptable hosts are accepted. The SNMP agent rejects messages from other hosts and sends an authentication trap.

Send Authentication Traps. When an SNMP agent receives a request that does not contain a valid community name or the host that is sending the message is not on the list of acceptable hosts, the agent can send an authentication trap message to one or more trap destinations (management systems).

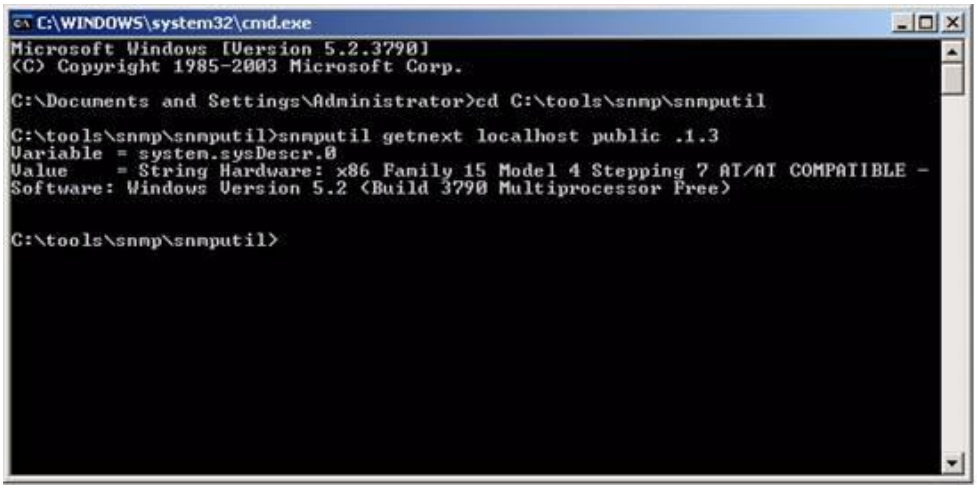
CHECK WINDOWS SNMP AGENT WITH SNMP QUERY TOOLS

The easiest way to check that the agent is working is to use the snmputil tools.

Snmputil.exe is a command line utility (included with the Windows 2000 Server resource kits) that allows the querying of MIB information.

The LAN Manager MIB-II Agent for Windows is installed automatically with the SNMP agent, so it is the most convenient MIB to test against. Following query will query the system description.

Snmputil getnext localhost public .1.3



WINDOWS EVENT CONFIGURATION

The translation of events to traps, trap destinations, or both based on information in a configuration file. This configuration file has been created and named **Solidus_eCare_events.cnf.** This file configures the traps but not trap destinations. **ConfigureSNMPTraps.bat** is a batch file which will execute envtcmd command to quickly configure traps on the target computer.

MICC ENTERPRISE EVENT MIB

This MIB defines traps sent by the event-to-trap function on MiCC Enterprise servers.

The MIB may be imported into an SNMP mangers like HP OpenView, Tivoli or Netview and be used as a starting point of an alarm definition.

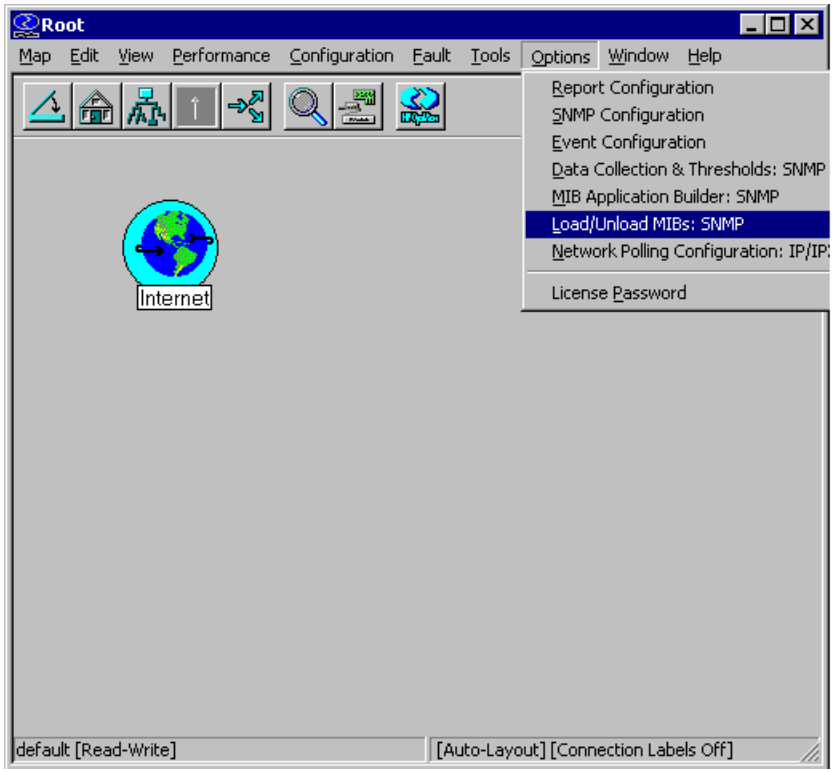
Base Enterprise OID for Event-to-trap notification shall correspond to the value of the Windows Registry name BaseEnterpriseOID located at

`HKLM\SOFTWARE\Microsoft\SNMP_EVENTS\EventLog\Parameters`

This means for MiCC Enterprise traps, BaseEnterpriseOID in Windows Registry must be “1.3.6.1.4.1.193.132.5.1”.

USING MICC ENTERPRISE EVENT MIB WITH HP OPENVIEW

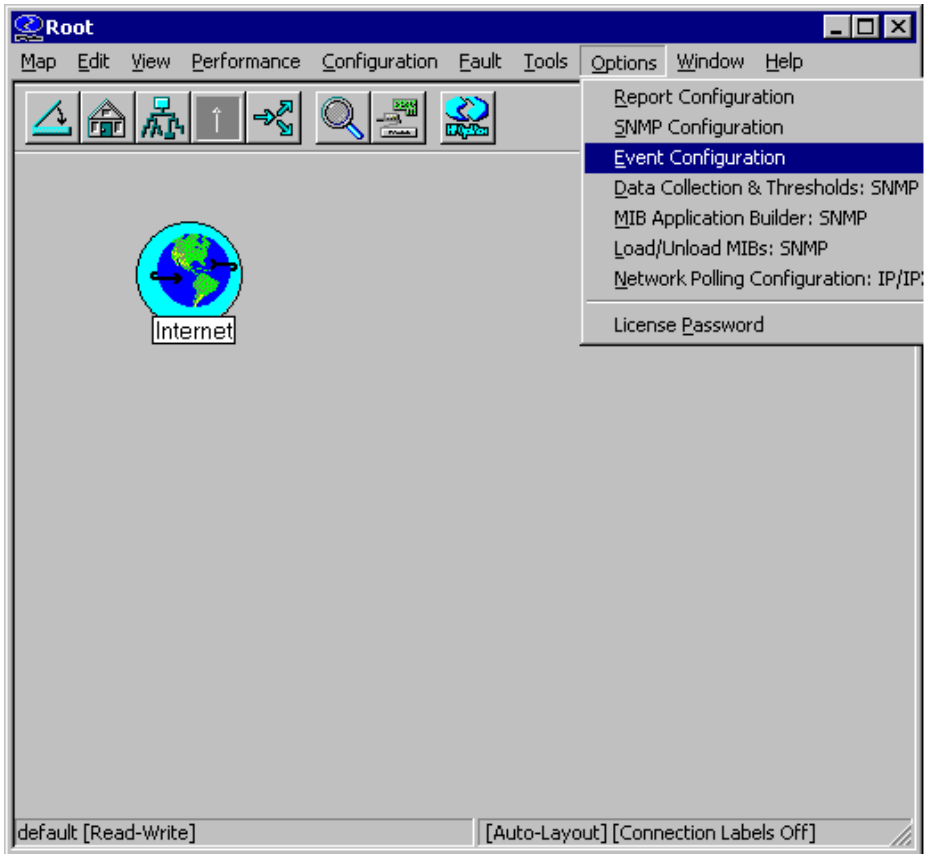
1. To load the MiCC Enterprise Event MIB, copy the MIB on the local drive of Server running OpenView. Select **Load/Unload MIBs: SNMP** from the **Options** menu.



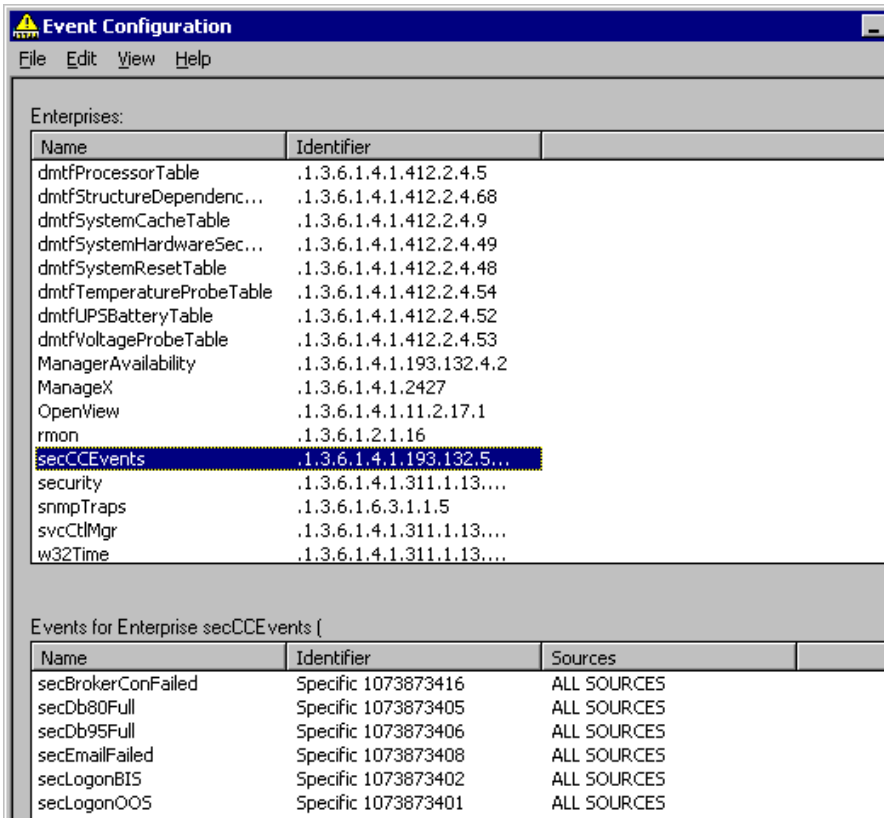
2. Load the MIB by help of this tool.

   After loading the MIB the events should be configured to be presented correctly by OpenView.

3. Open **Event Configuration** from **Options** menu.

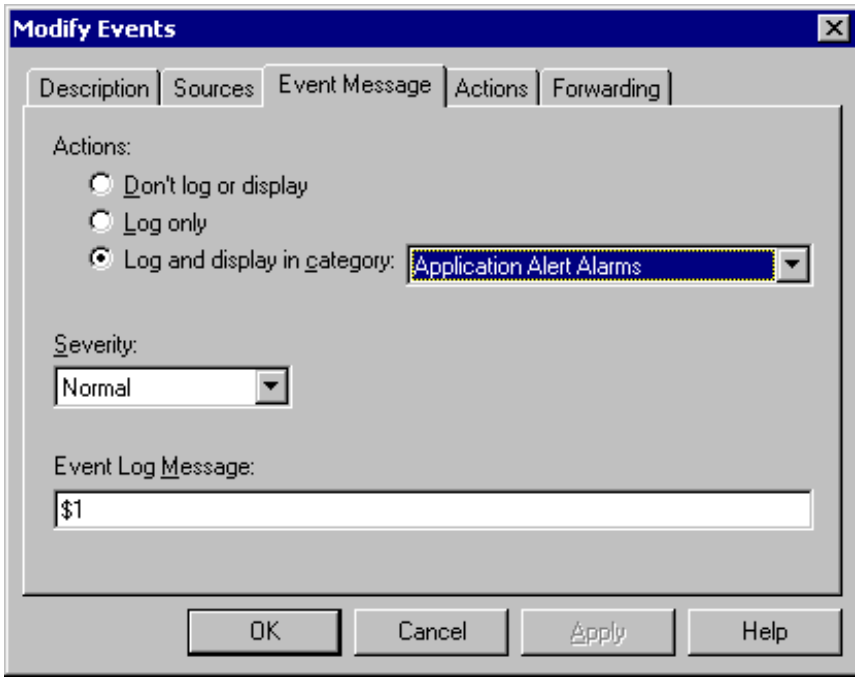


4. Event configuration tools will be presented. In the Enterprise window, find the **secCCEvents**.



5. Open the first event **secBrokerConFailed** from **Events for Enterprise secCCEvents**, by double clicking on the events.

