A MITEL
PRODUCT
GUIDE

MiContact Center Enterprise

Real Time Interface User Guide

Release 9.6
Document Version 1.0

September 2022

Mitel
Powering connections

Notices

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

Trademarks

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

®,™ Trademark of Mitel Networks Corporation

INTRODUCTION

The MiCC Enterprise Real Time Interface Service provides a solution for networking independent MiCC Enterprise systems. With the networked solution, the MiCC Enterprise systems are connected as a complete virtual call center. The concept of this solution is similar to that of Automatic Networked Call Distribution (ANCD). Using MiCC Enterprise Script Manager, it is possible to make routing decisions for incoming calls across the entire set of networked MiCC Enterprise nodes, rather than just on the local node. Calls can be distributed via the virtual network to find the best node to handle the call in the most efficient manner.

SYSTEM ARCHITECTURE AND COMPONENTS

The Real Time Interface Service coexists with the MiCC Enterprise services. See Figure 1 for an illustration of a networked MiCC Enterprise solution.

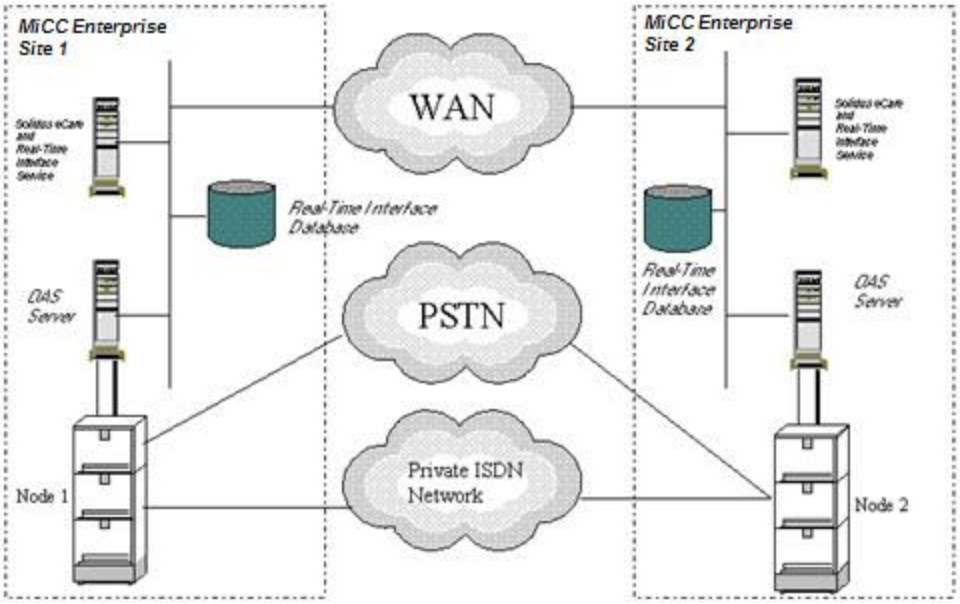

Figure 1: Networked MiCC Enterprise system

As the figure shows, each MiCC Enterprise site in the network will have the Real Time Interface Service installed, along with the Real Time Interface Database.

MICC ENTERPRISE REAL TIME INTERFACE SERVICE

The MiCC Enterprise Real Time Interface Service resides on a Windows Server (or it may co-reside on the MiCC Enterprise Server) in the local network for the MiCC Enterprise site. At startup, it will connect to the Broker Service to retrieve the locations of other MiCC Enterprise services and the MiCC Enterprise database. It will also connect to the Configuration Service to retrieve service group information, and to receive updates to configuration data.

The Real Time Interface Service also connects to the MiCC Enterprise Event Service. The Real Time Interface Service will subscribe to the Event Service in order to receive real-time events about service group access activity and agent group activity. As information events are received (approximately every 5 seconds), the service will process the event to determine if it is for a service group currently being monitored for this site. If so, it will write the data to the Real Time Interface Data-base.

Note: It is also possible to configure the threshold value of 5 seconds via a configuration option, in order to accommodate sites with less traffic.

The Real Time Interface Service also connects to other remote MiCC Enterprise sites which have been configured for connection. When performance updates are received, the Real Time Interface Service will broad- cast the data to other Real Time Interface Services in networked MiCC Enterprise sites. Remote Real Time Interface Services will write the data, along with the appropriate site ID, to their local Real Time Interface Data-base.