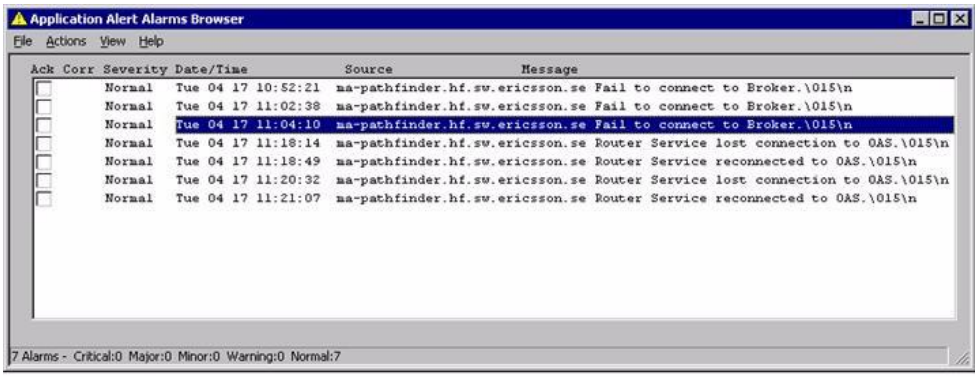
6. Choose the **Event Message** tab and from **Action**: and choose **Log and display in category**: and choose **Application Alert Alarms**. From **Severity**: choose a level accordingly, and change the Event Log Message: to $1.



7. Click **OK** and finish modification for the first event.

   Repeat the previous for all events.

Each time a trap is sent from MiCC Enterprise this is shown in the **Alarm Browser** with source and message.



APPLICATION COMMAND LINE PARAMETERS

Command line parameters may be specified when starting most MiCC Enterprise desktop applications. These applications include:

Agent (Agent.exe)
Configuration Manager (CM.exe)
Information Manager (IM.exe)
Report Manager (RM.exe)
Database Maintenance Utility (DBMT.exe)
Tenant Client Configuration Utility (SeCTenant.exe)

GENERAL PARAMETERS

All applications support a default set of command lines parameters.

/user:<LoginID>

MiCC Enterprise user logon ID.

/password:<Password>

MiCC Enterprise user password.

/webserver:<WebServer[:Port]>

Overrides the Web server and port. Port may be omitted which will default to port 80.

/stdlogin

Forces conventional login using login ID and password bypassing logon using the single sign-on process if configured for the current tenant.

This parameter does not apply to the Tenant Client Configuration Utility. The Tenant Client Configuration Utility uses conventional login by default.

Tip: Holding the SHIFT key down while starting the application will also force conventional login. The key may have to be held for several seconds while the application initializes.

/user and /password must be passed together. If these parameters are specified, conventional logon will take place bypassing the single sign-on process if configured.

For all general and application specific parameters, if the parameter contains spaces, the parameter must be enclosed in quotes.

If no further information is required for the application when specifying command line parameters, the application will start with any additional prompts.

Example:

IM.exe /user:user1 /password:12345

AGENT

The following parameters may be specified for Agent in additional to the general parameters:

/extension:<Extension>

Extension to log into.

/extensionpassword:<Password>

Password, if any, required for the extension.

/oas:<Call Manager Server>

Name of the call manager server if multiple servers are defined. If the Telephony Application Service is used as the call manager type, specifying this parameter will force the specific call manager to be used bypassing the load balancing support used when sites are selected.

/site:<Site>

Name of the site if multiple sites are defined. Only applicable when using Telephony Application Service as the call manager type. If /oas is specified, this parameter is ignored

/softphone

Use soft phone for the extension. May not be combined with /hardphone.

/hardphone

Use hard phone for the extension. May not be combined with /softphone.

/reset

Resets all user preferences.

/resetbars

Resets location and state of the application toolbars.

/logfile:<LogFileName>

Overrides the name of the application log file.

/callto:<Number>

Dials the specified number. An existing instance of Agent must already be running and logged in. The specified number will be passed to that instance to dial.

/deflectto:<Number>

Deflects the active call to the specified number. An existing instance of Agent must already be running and logged in. The specified number will be passed to that instance to perform the deflect.

/tel:<Number>

Dials the specified number. An existing instance of Agent must already be running and logged in. The specified number will be passed to that instance to dial.

/nodcc

No check is performed for the default communications client.

Example:

Agent.exe /user:user1 /password:12345 /extension:5500 /softphone "/oas:Default Server"

CONFIGURATION MANAGER

The following parameters may be specified for Configuration Manager in additional to the general parameters:

/brokername:<ServerName>

Specifies the name of the machine where the Broker service is located.

/brokerport:<Port>

Specifies the port used to connect to the Broker service.

REPORT MANAGER

The following parameters may be specified for Report Manager in additional to the general parameters:

/tenantid:<TenantID>

Specifies the ID of the MiCC Enterprise tenant to use. This can be used to override the configured tenant when connecting to a NOC server.

TENANT CLIENT CONFIGURATION UTILITY

The following parameters may be specified for the Tenant Client Configuration Utility in additional to the general parameters:

/sso

Allows single sign-on to be used for login if configured for the default tenant. Conventional login is used by default.

CONFIGURATION OF PERSONAL CALL ROUTING

Personal calls can be routed to agents through service groups defined as *Voice – Manual Routing* by checking the option *Personal Calls* when defining the service group in Configuration Manager.

When the option *Personal Calls* is set for a *Voice - Manual Routing* service group, preferred agent routing will automatically be set for the service group. When calls arrive to the service group, the called number will be compared to the Personal Directory Number configured for the agent. If the numbers match and the agent is idle, the call is routed to the preferred agent, regardless of the agent's Voice Ready/Not Ready status. If the agent is currently busy with a voice call, the call is added to the Dispatch Call queue for the service group.

It is also possible to designate a call to be sent to a preferred agent from Script Manager when the call is routed to a *Voice – Manual Routing* service group with the *Personal Calls* option set.

If the agent cannot be identified, the call will be added to the Dispatch queue for the service group, and it will be generally available to all agents skilled to serve the service group. The agent can select a call from the Dispatch queue for immediate routing to that agent.

If the agent is not logged on, the call will be diverted to the agent's configured Default Destination for personal calls. If this fails, the call will remain in the Dispatch queue.

If the call is routed to the agent and the agent doesn't answer the personal call within the configured *Ring Time Supervision* value, the call will be added to the Dispatch queue for the service group, and it will be generally available to all agents skilled to serve the service group.

If the agent is busy, the call will wait in the Dispatch queue for the configured *Maximum Wait Time for Personal Call* value; if the agent doesn't retrieve the call within that time period, the call will be generally available to all agents skilled to serve the service group.

Note: Accessing personal calls from the Dispatch queue requires that a Dispatch agent license

is available and assigned to the agent.

PERSONAL CALL ROUTING WITHOUT DISPATCH

If an agent logs on without access to the Dispatch queue, either due to no assigned privilege or license available, personal call routing is still supported. When personal calls arrive, they will be sent directly to the agent, if logged on and idle. The ring timeout value for the call will be the *Maximum Wait Time for Personal Call* value configured for the service group instead of the regular *Ring Time Supervision* value for the system.

If the agent is busy, the call will wait for the agent based on the amount of time configured as the *Maximum Wait Time for Personal Call* value for the service group. Once that time has expired, the call will be deflected to the agent's personal call default destination. It will not be added to the Dispatch queue.

If the call fails to deflect to the agent's personal call default destination, MiCC Enterprise will attempt to deflect the call to the system default destination. If that fails, the call will be added to the Dispatch queue so it can be answered by another agent with Dispatch privilege. Note that once the call is added to the Dispatch queue, it will not be sent directly to the agent. It must be selected from the Dispatch queue.

This allows agents without Dispatch licenses to receive personal calls.

CONFIGURING PERSONAL CALL ROUTING

To configure Personal Call Routing:

1. Configure the agents' defined personal numbers to route to a particular BVD. This is based on the call manager used. It is required that the dialed number is indicated as the agent's personal number. For the MX-ONE, it can be configured as follows:
    - Create a CTI group in the MX-ONE to be used for the personal queue.
    - Define a Personal Directory Number (PDN) in the MX-ONE as a PBX group and divert it to the CTI group created.

      OR
    - Use the DNIS feature to route agents' defined personal numbers to the CTI group created
2. Create a BVD associated with the CTI group created.
3. Using Configuration Manager, define the personal number for the agent in the *Directory Number* field of the *General* tab in the User Properties.
4. Create a service group of type *Voice – Manual Routing*. Check the *Personal Calls* checkbox when defining the service group. This option indicates that the service group will handle personal calls.
5. Create a service access associated with the BVD and set it to route to the personal calls service group.

The call flow will be as follows:

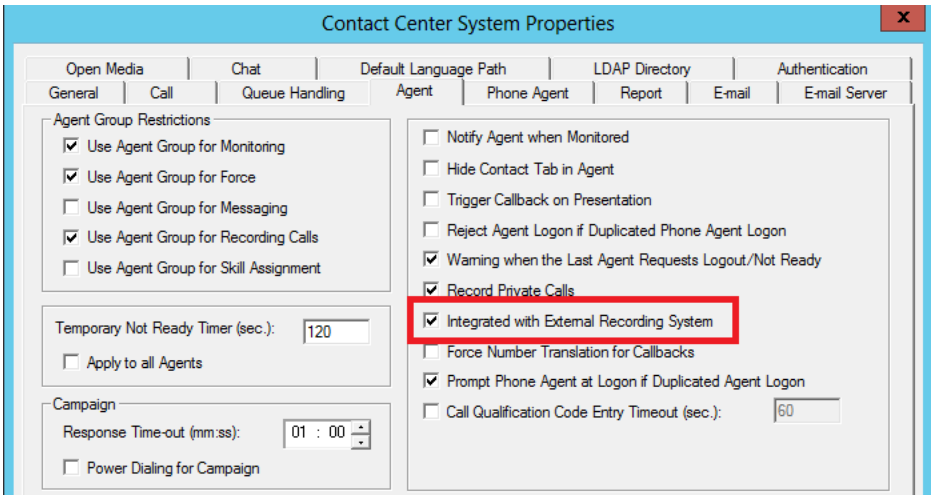
- The caller calls the agent's configured personal directory number
- The call diverts to a CTI group which is associated with the service access that was created to handle the personal calls.
- The service access routes the call to the personal calls service group.
- The service group routes the call directly to the agent with the configured personal directory number

  OR

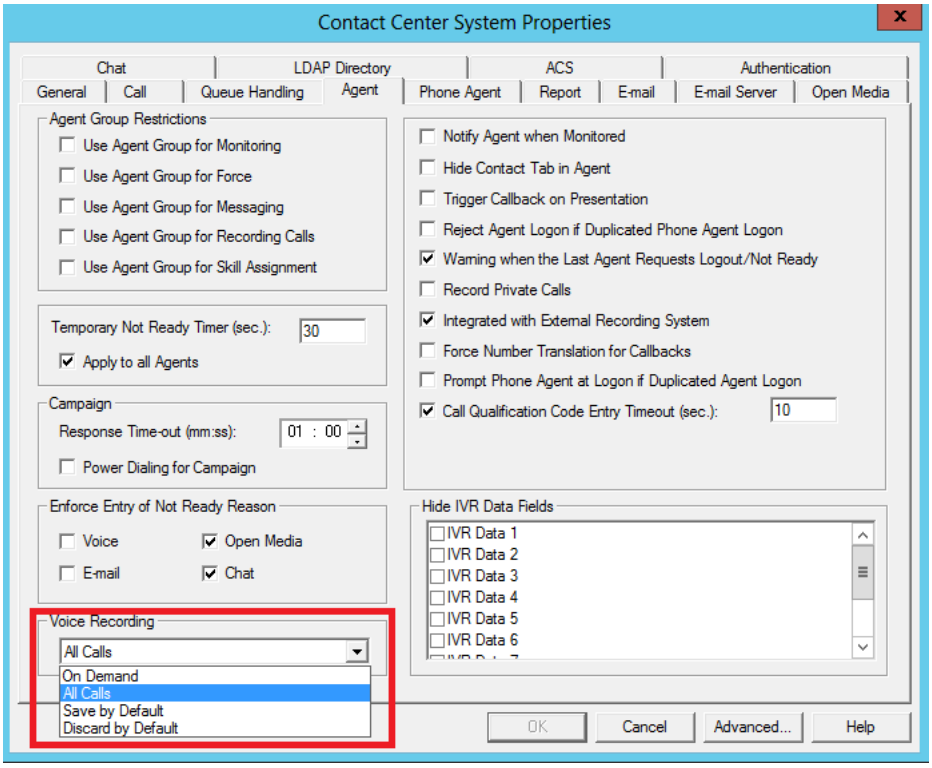
- The service group queues the call in the Dispatch call queue.

INTEGRATING WITH MIVOICE CALL RECORDING

To integrate MiCC Enterprise with MiVoice Call Recording, select the MiCC Enterprise system property "Integrated with External Recording System" from the Agent tab, as shown below:



In addition, it is possible to configure the Voice Recording options for MiCC Enterprise on the Agent tab of the Contact Center System Properties.



The MiVoice Call Recording Recording Action can also be configured through the MiVoice Call Recording Admin tool.