REAL TIME INTERFACE DATABASE

Each Real Time Interface Service will create a Real Time Interface Database when it is run for the first time. This is just an additional table within the MiCC Enterprise SQL Server database. The database contains information about monitored service groups from each MiCC Enterprise site, which have been selected for monitoring.

The database allows the MiCC Enterprise system to have a preview of the current performance on other MiCC Enterprise nodes, in order to make accurate, timely routing decisions between MiCC Enterprise nodes.

Information stored includes:

1. Site ID of the MiCC Enterprise node.
2. Service Group Name and ID.
3. Current Service Level for the Service Group.
4. Current Skill Choice (1, 2 or 3) for the Service Group.
5. Current Longest Call Wait Time for the Service Group. This value is stored as a time duration value.
6. Number of agents in the Service Group for each of the following categories:
7. Free - Idle.
8. Busy - Currently serving a call for this service group.
9. Busy Other - Currently serving a call for another service group.
10. Logged On - Logged on to the MiCC Enterprise system.
11. Not Ready - Currently not ready for service group calls.
12. Number of calls in queue for this Service Group.

13. Estimated Waiting Time for the Service Group. This value is stored as a time duration value. Three values are included, one for each skill choice.
14. Current number of answered calls based on the performance interval.
15. Current number of abandoned calls based on the performance interval.
16. Current number of calls overflowed out based on the performance interval.
17. Timestamp of when the information was last updated. It can be used by the script to determine whether the information is current or not, in case of loss of connection to another node's Real Time Interface Service.

Each MiCC Enterprise site will have the same Real Time Interface Database, because it will contain data for its local service groups, as well as data for service groups on other networked MiCC Enterprise sites.

SCRIPT MANAGER

As calls arrive to a Service Access in a MiCC Enterprise site, the appropriate call flow as defined by Script Manager will execute. If this particular call flow is programmed to check remote MiCC Enterprise sites, Script Manager can access the Real Time Interface Database table, via Open Database Connectivity (ODBC), to check the performance of configured service groups at other MiCC Enterprise nodes. If another node is determined to be a better option for this call, due to decreased wait time, the call will be deflected to the other node. At that node, it will be routed accordingly, based on the Service Access to which it is deflected and the routing rules for that MiCC Enterprise node.

This solution uses capabilities already in Script Manager. The only requirement is to program the call flow properly, to perform the ODBC database lookup and use the data in the table to compare performance with other MiCC Enterprise sites.

COMPATIBILITY

System requirements for the MiCC Enterprise Real Time Interface Service:

- MiCC Enterprise with Script Manager.
- For Windows Server requirement, see document Installation Preparations - MiCC Enterprise.
- For Windows SQL Server requirement, see document Installation Preparations - MiCC Enterprise.

INSTALLATION

MiCC Enterprise Real Time Interface Service is installed along with an existing MiCC Enterprise system.

Note: In order for the Real Time Interface Service to start up successfully, the MiCC Enterprise Broker and Configuration services must be running and the Real Time Interface site license must be installed.

CONFIGURATION

After installation is complete, it is necessary to configure the Real Time Interface Services. You should determine how Real Time Interface Services will communicate with each other. One service should be configured as the master and the remaining services should be configured as clients. Only the master may accept connections from other RTIs. An RTI service must connect to a master. If a single RTI is to be used, it should be configured as a master. Refer to the Registry Settings section for more information on master/client settings.

Once the Real Time Interface Services are running with their proper master/client settings, it is necessary to select the service groups, agents and service accesses that should be monitored in the local MiCC Enterprise system.

To configure the system, perform the following procedure:

1. Ensure that the MiCC Enterprise Real Time Interface Service is running via the Services control applet.
2. From Configuration Manager, click **Real Time Interface** on the **Tools** menu. The **Real** Ti**me Interface Service Configuration** dialogue box will appear. It consists of five tabs: **Service Group**s, **Age**nts, **Service Access**, **General** and **Connection Status**.

   By default, the Service Groups tab will appear. The Service Groups tab allows you to select which service groups will be monitored by the local Real Time Interface Service. These are the service groups for which data will be stored in the local database and broadcast to other networked MiCC Enterprise sites for storage in the remote databases.

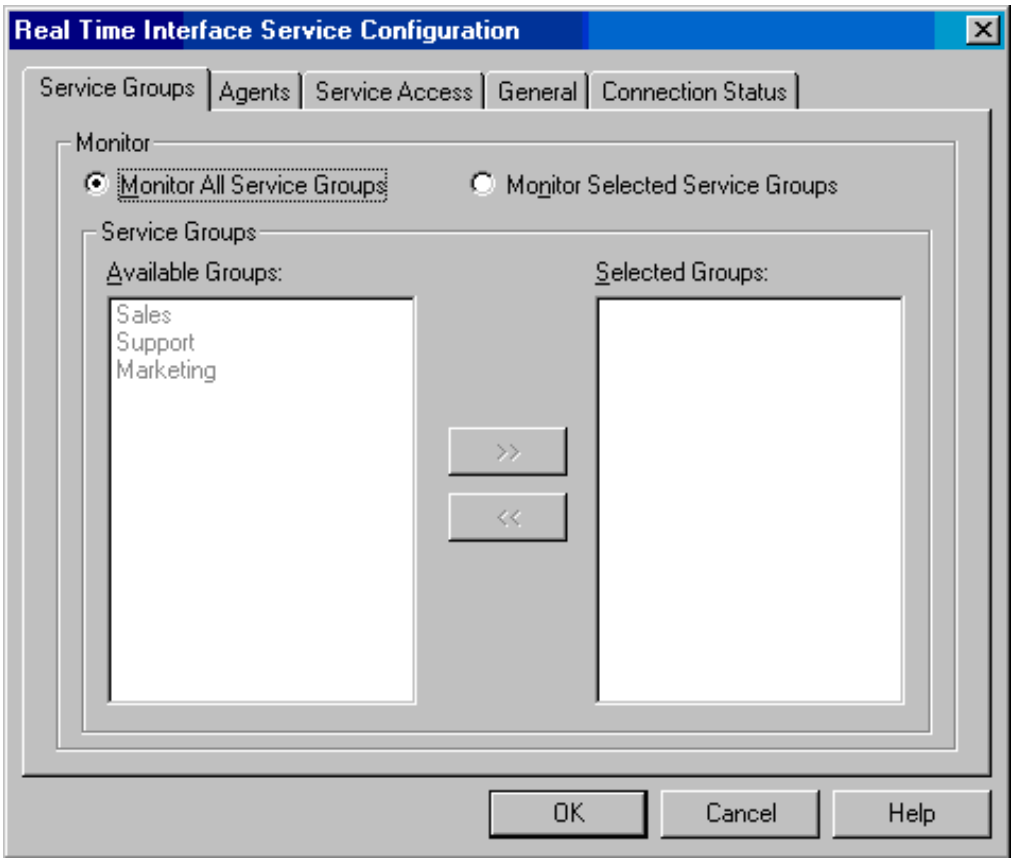

Table 1:

OBJECT	DESCRIPTION
Monitor All Service Groups option button	If selected, all defined service groups will be monitored by the Real Time Interface Service. **Note**: If this button is selected, all future service groups added will automatically be monitored.
Monitor Selected Service Groups option button	If selected, only the groups in the Selected Groups list will be monitored by the Real Time Interface Service. The maximum number of groups that may be selected is 1000.
Available Groups list	Lists all the defined service groups not currently being monitored by the Real Time Interface Service.
Selected Groups list	Lists all the service groups currently being monitored by the Real Time Interface Service.

3. Complete the necessary information. Click **Agents** tab. The **Agents** tab will appear. The **Agents** tab allows you to configure which agents should be monitored by the local Real Time Interface Service. These are the agents for which data will be stored internally and broadcast to other networked MiCC Enterprise sites. Agent information is not stored in the Real Time Interface Service Database. It is only stored internally for use with the API Interface.