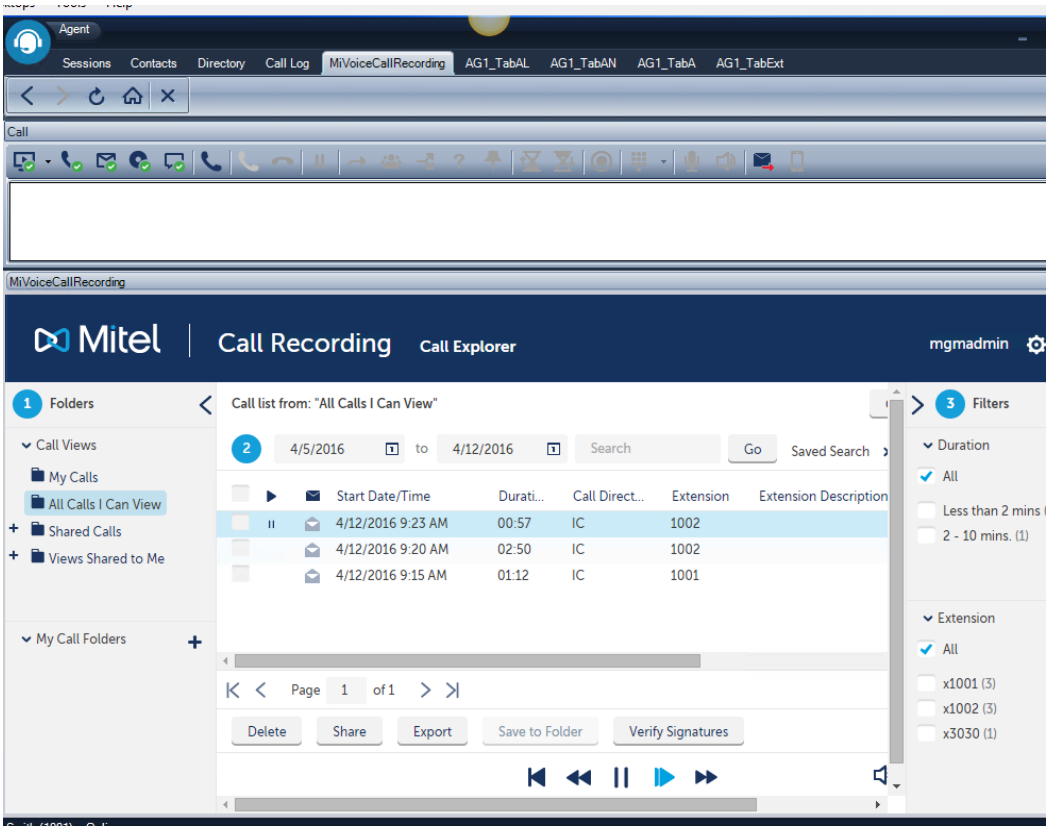
The following configuration options are supported. Note that the MiVoice Call Recording Rule *Do Not Record – No Manual Override* is not applicable when integrating with MiCC Enterprise.

MICC ENTERPRISE	MIVOICE CALL RECORDING RULE	DESCRIPTION
On Demand	Always Record – No Manual Override	Recording will always be performed. The agent will not be able to start/stop recording.
	Do Not Record – Allow Manual Override	Recording will be performed on demand, when the agent selects to start/stop call recording. Recording will not be performed automatically.
	Always Record – Allow Manual Override	Recording will always be performed. The agent will be able to stop recording.

MICC ENTERPRISE	MIVOICE CALL RECORDING RULE	DESCRIPTION
All Calls Save by Default Discard by Default	Always Record – No Manual Override	Recording will always be performed. The agent will not be able to start/stop recording.
	Do Not Record – Allow Manual Override	Recording will always be performed. The agent will be able to start/stop recording if the agent has *Record Calls* privilege.
	Always Record – Allow Manual Override	Recording will always be performed. The agent will be able to start/stop recording if the agent has *Record Calls* privilege.

In addition, it is possible to configure the Voice Recording options for MiCC Enterprise on the Agent tab of the Contact Center System Properties.

Note that recorded files will not be visible in the MiCC Enterprise Agent application. Recorded files can be viewed through MiVoice Call Recording. The MiVoice Call Recording Navigator web client can be added as a browser tab in the Agent application by configuring http://<MiVoice Call Recording Server Name>/navigator as an Agent tab. This will display the Navigator web client in Agent as shown below.



MICC ENTERPRISE SUPPORT DLLS

MiVCR integrates with MiCC Enterprise using the following three DLLs. These files will be installed by MiVCR; however, ensure that the version of the files matches the version installed with MiCC Enterprise. If the versions are not the same, copy the files from the MiCC Enterprise server to the MIVCR server:

DLL Name	Location on MiVCR Server	Location on MiCC-E Server
CCASComClient.dll	<InstallDir>\CTS.NET\Solidus DLLs	<InstallDir>\Common Files\EricssonShare\NextCCShare
SocketManager.dll	<InstallDir>\CTS.NET\Solidus DLLs	<InstallDir>\Common Files\EricssonShare
SeCTraceLog.dll	<InstallDir>\CTS.NET\Solidus DLLs	<InstallDir>\Common Files\EricssonShare

USING CALL RECORDING IN AGENT

When MiCC Enterprise is integrated with MiVCR for call recording, the Record Calls button on the Sessions tab of the Agent application will be displayed if the agent has Record Calls or Record Others privilege.

Pressing the Record Calls button displays the Record Calls dialog, which allows the agent to start, stop and pause recording on the MiVCR system.

USING CUSTOMER AUTHENTICATION

OVERVIEW

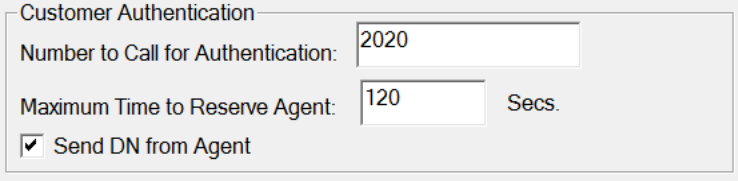
The Customer Authentication feature enables a MiCC Enterprise Agent to transfer a caller to a designated service access where a script authenticates the caller and then returns the caller to the same agent to continue the transaction with the agent.

While the caller is at the service access, the agent remains in “Call Waiting” state, waiting for the caller to return. When the call is returned, it may optionally include an associated script variable to be returned to the agent with the status of the authentication.

CONFIGURATION

To configure Customer Authentication, execute the following steps:

1. In SeCCfg, under the Agent Service tab, enter the following parameters as shown:



- Number to Call for Authentication
  Set to the BVD value used for the Authentication service access. Note that the value entered must exactly match the number provided in the *New Destination* field for the Diverted event when the call is diverted to the Authentication service access.

- Maximum Time to Reserve Agent
  Set to the maximum number of seconds that an agent will wait for a customer call to return from the Authentication service access. If the call does not return within the configured time period, the agent will be cleared and available for another service group call.

- Send DN from Agent
  If checked, the agent's extension will be sent in the private data when the call is diverted to the Authentication service access. The script can use this data to divert the call back to the agent after authentication is complete.

2. Create a Script Manager service access to handle Authentication calls and associate it with a script. The script must contain the following elements:

   - An "Assign" block should be used to store the agent's extension from the system variable @@MediaLib.PrivateData.

   - A "Deflect with Data" block should be used to return the call to the agent's extension. The "Destination Number" field can be set to the variable containing the saved agent's extension. The "Associated Data" field can be set to a variable containing the result of the authentication.

   - There should be an appropriate failure branch from the "Deflect with Data" block to handle cases where the agent is no longer available or the agent reserve timer expires.

AGENT OPERATION

When the agent is ready to send the caller to the Authentication service access, the Divert dialog (accessible via the F8 key in Agent) should be used to divert the call to the configured number to call for authentication.

The call will be diverted to the Authentication service access. The agent will continue to display the call, with the state *Call Waiting*. The agent will be considered as busy to the MiCC Enterprise system and no further voice calls will be allocated. If the call is returned to the agent before the

configured agent reserve timer expires, it will be displayed with the call state *Callback*. The agent can then answer the call and continue assisting the customer. The data associated with the call which shows the result of the authentication will be displayed in the Session information.

If the call is not returned to the agent before the agent reserve timer expires, the call will be cleared, and the agent will enter Clerical state, if configured for the service group. The agent will be available to receive other service group calls.

When the call is sent to the Authentication service access, Call Detail events indicating *Parked* and *Call Deflected to other Destination* will be generated.

USING CUSTOMER AUTHENTICATION WITH PHONE AGENTS

The Customer Authentication feature can also be used with phone agents (i.e. agents logged on direction to MiCC Enterprise and not using the Agent application). Configure the Customer Authentication parameters as indicated in the Configuration section above. *DeflectPhoneAgentCall* can be used from the CCAS Open Interface via an integration application to divert the call from the phone agent to the configured Customer Authentication number. While the call is diverted, the phone agent will remain in Talking state.

If the phone agent is not using an application that can indicate whether the customer was successfully authenticated, the script can disconnect the caller if authentication fails. If it is possible to indicate success/failure to the phone agent, the CCAS Open Interface API *SetAgentIVRData* can be used to indicate whether the customer successfully authenticated. The script can return the call to the phone agent by diverting it back to the agent's extension, which is provided in the private data for the call.

If the phone agent fails to answer the return call, or the configured timeout occurs before the call is answered, the call will be treated as a disconnected call and the phone agent will be available to receive the next service group call.

E-MAIL, CHAT AND SMS RESPONSE FILES

Response files may be configured for e-mail, chat and SMS service groups allowing agents to insert predefined responses. Response files are standard XML based files. All standard XML encoding rules must be observed. For example, if < is to be used in the response text, it must be entered as < which is the XML escape sequence for the < symbol.

FORMAT

```
<?xml version="1.0" encoding="utf-8"?>
<ResponseFile>
  <Response Name="Simple Response with Description" Description="This is a plain
text response with a description">This is the response</Response>
  <Response Name="Simple Response without Description">This is the
response</Response>
```

```
  <Response Name="Response with a replacement" Description="This is a plain text
response with replaceable identifier">This is the agent name:
$Agent.Name$</Response>
  <Response Name="Simple Response without Description">This is the
response</Response>
  <Response Name="Multiline Response" Description="This is a multiline response
with encoded CRLFs">Response Line 1&#x0d;&#x0a;Response Line 2</Response>
  <Response Name="Image Response" Src="Images/image1.jpg" Description="This is an
image response using a relative URI" />
  <Response Name="HTML Response" Src="HtmlFiles/HtmlResponse.htm"
Description="This is a HTML response using a relative URI" />
  <Response Name="Image Response with Absolute URI"
Src="www.mitel.com/Images/logo.jpg" Description="This is an image response using
an absolute URI" />
  <Response Name="External Text Response" Src="TextFiles/TextResponse.txt"
Description="This is a text response using an external source" />
  <Group Name="Group 1">
    <Response Name="Response in Group 1">This is the response</Response>
    <Group Name="Group 2 Inside Group 1">
      <Response Name="Response in Group 2">This is the response</Response>
    </Group>
  </Group>
  <Group Name="Group 3" Expanded="true">
    <Response Name="Response in Group 3" Description="This response is in a group
that is expanded by default">This is the response</Response>
  </Group>
</ResponseFile>
```

Nesting level of the groups is unlimited.

Nodes

NODE	MEMBER	TYPE	DESCRIPTION	REQUIRED
<ResponseFile>		Node	Root XML node. May contain any combination of <Group> and <Response> child nodes.	Yes
<Group>		Node	Group node. May contain any combination of <Group> and <Response> child nodes. Groups that do not contain any sub items due to filtering or unsupported response types will not be displayed.	No
	Name	Attribute	Specifies the name of the group.	Yes
	Expanded	Attribute	Specifies if the group should be expanded by default in the response tree. Value values are “true” and “false”. Default = “false”.	No
<Response>		Node	Response node.	No

	Name	Attribute	Specifies the name of the response.	Yes
	Description	Attribute	Specifies the description of the response that will be shown in tooltips. If this is omitted, the inner text will be used for the description on plain text responses.	No
	Src	Attribute	Specifies a URI to an external source file that is to be used as the response. External files may be images, HTML files or plain text files. URI may specify an absolute or relative path. Relative paths are relative to the location of the XML response file. External source files must be accessible by the agent clients. Any references inside HTML files must also be accessible by the clients. **Supported Files:** Any file extension not listed below will be treated as a plain text file. HTML (*.htm, *.html) Plain Text (*.txt, *.text) **Images:** Bitmap (*.bmp, *.dib) JPEG (*.jpg, *.jpeg GIF (*.gif) (non-animated) TIFF (*.tif, *.tiff) PNG (*.png) HTML and images are supported only for e-mail.	Src or Inner Text must be specified
		Inner Text	Specifies the response text. Unused if Src is specified.	Src or Inner Text must be specified

REPLACEABLE IDENTIFIERS IN RESPONSE FILES AND KB RESPONSES

Plain text and HTML responses whether contained in the response inner text or external source files or in knowledge base responses may contain identifiers that will be replaced when the response is inserted. It is important to ensure that replaceable identifiers are entered as a continuous string in the responses. HTML editors such as Microsoft Word may split the text while inserting HTML format tags. This will prevent the identifiers from being replaced. This may occur if text is identified as a misspelled word. The underlining used in Microsoft Word to indicate the misspelled word will be stored in the HTML file as formatting information. Always ensure that replaceable identifiers are ignored for spell checking.

IDENTIFIER	REPLACEMENT	APPLIES TO
$CallID$	Unique Identifier	Incoming E-mail/SMS, Chat
$Subject$	E-mail subject	Incoming E-mail
$From$	Sender name and address. For example, John Smith (john.smith@company.com)	Incoming E-mail/SMS
$From.Name$	Sender name. If the name is not available, the sender address will be used. For SMS, the sender address will be used.	Incoming E-mail/SMS
$From.Address$	Sender address	Incoming E-mail/SMS
$Date$	Current date formatted using the short date format of the current locale	All
$Time$	Current time formatted using the short time format of the current locale	All
$Received$	Date and time the e-mail, SMS or chat was received formatted using the short date and short time formats of the current locale	Incoming E-mail/SMS, Chat
$Received.Date$	Date the e-mail, SMS or chat was received formatted using the short date format of the current locale	Incoming E-mail/SMS, Chat
$Received.Time$	Time the e-mail, SMS or chat was received formatted using the short time format of the current locale	Incoming E-mail/SMS, Chat
$ServiceGroup$	Service group name	All
$ServiceGroup.Name$	Service group name	All
$ServiceGroup.Email$	E-mail address configured for service group	Chat
$Agent$	Agent name	All
$Agent.Name$	Agent name	All

$Agent.FirstName$	Agent first name, if defined	All
$Agent.LastName$	Agent last name, if defined	All
$Agent.ChatName$	Agent Chat Display Name, if defined. Otherwise, set to agent name.	All
$Customer$	Customer name	Chat
$Customer.Name$	Customer name	Chat
$Customer.Email$	Customer e-mail address	Chat

CUSTOMER CONFIGURATIONS FOR WEB INSTALLATION AND CLIENT UPDATES

During installation of the MiCC Enterprise server, a repository is setup that holds the configuration information used for Web installations and update downloads. The location of the repository is:

<InstallDir>\WebDeployment

The repository contains a subfolder for each tenant or "Customer". Configurations need not be exclusive for each tenant. A single tenant can contain multiple configurations. The name of the folder determines the URL used to access the client download page as well as the page for ClientSetup.ini file generation. The format of the URL for the client download page is:

http://SECWEBSERVER/MiCCEInstallation/install/[CustomerID]

Where: **SECWEBSERVER** is the computer running the MiCC Enterprise Web Services and **CustomerID** is the name of the subfolder created in the repository.

If the folder is called "CustomerOne", the URL used to access the client download page would be:

http://SECWEBSERVER/MiCCEInstallation/install/CustomerOne

which would be using the configuration repository folder:

<InstallDir>\WebDeployment\CustomerOne

The default server installation already contains configuration for the default tenant. **CustomerID** may be omitted for the default tenant and the URL may simply be:

http://SECWEBSERVER/MiCCEInstallation/install

Each configuration folder in the repository must contain the configuration file Setup.config which specifies the files to download and applications to execute during installation.

SETUP.CONFIG FORMAT

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <execute command="Solidus\ClientSetup.exe" args="/Q /F /C:ClientSetup.ini"
hidden="true" />
  <customerName>Default Tenant</customerName>
  <tenantID>-1</tenantID>
  <files>
    <file include="$(WebServerUri)/MiCCEInstallation/NextCCClient/*.*"
subDirs="true" outputFolder="Solidus" />
```

```
    <file include="$(WebServerUri)/MiCCEInstallation/ScriptManager/*.*"
subDirs="true" outputFolder="ScriptManager" />
  </files>
</configuration>
```

Nodes

NODE	MEMBER	TYPE	DESCRIPTION	REQUIRED	DEFAULT
<configuration>		Node	Root XML node.	Yes	
<execute>		Element	Specifies the command to execute to start the installation.	No	
	command	Attribute	Command to execute. If this contains a relative path, it is assumed to be relative to the base download location.	Yes	
	args	Attribute	Arguments to pass to the command.	No	Empty string
	hidden	Attribute	“true” or “false”. If true, application will be launched hidden.	No	false
<customerName>		Element	Customer name that is displayed in the download page.	No	Empty string in which case CustomerID is used.
<tenantID>		Element	ID of the tenant associated to this customer.	Yes	
<files>		Node	Root node containing a list of <file> nodes. At least 1 <file> node must exist.	Yes	
<file>		Element	Specifies a list of files that should be downloaded during the installation.		
	include	Attribute	Specifies the list of files to include. May contain wildcards * and ?. Files specification must be network accessible by the client. Supported protocols are: http://	Yes	

			https:// ftp:// file:// \\UNCPATH Files need not be located on the MiCC Enterprise server. Files may be specified that are on the client's local network. One example of this would be downloading a ClientSetup.ini that is on the client's local network.		
	exclude	Attribute	Specifies a list of file specs to exclude. Only applies if *include* contains wildcards. File specs may contain wildcards * and ?. Multiple file specs may be listed separated by a semicolon (;).	No	Empty string
	subDirs	Attribute	"true" or "false". If true and *include* contains wildcards, subdirectories are searched.	No	false
	outputFolder	Attribute	Specifies the subfolder to place the files.	Yes	

In the default Setup.config file, all files from the NextCCClient and ScriptManager shares on the computer running the Broker Service are to be downloaded and placed in the "Solidus" and "ScriptManager" folders.

After all files have been downloaded, ClientSetup.exe is executed.

Replaceable Identifiers

The Setup.config files may contain identifiers that will be replaced when they are loaded. Path names will always contain the ending backslash (\).

IDENTIFIER	REPLACEMENT
$(InstallDir)	MiCC Enterprise installation folder.
$(SMInstallDir)	Script Manager installation folder.
$(SMSGInstallDir)	Mitel SMS Gateway installation folder.
$(TASInstallDir)	TAS installation folder.
$(EricssonShareDir)	Common shared folder. Typically, C:\Program Files (x86)\Common

	Files\EricssonShare\.
$(NextCCShareDir)	MiCC Enterprise common shared folder. Typically, C:\Program Files (x86)\Common Files\EricssonShare\NextCCShare\.
$(BrokerServer)	Broker server computer.
$(WebServer)	MiCC Enterprise Web server.
$(WebServerPort)	MiCC Enterprise Web server port.
$(WebServerUri)	The full URI to the MiCC Enterprise Web server. For example, http://WEBSERVER:80
$(WindowsDir)	Windows folder.
$(System32Dir)	System folder.

EXAMPLE

The following examples will demonstrate setting up a repository folder for a new customer. It is assumed that a tenant has already been created for use by the customer and the tenant ID is known. The tenant ID can be viewed in the *Defined Tenants* page of the MiCC Enterprise Setup Utility.

The following properties will be assumed:

Customer ID: acmecorp
Customer Name: ACME Corporation
Tenant ID: 1
Tenant Name: ACME
MiCC Enterprise Server: MICCESERVER
MiCC Enterprise Web Server: MICCSERVER

1. Create the customer repository folder.

    a. Locate the <InstallDir>\WebDeployment folder.

    b. Create a new subfolder under WebDeployment called acmecorp. This name must match the customer ID. The name is not case sensitive.

    c. Copy the default configuration file, Setup.config, from the WebDeployment\Default folder to WebDeployment\acmecorp.

    d. Open the WebDeployment\acmecorp\Setup.config file in the text editor of your choice.

    e. Change the <customerName> entry to ACME Corporation.

    f. Change the <tenantID> entry to 1.

    g. Save the file.

2. Generate the ClientSetup.ini file.

    The ClientSetup.ini file may be generated by the MiCC Enterprise Host Administrator or it may be generated by the customer. Installing the MiCC Enterprise client requires local administrator rights on the computer. If the users installing the client do not have the rights, it may be useful to run the installation under a specified user account that does have the rights. If the user does not have the rights when starting the installation and a user account has not been specified, the user will be prompted for an account with administrator rights. A user account and password may be specified in the ClientSetup.ini file. The customer may not want this information known outside of their company so the ClientSetup.ini file should be generated and hosted by the customer.

a. Access the Client Setup INI File Generation page using the following URL:

   http://MICCSERVER/MiCCEInstallation/ClientSetup/acmecorp

b. Enter the user account, password and default features to use during the client installation.

c. Click the Generate button and save the file. Note the location.

3. Customer Generated and Hosted ClientSetup.ini.

   The following applies if the ClientSetup.ini file was generated by the customer.

   The generated ClientSetup.ini file must be placed in a location that will be accessible by all users installing the client. For this example, let's assume that the customer has already setup a shared folder called:

   \\ACMESERVER\SharedFiles

   a. Copy the generated file to the \\ACMESERVER\SharedFiles folder.

   b. Notify the MiCC Enterprise Administrator of the full path and filename of how users will access the file. In this case:

      \\ACMESERVER\SharedFiles\ClientSetup.ini

4. MiCC Enterprise Administrator Host Generated ClientSetup.ini.

   The following applies if the ClientSetup.ini file was generated by the MiCC Enterprise Host Administrator.

   a. Copy the generated file to the WebDeployment\acmecorp folder.

5. Set the location of ClientSetup.ini.

   a. Open the WebDeployment\acmecorp\Setup.config file in the text editor of your choice.

   b. Add a new <file> entry to the <files> node to download the ClientSetup.ini file.

      For a customer generated ClientSetup.ini file, add the following line:

      ```
      <file include="\\ACMESERVER\SharedFiles\ClientSetup.ini"
      outputFolder="Solidus" />
      ```

   c. For an Administrator generated CientSetup.ini file, add the following line:

      ```
      <file
      ```