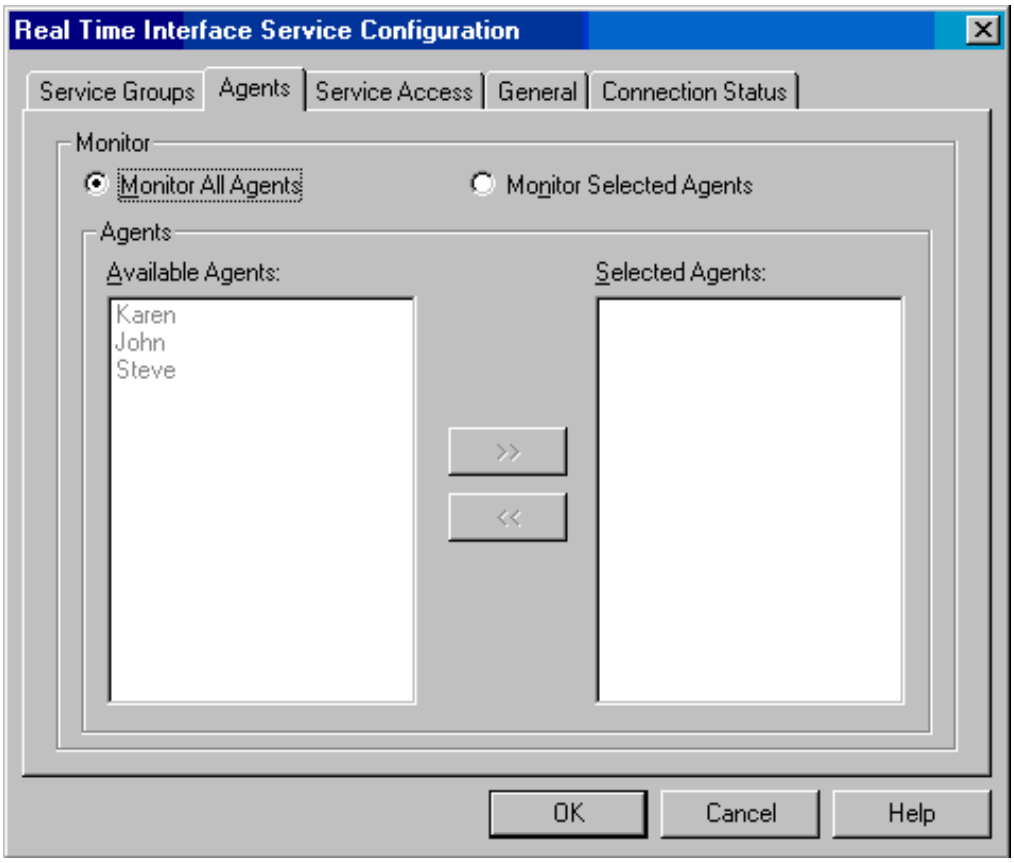

Table 2:

OBJECT	DESCRIPTION
Monitor All Agents option button	If selected, all defined agents will be monitored by the Real Time Interface Service. **Note**: If this button is selected, all future agents added will automatically be monitored.
Monitor Selected Agents option button	If selected, only the agents in the Selected Agents list will be monitored by the Real Time Interface Service.
Available Agents list	Lists all the defined agents not currently being monitored by the Real Time Interface Service.

4. Complete the necessary information. Click **Service Access** tab.

The **Service Access** tab will appear. The **Service Access** tab allows you to configure which service accesses should be monitored by the local Real Time Interface Service. These are the service access for which data will be stored internally and broadcast to other networked MiCC Enterprise sites. Service access information is not stored in the Real Time Interface Service Database. It is only stored internally for use with the API Interface.