```
include="$(WebServerUri)/MiCCEInstallation/WebDeployment/acmecorp/ClientSet
up.ini" outputFolder="Solidus" />
```

d. The new <file> entry must be after the existing <file> entries.

e. Save the file.

CUSTOM CUSTOMER VIEWS

The client download page may be customized for each customer. The default page is:

<InstallDir>\Services\Web\WebDeploy\Views\Customers\Index.cshtml

The default page may be used by all customers in which case it simply displays the name of the customer. To use a custom page for the customer, copy the default page to the customer's repository folder. For example, copy:

<InstallDir>\Services\Web\WebDeploy\Views\Customers\Index.cshtml

To:

<InstallDir>\WebDeployment\acmecorp\Index.cshtml

Modify the customer's Index.cshtml file as required. Modification of the file requires knowledge of Microsoft MVC/Razor technology.

HOTFIX/INSTALLATION UPDATES

Beginning in release 9.2, an updater service is installed on each MiCC-E client. This service monitors the MiCC Enterprise server for new installation packages, including major/minor releases, service packs and hotfix updates. If enabled, the service will check for updates once per day and download them to the local computer so they are ready to be installed by the user. An update check is also performed each time a user starts one of the MiCC Enterprise applications. If updates are available, the user will be prompted to install the update. If there are any pending updates, they are required to be installed. Refusing the update will terminate the application.

Settings such as whether to perform background update checks or whether to check for updates on application startup can be changed through the MiCC Enterprise Registry Configuration utility.

The updater service is included with every MiCC Enterprise installation; however, it will only perform update operations if no other MiCC Enterprise service is installed on the local computer, i.e. the local computer only contains MiCC Enterprise client applications.

NEW INSTALLATION PACKAGES

After a new installation package has been applied to the MiCC Enterprise server, the client will be prompted to upgrade its installation the next time the user starts a MiCC Enterprise application. The new installation package may have already been downloaded during the background update check. Follow the prompts to install the new package. The package will be installed according to the ClientSetup.ini file setup for the customer in the repository. The installer will be launched under the user account configured if any.

HOTFIX UPDATES

Occasionally, hotfix updates may need to be applied to the MiCC Enterprise server(s) and clients. Hotfixes may be supplied which will include one or more files. A hotfix will always contain at least a configuration file, *.config and may also contain additional files necessary for the update. HotFixes are manually applied to the MiCC Enterprise server(s). The updater service installed on each client will download and install and hotfixes available on the server.

1. Install the hotfix on the server(s).

   A hotfix is applied to the server using the MiCCEHotFixInstaller.exe located in the following folder:

   <InstallDir>\NextCC Setup

   A new MiCCEHotFixInstaller.exe may also be supplied with the hotfix in which case that file should be used instead.

   On every MiCC Enterprise server, perform the following actions. If the MiCC Enterprise services have been split onto multiple servers, the hotfix must be applied to each server.

   a. Launch the MiCCEHotFixInstaller.exe utility.

   b. Click the Browse button (…) to select the *.config file included with the hotfix.

   c. Click the Apply button to apply the hotfix to the server. During the application of the hotfix, it may be necessary for the installer to stop one or more MiCC Enterprise services which could affect system operation. If this is necessary, a warning message will be displayed first.

   If the hotfix is being applied to the main MiCC Enterprise server where the Broker service is installed, the hotfix will also be added to a repository where clients will retrieve the hotfix.

2. Update the clients.

   The client updates are mostly performed automatically. The updater service will perform background update checks against the MiCC Enterprise server. The next time a client application is started, the user will be prompted to install the update. Installation of the update is done by the updater service and does not require any additional user input.

HotFixes may also be manually applied to the clients using the MiCCEHotFixInstaller.exe utility. To manually apply the update, perform the same steps as described for the MiCC Enterprise servers.

MICCEHOTFIXINSTALLER

COMMAND LINE PARAMETERS

MiCCEHotFixInstaller [/i:<Configuration Filename>] [/s]

Configuration Filename: Specifies the name of the configuration file to load.

/s: Installs the HF silently and exits. The exit code for the application will be 0 if successful or 1 if an error occurs. If the HF is already installed, it will be reinstalled.

LOG FILE

Log output will be contained in the following file:

%USERPROFILE%\AppData\Local\Mitel\MiCC Enterprise\MiCCEHotFixInstaller\MiCCEHotFixInstaller.log"

PASSWORD MANAGEMENT

Password management may be enabled to implement security measures for user accounts. Properties may be set at the tenant level to require password changes and to lock user accounts after a specified number of failed logon attempts.

Accounts that are locked due to too many failed logon attempts may be unlocked by accessing the user properties in Configuration Manager or Web Manager. Accounts may also be manually locked preventing user logon. When an account is locked due to too many failed logon attempts, the TCP/IP address of the computer which caused the lockout will be logged to the log file for the Logon Web Service.

In a hosted environment, the host administrator account may be locked out by any computer in any tenant on the system. There is no way to prevent this as it would defeat the security measures. In the event that the host administrator account is locked and there is no other account which can access Configuration Manager or Web Manager, the host administrator account can be unlocked through the MiCC Enterprise Installation. Run the MiCC Enterprise Installation package from the installation media or Control Panel. An option will be given to change the Administrator password and lock status. This must be run on the MiCC Enterprise server.

SERVICE SECURITY

WSDL SUPPRESSION

ASP.NET Web Services running under IIS and most WCF services emit WSDL (Web Service Definition Language) if specific URLs are used to access the service. For example, the following URLs may be used to emit the WSDL for the MiCC Enterprise Logon Web Service and MiCC Agent Service:

http://localhost/seclogonws/seclogonws.asmx?wsdl
http://localhost:12613/RequestService?wsdl

The WSDL also allows references to the services to be created using development tools. For security reasons, this ability to retrieve the WSDL may be suppressed. The procedure is different for Web Services running under IIS and the WCF services.

WEB SERVICES RUNNING UNDER IIS

For each MiCC Enterprise Web Service the web.config file must be modified. All services are located under <InstallDir>\Services\Web. Locate the web.config file for each service and add the following configuration inside the <system.web> node:

<webServices>
<protocols>
<remove name=”Documentation”/>
</protocols>
</webServices>

The change may also be done at a system level rather than in each individual service. Doing so will disable the WSDL for the entire computer including all services. Not just MiCC Enterprise services. To disable the WSDL at the system level, enter the same configuration information described above into the following file:

<WindowDir>\Microsoft.NET\Framework\v4.0.30319\Config\machine.config

The Web Services or IIS do not need to be restarted after changing the configuration files.

WCF SERVICES

The WSDL for the WCF services may be disabled by setting the following registry value on each machine where MiCC Enterprise services are installed:

Key: HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Mitel\SeC\Common\Parameters
Value Name (REG_DWORD): DisableWSDL
Value: 1

The MiCC Enterprise services must be restarted for the setting to take effect.

CALLER ID FOR OUTGOING CALLS

For outgoing calls, custom caller IDs may be defined and used. Custom caller IDs are defined per tenant with a name and caller ID string. For private calls, one of the custom caller IDs may be selected as the default and whether the agent is allowed to select a different caller ID. For outgoing service group calls, selection and change permission is configured for each service group. If a custom caller ID is used, that caller ID will be sent to the remote party during the call.

For callback, campaign and private calls made through the Manual Dial form, the caller ID may be selected by the agent if allowed. For private calls made without using the Manual Dial form such as dialing from the directory, the default caller ID selected for the tenant will be used.

Configuration of the caller ID strings is dependent on the type of call manager being used.

OPEN APPLICATION SERVER

For Open Application Server, the defined caller ID string will be prepended to the called number when making the outgoing call. The call manager must be configured to perform number translation on the prepended route access code. Due to the routing translation done in the call manager, number translation in the Agent will be disabled when making a call using a custom caller ID.

Different route access codes can be set on the outgoing route via the MX-ONE MML command:

'*RODDI:ROU=n,DEST=nn,…*'

Then, the 'number_conversion_initiate' command can be used to remove the caller's (agent's) real extension number and replace it with something else.

An example: let's say the MX-ONE is using a 5 digit long extension number for the agents and they all start with digit 6. Then, for calls to numbers prefixed with 811 we want to replace the agents' extension numbers with number 33355. This command can be used:

number_conversion_initiate -conversiontype 1 -numbertype 10 -entry 6 –truncate 5 -pre 33355 -targetdest 811.

Note: The *numbertype* parameter indicates the type of number for the trunk, where 10 indicates public number and 11 indicates private tie-line. Set the *numbertype* parameter as appropriate for your system.

TELEPHONY APPLICATION SERVICE

For Telephony Application Service, the caller ID string defined will be sent in its entirety to the remote party. No additional configuration is required in the call manager.

CALL MANAGER LOAD BALANCING

When using the Telephony Application Service call manager type, sites are selected rather than specific call managers during agent logon. Multiple call managers may be defined per site. Each time the agent connects to the call manager, the best call manager/call control service will be used to distribute the load evenly. When failures occur within the call manager or call control service, attempts are made to avoid the failing call manager or call control service. A specific call manager may still be used by specifying the /oas command line parameter when starting the Agent application. In this case, only that call manager will be used.

A typical configuration of multiple call managers would be to install the MiCC Enterprise Call Control service on the computer along with the call manager. The configuration of the call manager in Configuration Manager should point to the call control service on the same machine. A specific call control service does not need to be specified for the call manager in which it will use any call control service registered in the system, however, no attempt will be made to avoid the call control service when failures occur.

LOAD BALANCING FOR PHONE AGENTS

When a phone agent logs in using the Mobile Agent application, or via the CCAS Open Interface, the call manager ID is provided in the logon request. If multiple TAS systems are installed on the same site as the selected call manager ID, the MiCC-Enterprise Router Service will find the least used system and start the monitor for the phone agent on that TAS system. Phone agents automatically logged on at startup via the Phone Agent Automatic Logon feature will follow the same load sharing strategy.

When a phone agent logs in by calling the configured Logon BVD, the system called into will always be used to start the monitor for the phone agent.