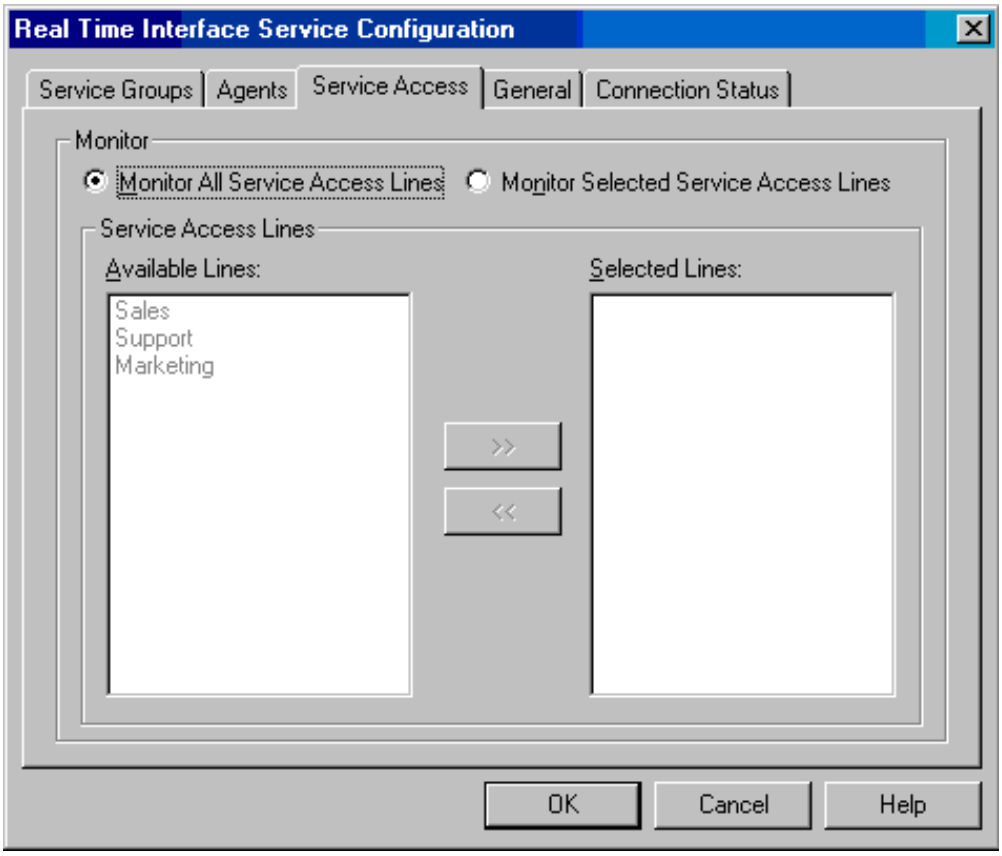

Table 3:

OBJECT	DESCRIPTION
Monitor All Service Access Lines option button	If selected, all defined service accesses will be monitored by the Real Time Interface Service. **Note**: If this button is selected, all future service accesses added will automatically be monitored.
Monitor Selected Service Access Lines option button	If selected, only the service accesses in the Selected Lines list will be monitored by the Real Time Interface Service.
Available Lines list	Lists all the defined service accesses not currently being monitored by the Real Time Interface Service.
Selected Lines list	Lists all the service accesses currently being monitored by the Real Time Interface Service.

5. Complete the necessary information. Click the General tab. The General tab will appear. The General tab allows you to configure parameters for the Real Time Interface Service.

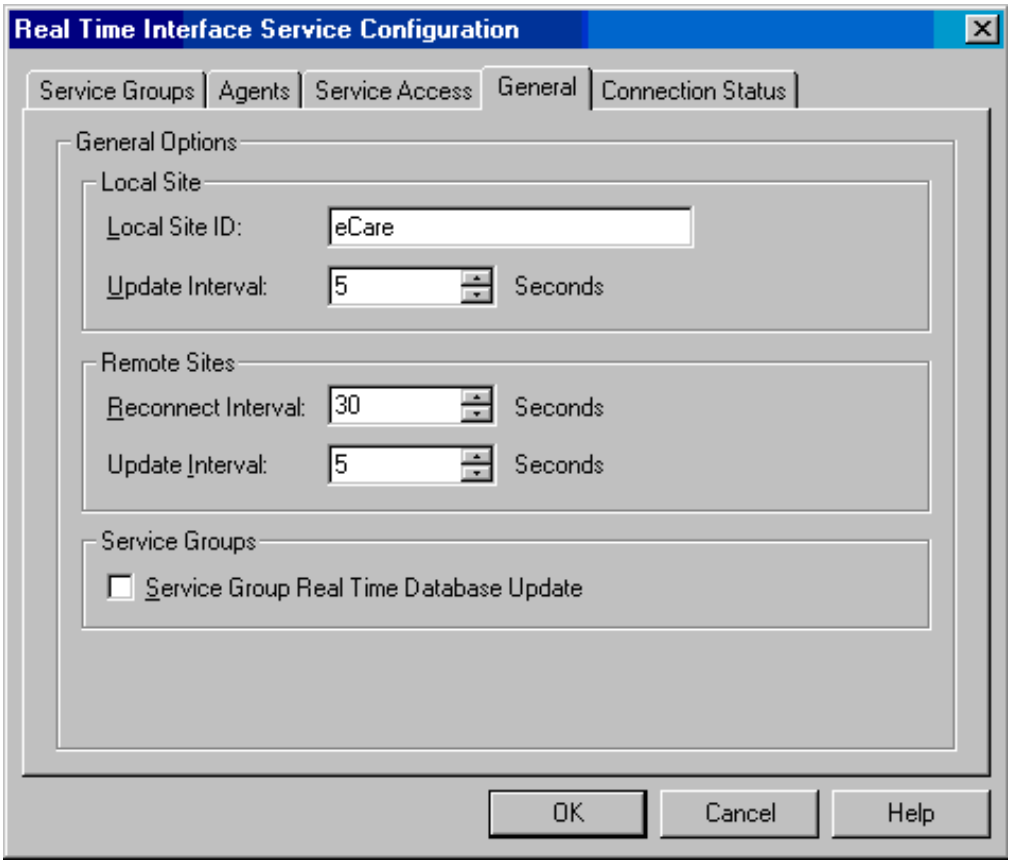

Table 4:

OBJECT	DESCRIPTION
Local Site ID text box	Unique site ID for the Real Time Interface Service on the local MiCC Enterprise node. The default at installation is the name of the machine on which the Real Time Interface Service is installed. **Note**: This field is required, since it is used to distinguish data for this site in the Real Time Interface Database Site IDs must be unique among MiCC Enterprise sites; otherwise, connection to other MiCC Enterprise Real Time Interface Services would fail.
Local Site Update Interval spin box	Indicates the amount of time between updates to the local database of performance data. The default is 5 seconds.
Remote Sites Reconnect Interval spin box	Indicates the amount of time between attempts to connect to services before reattempting. The default is 30 seconds.
Remote Sites Update Interval spin box	Indicates the amount of time between updates sent to the master Real Time Interface Service. The default is 5 seconds.
Service Group Real Time Database Update checkbox	Enables updating of the Real Time Interface Service Database. This option must be enabled in order for service group information to be written to the local and remote databases.
Disable COM Client Interface check box	Check this box to disable the client interface, allowing outside applications to connect to the RTI Service to receive updates and send requests.

6. Complete the necessary information. Click the **Connection Status** tab. The C**onnection Status** tab will appear. The **Connection Status** tab identifies the MiCC Enterprise sites to which this MiCC Enterprise site is connected as a client as well as the status of the current MiCC Enterprise sites networked to this MiCC Enterprise site.