If the Call Control Service or TAS fail while a phone agent is logged on, the MiCC-Enterprise Router Service will check for another active system on the same site. If one is found, the phone agent monitor will attempt to be started on that system. If one cannot be found, a timer will be set to retry starting the monitor after a default time of 30 seconds.

Note that this feature is only applicable for TAS based systems, and not OAS systems.

AGENT DATA ACCESS

Many data items in Agent are affected by the permission settings on the data object in Configuration Manager, while other data items are based on skill access. The table below indicates which data objects are visible in the Agent application.

DATA	VISIBILITY TO AGENT
Service Groups in Real-Time Window	Displays all service groups for which the agent has read, read/write, or write permission, as well as all service groups that the agent is skilled to serve. Note that service groups which the agent is skilled to serve may be displayed even if

	the agent doesn't have permission for the service group.
Service Groups in Dispatch Window	By default, displays all service groups that are of type *Manual Routing*, or which are defined with the option *Display in Dispatch* and that the agent is skilled to serve. If the SeCCfg option *Allow Agents to View all Dispatch Groups* is set, the window will display all service groups of type *Manual Routing*, or which are defined with the option *Display in Dispatch*. In this case, if the agent is not skilled to serve the service group, the calls will be read-only. The agent will not be able to answer calls from the service group or move calls to another service group.
Agent Skill Assignment and Skill Matching – Agent List	If the agent has *Change Skills for Other Agents* privilege, all agents are displayed in the list of available agents to be configured. If the system option *Use Agent Group for Skill Assignment* is selected, only agents that are members of agent groups for which the agent has read, read/write, or write permission will be displayed.
Agent Skill Assignment and Skill Matching – Skill List	Displays skills assigned to service groups for which the agent has read, read/write, or write permission. If a skill is not assigned to a service group, it will not be visible as an available skill. If a skill is assigned to a service group for which the agent does not have permission, it will not be visible as an available skill.
Agent Skill Assignment and Skill Matching – Skill Templates	Displays templates that contain at least one skill that the agent is allowed to view (i.e. the skill must be assigned to a service group for which the agent has permission).
Contacts	By default, displays all logged on agents. If the SecCfg option *Use Agent Group for Dial* is set, only agents in the same agent group are displayed. If the SeCCfg option *Suppress Agents outside Department* is set, only agents in the same department (as configured with the DepartmentConfig.exe utility application) are displayed. If the system option *Use Agent Group for Messaging* is set, agents will not be able to send messages to agents in a different agent group.
Force dialog	By default, displays all logged on agents. If the system option *Use Agent Group for Force* is set, it only displays agents in the same agent group as the requesting agent. If the SeCCfg option *Suppress Agents outside Department* is set, only agents in the same department (as configured with the DepartmentConfig.exe utility application) are displayed.
Monitor dialog	By default, displays all logged on agents. If the system option *Use Agent Group for Monitor* is set, it only displays agents in the same agent group as the requesting agent.
Assist, E-mail Assist, Chat Assist dialogs	Displays all agents with privilege to Provide Assistance
Divert dialog – Service Groups	Displays all open service groups for the media type. A service group is considered open if at least one agent is logged on, or at least one agent is logged on and in Ready status, depending on the configuration for the service group.

PHONE AGENT AUTOMATIC LOGON

Phone Agents can be configured to automatically logon to the MiCC-E system when the MiCC-E Router Service restarts. There are two configuration options available:

- Tenant System Property – Automatically Restore Logon Status After Restart

  This option is available on the Phone Agent tab of the Contact Center System Properties in Configuration Manager. It allows the user to set automatic logon for all phone agents at once. If the option is set, any phone agents previously logged on when the MiCC-E system (or Router Service) restarts will automatically be logged on again to the same extension. The voice ready status will also be restored.

- User Property – Phone Agent Logon

  This option is available on the General tab of the User Properties in Configuration Manager, as well as in Web Manager. By default, all agents will be set to use the system setting for phone agent automatic logon. To override the system setting, set the user option to “Restore Logon State” which will always restore the logon state of the phone agent, regardless of the system setting, or set to “Don’t Restore Logon State” to never restore the logon state of the phone agent after a system restart.

  Since it can be time consuming to logon many phone agents when the MiCC-E system restarts, it is possible to prioritize certain agents to be logged on first. To do this, set the user option to “Restore Logon State with Priority”. Agents marked with priority will be logged on before any other phone agents to set to logon automatically.

Either of these configuration options can be set dynamically and will take effect immediately.

DATABASE HANDLING

In order for phone agents to be automatically logged on after a restart, the phone agent must have previously logged on to the MiCC-E system. This is necessary in order to associate an extension and call manager with the phone agent. Phone agent logon information is stored in the MiCC-E database, in the phone_agent_logon table.

If the system or user setting is configured to restore the phone agent logon state after a restart, agent information will be added to this table when phone agents logon or change voice ready status. When phone agents logoff, they are removed from the table. When a restart occurs, any agents stored in the table will be automatically logged on to the MiCC-E system.

There are multiple methods of logging on phone agents, including: calling the logon device, using Script Manager, using Mobile Agent, and using the Agent Service Open Interface. Any of these methods will automatically update the database table. For customers that wish to populate this table without actually logging on phone agents, the following fields must be updated.

FIELD	VISIBILITY TO AGENT
agent_id	Record ID of the agent from the cc_user table
tenant_id	Tenant ID of the agent
call_manager_id	Record ID of the call manager to which the phone agent will be logged on. This corresponds to the id field from the site_server_param table.
phone_num	Extension number to which the phone agent will be logged on.
ready_status	Indicates whether the phone agent will be set to ready (ready_status = 1) or remain not ready (ready_status = 0) for voice calls after logging on.
priority_logon	Indicates that the agent is set for priority logon, so it should be logged on before any agents that do not have this value set.
time_stamp	Time stamp of last update.

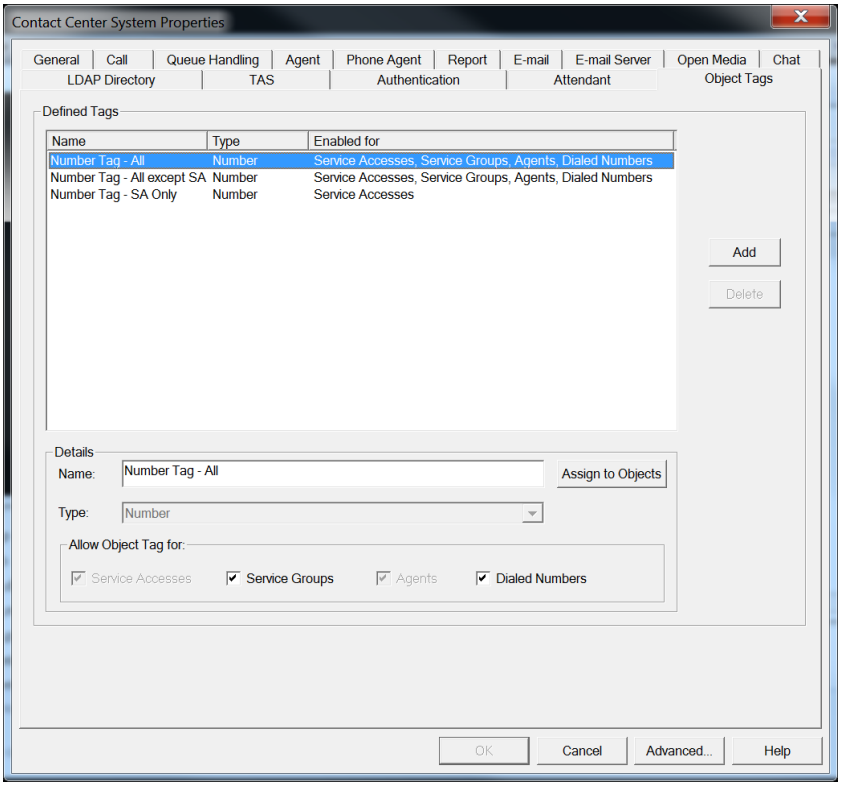
The contents of the table will be read and processed the next time the MiCC-E Router Service is restarted.

OBJECT TAGS

Object tags provide a method of grouping Agent, Service Group, Service Access and Dialed Number objects in the MiCC Enterprise system. An object tag can be assigned to any individual Agent, Service Group, Service Access or Dialed Number.

Object tags can be any of the following types: Number, Text, Service Access, Service Group, Scheduler, Agent or Play message.

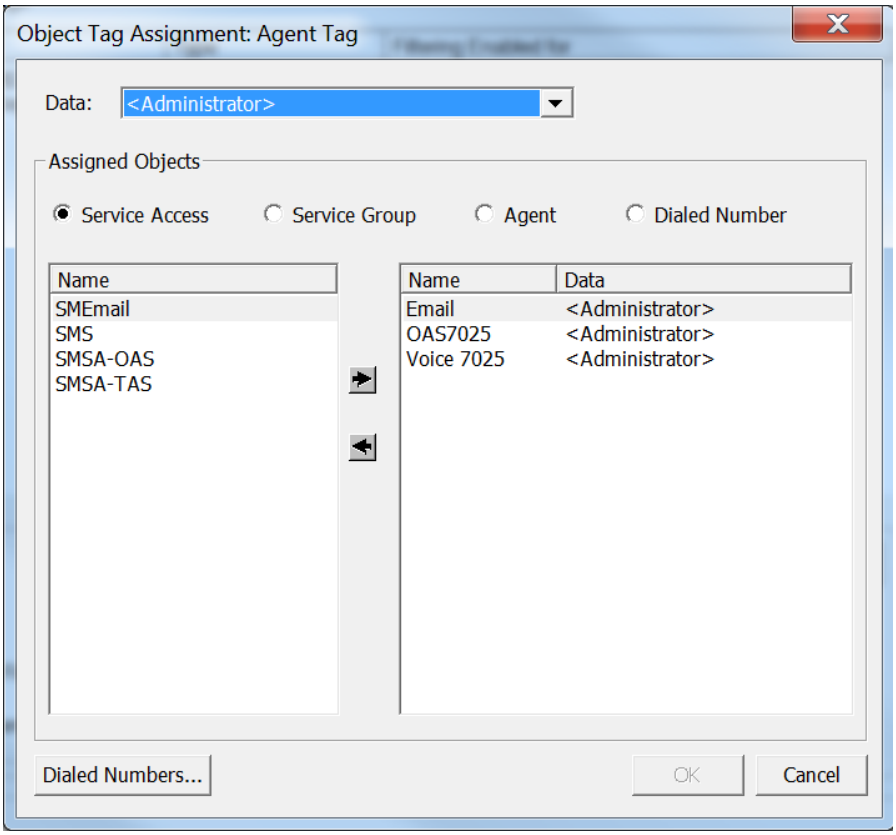
Object tags are defined using Configuration Manager. The Object Tags tab of the Tenant System Properties displays all defined object tags for the tenant. Up to 50 object tags are permitted per tenant.



Individual object tags can be marked as allowing assignment to service accesses, service groups, agents and/or dialed numbers by checking the appropriate box under ***Allow Object Tag for***. Note that if an object tag is already assigned to an object type, that object type cannot be unchecked until the object is unassigned from the object tag.

After an object tag is defined, it can be assigned to one or more objects. To assign an object tag, press the ***Assign to Objects*** button to display the Object Tag Assignment dialog. This dialog displays all objects currently associated with the object tag. It also allows assignment of additional objects to the object tag.

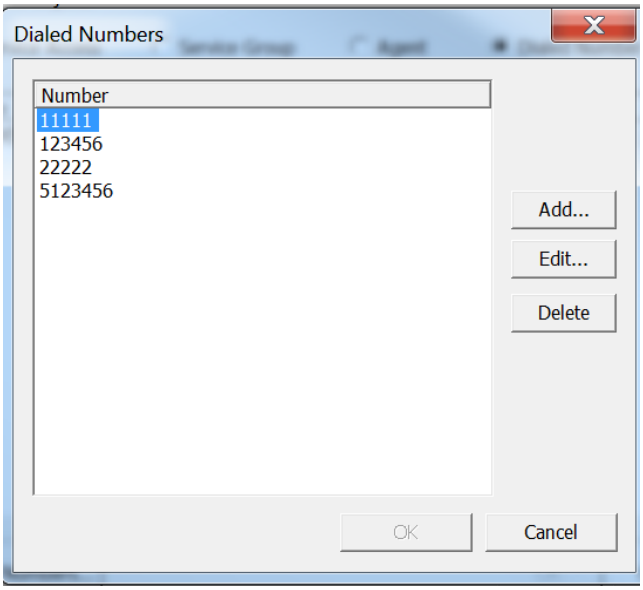
Note that objects can be assigned from the Service Access, Service Group, Agent or Dialed Number categories. If an assigned object type is not checked when the object tag is defined, the assigned object type will be disabled and an object of that type cannot be associated with the object tag.



When an object is assigned to an object tag, associated data must be selected from the Data field. The Data field will represent the type of data associated with an object tag. For example, if an object tag is of type Agent, as in the example above, the associated Data values will be any of the defined agents in the MiCC-E system. If the object tag is of type Number or Text, a value must be entered in the Data field to represent the number or text value for the object tag association.

When the properties for an Agent, Service Group or Service Access are displayed, it is possible to view and assign object tags to the specific Agent, Service Group or Service Access object. The Object Tag page displays all defined object tags in the system which are enabled for the selected object type, and allows assignment of the object tag to the Agent, Service Group or Service Access object.

Dialed Numbers are managed through the Object Tag Assignment dialog. Pressing the ***Dialed Numbers…*** button displays the Dialed Numbers dialog, allowing management of Dialed Numbers.



This dialog lists all defined Dialed Numbers for the tenant. It is used to add new Dialed Numbers, as well as edit or delete existing numbers. The number field can contain any character and may be up to 50 characters long. Dialed Numbers must be unique within a tenant, so the same number cannot be defined multiple times.

Note that Dialed Numbers are committed to the database as soon as updates are made in this dialog. This makes them immediately available for object tag assignment in the Object Tag Assignment dialog.

The object tag information is stored in the MiCC-E database, where it can be used to query information about objects, such as through Script Manager scripts.

The tables are defined as follows:

object_tag

Column	Type	Description
id	int	Unique identifier for the object tag
name	varchar(100)	Descriptive name for the object tag
type	int	Type of the tag, which can be number, text, service access object, service group object, scheduler object, agent object or play message object
tenant_id	int	Tenant associated with the entry
sa_enabled	Bit	Indicates whether the object tag can be assigned to service accesses or not
sg_enabled	bit	Indicates whether the object tag can be assigned to service groups or not
agent_enabled	bit	Indicates whether the object tag can be assigned to agents or not

dialednumber_enabled	bit	Indicates whether the object tag can be assigned to dialed numbers or not

Constraints: The same tenant cannot have identifiers with the same name and type.

object_tag_mapping

Column	Type	Description
object_tag_id	int	Corresponds to the id column in the object_tag table
object_tag_value	varchar(100)	The value assigned to the object tag for this object. This field depends on the object_tag type. For number and text values, it is the actual value. For service access, service group, scheduler, agent and play message objects, it is the object ID.
object_id	int	Unique identifier of the tagged object. This is the ID of the service group, service access, agent or dialed number to which the object tag is assigned.
object_type	int	Type of the tagged object, which is service group (0x0200), service access (0x0400), agent (0x1000) or dialed number (0x100000).

Constraints: The combination of object_tag_id, object_type and object_id must be unique.

AGENT ADVANCED CONFIGURATION

Advanced configuration options for Agent may be changed by modifying the file:

<InstallDir>\Applications\Bin\Agent.exe.config

These settings apply to all users running Agent on same computer. All settings are contained in the following section:

```
<configuration>
	<applicationSettings>
		<Solidus.ITA.Properties.Settings>
```

SETTING	DEFAULT	DESCRIPTION
OutlookCacheUpdateInterval	60	Update interval in seconds between refreshing Outlook contacts.
MaxEmailSearchResults	500	Maximum number of e-mail results in Customer search.
MaxChatSearchResults	500	Maximum number of chat results in Customer search.
MaxVoiceSearchResults	500	Maximum number of voice call results in Customer search.
MaxSMSSearchResults	500	Maximum number of SMS results in Customer search.

MaxKBResults	10	Maximum number of KB entries returned in KB lookup.
SIPReRegistrationTime	3600	Interval in seconds between SIP registration attempts to the call manager. 3600 indicates that call manager should determine the interval.
M5T_MinAudioPort	40000	Minimum audio port used for softphone communication.
M5TMaxAudioPort	42000	Maximum audio port used for softphone communication.
M5T_MinVideoPort	40000	Minimum video port used for softphone communication. The value is currently not used.
M5T_MaxVideoPort	42000	Maximum video port used for softphone communication. This value is currently not used.
PresenceServerPollingInterval	30000	Interval in milliseconds between presence server polling requests.

TRANSFER OF SERVICE GROUP VOICE CALLS

When a service group call is transferred from one agent to another agent, it will continue as a service group call on the target agent. The call can be transferred either before or after it is answered by the target agent. If the call is transferred before answer and the target agent does not answer before the ring supervision time, the call will be returned to the service group queue and the target agent will be forced to voice not ready state.

For call reporting, only the last agent handling the call will be counted as handling a call for the service group. If a call is transferred multiple times, the last agent completing the call is reported as the agent handling the call for the service group. It will be possible for an agent to have a count of 0 Answered Calls in the Agent report, but include time as Voice Busy, if the agent transferred a service group call.

Reports for the service group will calculate Alerting time as the duration of time that the service group call alerted at the first agent. Servicing time will include the time that the first agent spent in Talking state for the call, as well as time that other agents spent in Talking state after the call was transferred to the other agents. Similarly, Hold state will include the time that the first agent had the call on hold, as well as the time that other agents to which the call was transferred had the call on hold. Clerical time for the call will only be the time that the last agent handling the call spent in Clerical state.

This feature only applies to incoming voice calls, and not web callbacks or callback calls. If either of these types of calls are transferred, they will be completed when transferred and handled as a private call on the target agent.

Note that this only applies to agents running the Agent application, and not Web Agents or Phone Agents.

MICROSOFT AZURE INITIAL CONFIGURATION FOR USER SYNCHRONIZATION AND SINGLE SIGN-ON

In order to use Microsoft Azure Active Directory with User Synchronization or Single Sign-On as described in the following sections, some initial configuration must be performed in Azure AD. MiCC Enterprise will only work with a single Azure AD tenant per MiCC Enterprise tenant. It is assumed that a tenant has already been created in Azure AD and the tenant already contains the defined users that will be used with MiCC Enterprise.

1. Logon to the Microsoft Azure Portal and select *Manage Azure Active Directory*.
2. Switch to the desired Tenant if necessary.
3. Note the *Tenant ID* displayed on the *Overview* page. It will be used in the following sections.
4. Select the *App registrations* page.
5. Select the *New registration* button.
6. Enter the name of the application to be used with MiCC Enterprise. The name is unimportant, but for future use in this document, it will be referred to as MICCE. Select the *Register* button.
7. Note the *Application (client) ID*. It will be used in the following sections.

This section described registering the application to be used with MiCC Enterprise. Single Sign-On may use Identity Providers other than Microsoft Azure. Those Identity Providers may have a similar registration process. A new application must be registered with the Identity Provider. An application (client) ID is required for use with Single Sign-On.

USER SYNCHRONIZATION

Users from Microsoft Azure Active Directory and on-premise Active Directory can be automatically synchronized with MiCC Enterprise.

CONFIGURING MICROSOFT AZURE AD

The initial configuration of Microsoft Azure must be performed as described in the previous section *Microsoft Azure Initial Configuration for User Synchronization and Single Sign-On*. Some additional configuration is necessary for user synchronization.

1. Logon to the Microsoft Azure Portal and select *Manage Azure Active Directory*.
2. Switch to the desired Tenant if necessary.

3. Select the *App registrations* page.

4. Select the MiCCE application.

5. Select the *API permissions* page.

6. Select the *Add a permission* button.

7. Locate and select the *Microsoft Graph* group.

8. Select *Application permissions*.

9. The following permissions must be added for *Microsoft Graph.*

   User.Read.All
   Group.Read.All
   GroupMember.Read.All
   Directory.Read.All
   RolManagement.Read.Directory

10. After all permissions are checked, select the *Add permissions* button.

11. Back on the API permissions page, select the Grant admin consent for XXXX button and confirm the request.

12. Switch to the *Certificates and secrets* page.

13. Select *New client secret*.

14. Enter a description and expiration for the secret and select Add.

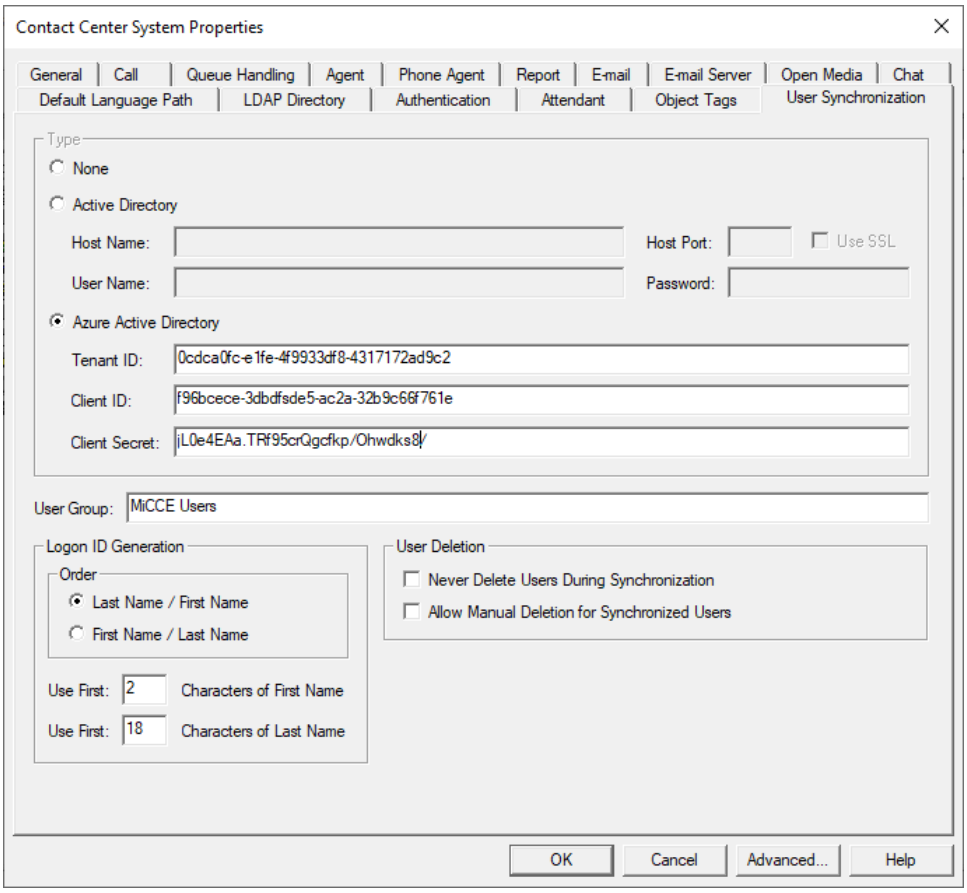
15. Copy the value for the secret. This will be used when configuring MiCC Enterprise. Note that the next time you logon to the portal you will not be able to view the secret value. If the secret value is lost or unknown, you must create a new secret value.

16. Switch back the main configuration page for the tenant. Select the *Groups* page.

17. MiCC Enterprise will synchronize all users belonging to a specified group. If you don't already have a Microsoft Azure AD Group containing the users to be synchronized, a new group must be created. Create a new security group and add the desired users as members. Note the name of the existing or newly created group. It will be used when configuring MiCC Enterprise.

CONFIGURING MICC ENTERPRISE

User synchronization is configured on the *User Synchronization* tab of *Tenant Properties* in *Configuration Manager*.



Select the type of synchronization to perform.

Active Directory

Enter the server and account information for connecting to the on-premise Active Directory. If the connection information is blank, a connection will be established to local domain.

Azure Active Directory

Enter the Tenant ID, Application (Client) ID and Client Secret configured in Microsoft Azure AD setup in the previous sections.

User Group

Specify the name of the user group in Active Directory or Microsoft Azure AD for which users will be retrieved.

Logon ID Generation

When new users are created during the synchronization process, the MiCC-E logon ID is generated according to the these parameters. The logon ID is made up of up to the specified number of characters from the first and last names of the user. The maximum length of the combined number of characters is 20. Examples:

First Name: Karen
Last Name: Sharp

First Name Characters: 2
Last Name Characters: 18
Format: First/Last
Result: kasharp

First Name Characters: 2
Last Name Characters: 18
Format: Last/First
Result: sharpka

First Name Characters: 10
Last Name Characters: 10
Format: Last/First
Result: sharpkaren

First Name Characters: 17
Last Name Characters: 3
Format: Last/First
Result: shakaren

User Deletion

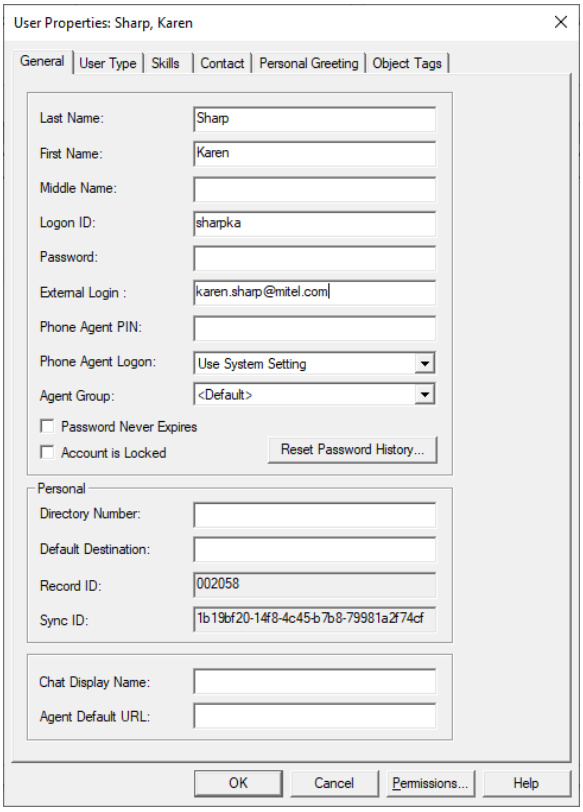
During the synchronization process, previously synchronized users are deleted if they no longer exist in the specified user group. The option to never delete these users may be enabled.

By default, users created during the synchronization process cannot be manually deleted in Configuration or Web Manager. The option to allow manually deleting these users may be enabled. Note that if a user is deleted, the next synchronization process may re-add the user if the user still exists in the specified user group.

SYNCHRONIZATION PROCESS

Synchronization is performed by the MiCC Agent Service (CCSolidusAgent). Synchronization is always performed for all tenants for which synchronization is configured. Users are retrieved

from AD and created, updated or deleted in MiCC Enterprise as necessary. MiCC Enterprise users that are synchronized with AD will have a Sync ID displayed on the General page of the User Properties in Configuration Manager.



New Users

If a MiCC Enterprise user is not found with the corresponding SyncID, a new user is created. If a previously synched user was found that is deleted, the user will be undeleted. A new user will not be created. The user will maintain all previous settings.

- The Logon ID will be generated according to the configured Logon ID Generation parameters. If an existing user is found with the same Logon ID, a number will be appended to the Logon ID of the new user until a unique ID is available. For example. sharpka1, sharpka2, etc. The Administrator may change the generated Logon ID in the *User Properties* from Configuration or Web Manager.

- User will be assigned to the <Default> agent group and user type. The user will not have permission to run any MiCC Enterprise applications. An Administrator must assign the user to the correct agent group and user type through Configuration or Web Manager.

- The password will be blank. Users with blank passwords are not allowed to logon to MiCC Enterprise using the conventional logon process using username/password. The user may only logon using the single sign-on process. To allow the user to logon using the conventional

process, an Administrator must assign a password to the user through Configuration or Web Manager.

- The External Logon ID is assigned to the user using the User Principal Name (upn) or e-mail address of the AD user. The External Logon ID is used to locate the user during the single sign-on process.

Updated Users

If a user is found with the corresponding SyncID, the user will be updated with the latest information in MiCC Enterprise. The following fields are not changed when updating the user:

Logon ID
Password
Agent Group
User Type

Deleted Users

Previously synchronized users that are no longer retrieved from AD will be deleted if the option to never delete these users is not enabled. If a user is mistakenly removed from the AD user group, it will be deleted. If that user is later re-added to the user group, the next synchronization process will undelete the user retaining all previous settings.

AUDIT LOG

Logging of users that are created, updated or deleted during the synchronization process is made to the following file:

<InstallDir>\Services\Bin\Log\UserSync.csv

This is a standard comma-delimited text file that can be opened in programs such as Notepad or Microsoft Excel. The format of each line of the file is:

“<Date/Time>”,”<Action>”,”<SyncID>”,<RecID>,”<LogonID>”,”<FirstName>”,”<LastName>”,”<ExternalLogonID>”,”<Comment>”

Date/Time: Date and time the user was added, updated or deleted in the format YYYY-MM-DD HH:MM:SS.

Action: Action that was performed. Actions include ADD, UPDATE, DELETE, UNDELETE and ERROR.

SyncID: ID retrieved from AD. This is the object ID uniquely identifying the AD user.

RecID: Internal MiCC Enterprise record ID.

LogonID: MiCC Enterprise user logon ID.

FirstName: User first name.

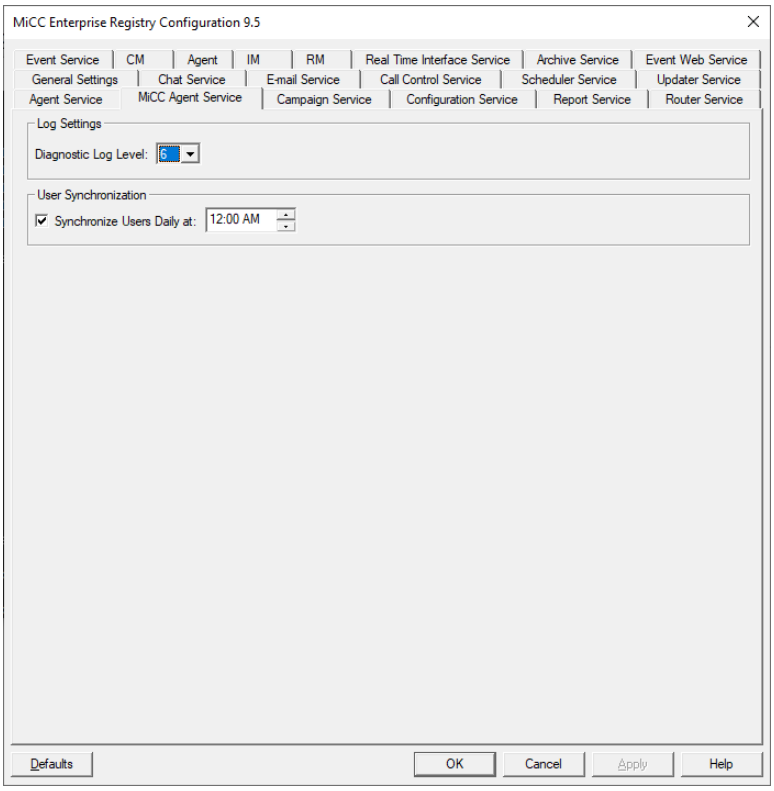
LastName: User last name.

ExternalLogonID: External logon ID used for single sign-on. This is generally the user principal name (upn) or e-mail address of the user.

Comment: General comment (if any) about the action.

EXECUTION AND SCHEDULING

By default, synchronization is performed automatically once per day at a specified time. This may be disabled, or the time changed on the MiCC Agent Service page in the MiCC Enterprise Registry Configuration utility. Synchronization should be performed during off-peak hours when possible.



Synchronization may also be executed manually by running the following command from a CMD prompt:

<InstallDir>\Services\Bin\AgentService.exe /adsync

SINGLE SIGN-ON

MiCC Enterprise supports single sign-on authentication using an external identity provider (IDP) when logging into desktop applications, Web Manager or Web Agent. The IDP must support the Open ID Connect protocol. Authentication is handled entirely by the IDP. MiCC Enterprise has no knowledge of or access to the user credentials supplied. The user identifier is returned to MiCC Enterprise during the authentication process and the associated MiCC Enterprise user is logged into the application.

This section will describe using single sign-on with Microsoft Azure AD or ADFS; however, any IDP may be used that supports the Open ID Connect protocol.

CONFIGURING MICROSOFT AZURE AD

If Microsoft Azure is being used for Single Sign-On, perform the following steps:

The initial configuration of Microsoft Azure must be performed as described in the previous section *Microsoft Azure Initial Configuration for User Synchronization and Single Sign-On*. Some additional configuration is necessary for single sign-on.

1. Logon to the Microsoft Azure Portal and select *Manage Azure Active Directory*.

2. Switch to the desired Tenant if necessary.

3. Select the *App registrations* page.

4. Select the MiCCE application.

5. Select the Authentication page.

6. 3 custom redirect URIs must be configured for use with MiCC Enterprise. If the platform has not been created, select the *Add a platform* button and choose the necessary platform type.

   Platform: Mobile and desktop application
   Redirect URI: http://localhost

   Platform: Web
   Redirect URI: https://<domain>/webapps/contactcenter/authentication/openidlogincallback

   Platform: Single-page application
   Redirect URI: https://<domain>/webagent

<domain> should be the fully qualified name of the MiCC Enterprise Web server including the port if necessary. For example:

https://mitel.com:4390/webagent

The redirect URIs are case sensitive and should be entered in lower case. The redirect URI passed to the identity provider is constructed from the URL used to access the Web application. A redirect URI must be configured using the same <domain> that is used to access the Web application.

The MiCC Enterprise Web server must be configured to use the https protocol.

7. After adding all redirect URIs, *Implicit grant* becomes available on the *Authentication* page. Enable implicit grant flow for Access and ID tokens and select the *Save* button.

8. Select the *Token configuration* page.

9. Select *Add optional claim*.

10. Select the ID token type and select the *upn* claim. Select the *Add* button to add the claim.

11. Select the *Overview* page.

12. Select the *Endpoints* button.

13. Note the *OpenID Connect metadata document* entry. This will be used when configuring MiCC Enterprise.

CONFIGURATION ADFS (ACTIVE DIRECTORY FEDERATED SERVICES)

If ADFS is being used for Single Sign-On, perform the following steps:

1. Open the ADFS Configuration Management console.

2. Right click on the *Application Groups* node and select *Add Application Group…*

3. Enter a name for the application group such as MiCC Enterprise. Select the *Native application* template and click Next.

4. Note the *Client identifier* value. It will be used when configuring MiCC Enterprise.

   Add the following 3 Redirect URIs:

http://localhost
https://<domain>/webapps/contactcenter/authentication/openidlogincallback
https://<domain>/webagent

<domain> should be the fully qualified name of the MiCC Enterprise Web server including the port if necessary. For example:

https://mitel.com:4390/webagent

The redirect URIs are case sensitive and should be entered in lower case. The redirect URI passed to the identity provider is constructed from the URL used to access the Web application. A redirect URI must be configured using the same <domain> that is used to access the Web application.

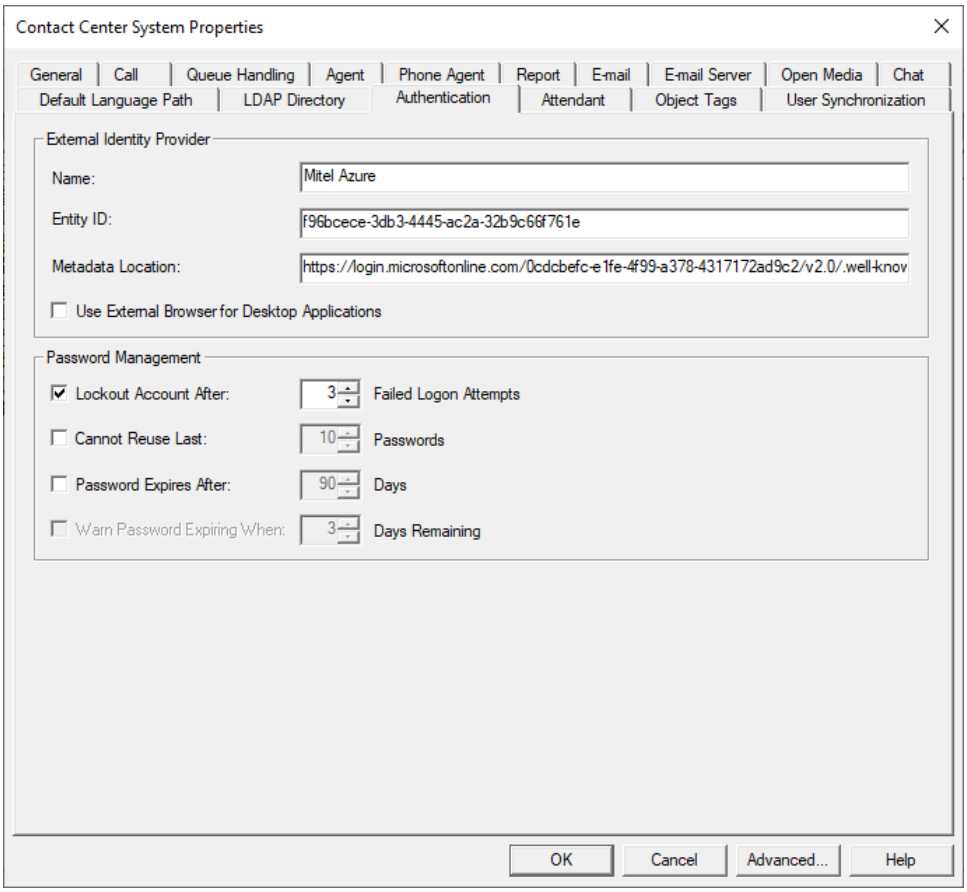
The MiCC Enterprise Web server must be configured to use the https protocol.

5. Complete the remaining pages in the Add Application Group Wizard.

Note that the single sign-on process for WebAgent requires CORS support on the IDP. CORS is only supported with ADFS 2019 or later. Using an earlier version of ADFS will most likely require the use of a Web Application Proxy for communicating with the IDP from WebAgent.

CONFIGURING MICC ENTERPRISE

Single sign-on is configured on the *Authentication* tab of *Tenant Properties* in *Configuration Manager*.



Enter the Name, Entity ID and Metadata location.

Entity ID: Application (client) ID of the registered application in the IDP.

Metadata Location: URL used to retrieve the metadata document from the IDP. In most cases for Microsoft Azure AD, it will be required to add an additional query parameter to the metadata document URL displayed in the Endpoints in Microsoft Azure AD. The format of the parameter is:

?appid=<Application (client) ID>

For example, if the URL displayed in the Endpoints is:

https://login.microsoftonline.com/0cd345fc-e1fe-4f99-a378-4317172ad9c2/v2.0/.well-known/openid-configuration

and the application (client) ID is:

f96bcece-3db3-4445-ac2a-32b9c66f761e

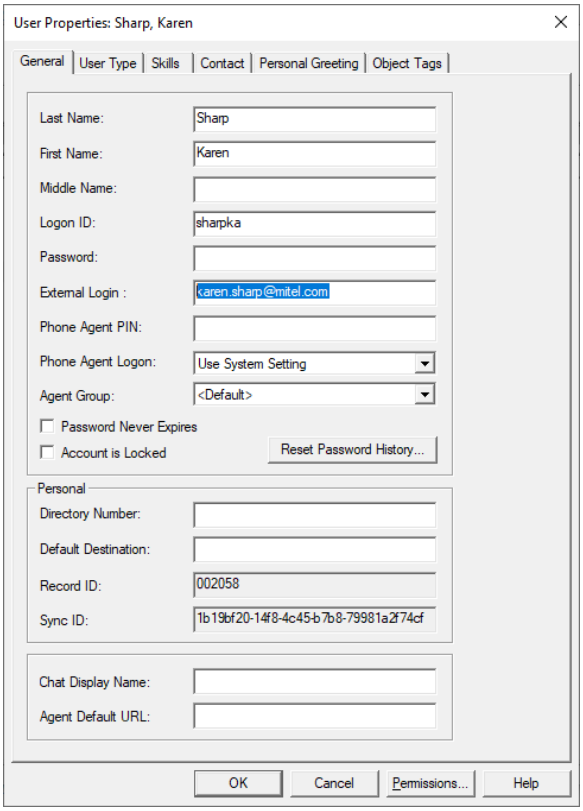
The correct Metadata Location would be:

https://login.microsoftonline.com/0cd345fc-e1fe-4f99-a378-4317172ad9c2/v2.0/.well-known/openid-configuration?appid=f96bcece-3db3-4445-ac2a-32b9c66f761e

Use External Browser for Desktop Applications: By default, an internal Web browser is used to display the authentication page of the IDP. If this option is checked, the default Web browser installed on the computer will be used.

User Matching

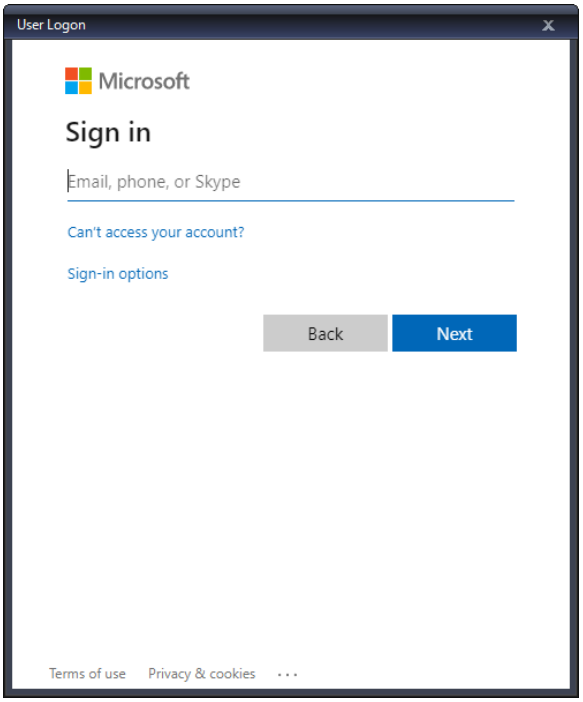
Upon successful authentication of a user by the IDP, the *User Principal Name* (upn) is returned by the IDP. The MiCC Enterprise user is located by matching the upn to the External Login ID configured for the MiCC Enterprise user.



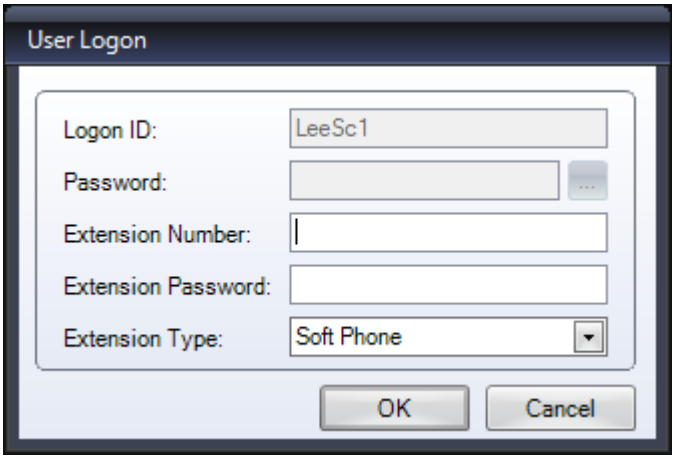
When user synchronization is enabled in MiCC Enterprise, the External Login ID is automatically populated for the synchronized users. For users manually created in MiCC Enterprise, an External Logon ID must be entered that matches the upn of the IDP user. Typically, the upn is the same as the e-mail address, but the IDP may use a different identifier.

SINGLE SIGN-ON PROCESS

If an External Identity Provider is configured for the MiCC Enterprise tenant, single sign-on is always used for authentication in the MiCC Enterprise desktop applications. On startup of the application, the IDP login page will be displayed. The authentication process is controlled entirely by the IDP.



After successful authentication, if any additional information is needed for the MiCC Enterprise application, the standard logon form will be displayed allowing entry of the additional information. For example, MiCC Agent requires the extension number to use.



The standard logon form is displayed with the Logon ID and Password disabled. In most cases, the additional information can be passed on the command line when starting the application and this prompt will be suppressed. See the section *Application Command Line Parameters* in this document for a list of possible parameters.

Single sign-on is always used if configured when starting the desktop applications. Conventional logon using standard MiCC Enterprise user credentials can be forced by supplying the /stdlogin parameter on the command line. For example:

Agent.exe /stdlogin

It can also be forced by holding the SHIFT key down while starting the application.

Logging into the Web applications Web Manager and Web Agent allows standard logon as well as single sign-on. Enter the standard MiCC Enterprise user credentials or click the <Identity Provider Name> button to use the single sign-on process of the IDP.

SINGLE SIGN-ON USING SAML 2.0 IN WEB MANAGER

In previous versions of MiCC Enterprise, Web Manager supported single sign-on using the SAML 2.0 protocol. This functionality is still supported; however, the standard External Identity Provider configuration will always use the Open ID Connect protocol. Configuring Web Manager to use the SAML 2.0 protocol requires manual changes to the web.config file for Web Manager. For more information, see the *External Authentication Using SAML 2.0* section in the *Web Applications Configuration Guide*.