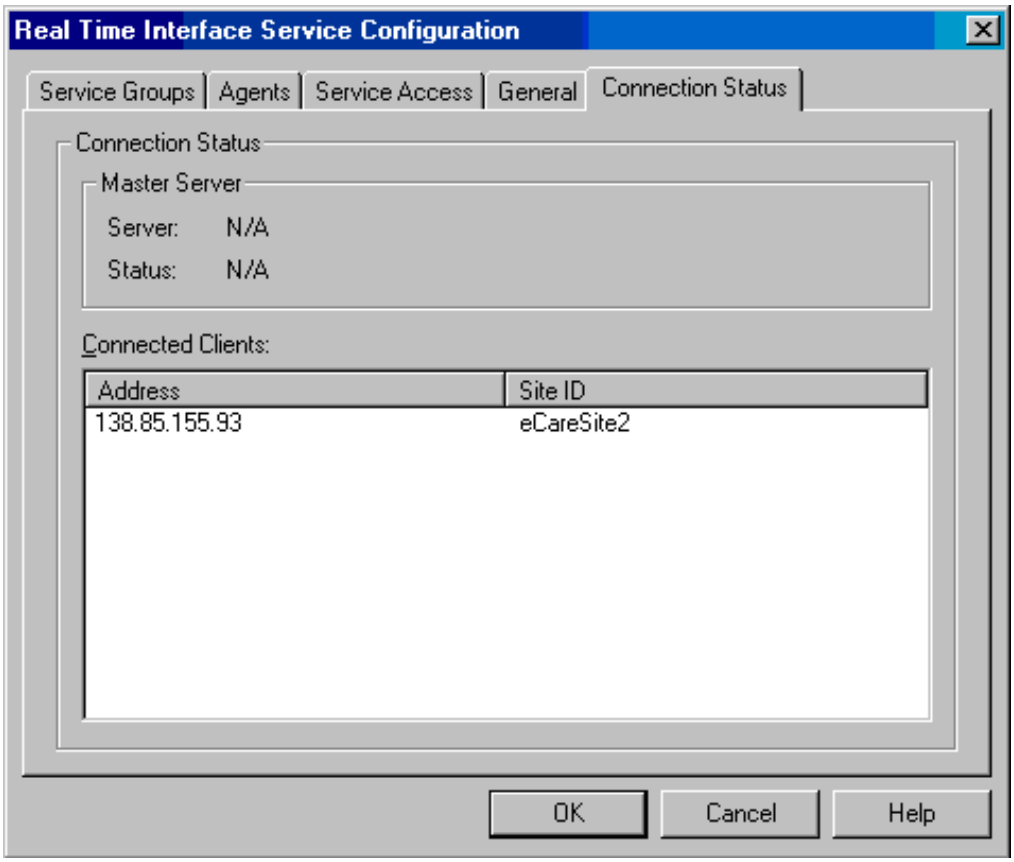

Table 5:

OBJECT	DESCRIPTION
Master Server Name text box	Identifies the name of the master server if this Real Time Interface Service is not configured as the master.
Master Server Status text box	Identifies the connection status of the master server if this Real Time Interface Service is not configured as the master.
Connected Clients list box	Identifies the sites that are connected to this Real Time Interface Service if configured as a master.

Note: If the MiCC-Enterprise system is tenanted, ensure that you configure the Real Time Interface site license for the <Default> tenant.

API INTERFACE

The Real Time Interface Service also supports an API interface via the NetMonClient.DLL. This DLL implements a COM object which can be used by external, third-party programs to communicate with the Real Time Interface Service in order to receive agent and service group real-time performance data.

During installation of the Real Time Interface Service, the NetMonClient.DLL will be installed and registered, so it is ready to be accessed. To interface with the service, the application should create an instance of the INetMonClientObj. Details are provided below.

INTERFACE OBJECT DETAILS

The following section describes interface object details.

INETMONCLIENTOBJ

This is the main object for communicating with the Real Time Interface Service.

Table 6:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
ServerName	BSTR	Read/ Write	Specifies the name of the machine where the Real Time Interface Service is running. If the Real Time Interface Service is running on the same machine, this attribute will be set automatically. Otherwise, it should be set prior to calling Connect().
ServerPort	long	Read/ Write	Specifies the TCP/IP port number on which the Real Time Interface Service will accept connections. If the Real Time Interface Service is running on the same machine, this attribute will be set automatically. Otherwise, it should be set prior to calling Connect().
UseSSL	VARIANT_BOOL	Read/Write	Specifies if a secure SSL connection should be used when connecting to the Real Time Interface Service.
STAEvents	VARIANT_BOOL	Read/Write	If enabled, events will be sent on the same thread in which the INetMonClientObj interface was initialized.

Table 7:

METHODS SUPPORTED

METHOD	PURPOSE
Connect()	Connects the application to the Real Time Interface Service. If the application is on a different machine than the Real Time Interface Service, the ServerName and ServerPort attributes must be configured prior to calling Connect.
Disconnect()	Disconnects from the Real Time Interface Service.
GetAgentList (BSTR SiteID, IAgentList** ppAgentList)	Retrieves an IAgentList object containing a list of all defined agents which the Real Time Interface Service is monitoring. Parameters: SiteID - Specifies which site to retrieve the list of agents for. If Site ID is empty, agents for all sites are retrieved. ppAgentList - Contains a pointer to an IAgentList object, which contains the defined agents
GetServiceGroupList(BSTR SiteID, IServiceGroupList** ppGroupList)	Retrieves an IServiceGroupList object containing a list of all defined service groups, which the Real Time Interface Service is monitoring. Parameters: SiteID - Specifies which site to retrieve the list of service groups for. If Site ID is empty, service groups for all sites are retrieved ppGroupList - Contains a pointer to an IServiceGroupList object, which contains the defined service groups
GetServiceAccessList(BSTR SiteID, IServiceAccessList ** ppServiceAccessList)	Retrieves an IServiceAccessList object containing a list of all defined service accesses, which the Real Time Interface Service is monitoring. Parameters: SiteID - Specifies which site to retrieve the list of service accesses for. If Site ID is empty, service accesses for all sites are retrieved ppServiceAccessList - Contains a pointer to an IServiceAccessList object, which contains the defined service accesses
GetSiteIDList (ISiteIDList ** ppSiteIDList)	Retrieves an ISiteIDList object containing a list of all sites connected in the system. This list will always contain the local site that INetMonClientObj is connected to in additional to the other sites in the system. Parameters: ppSiteIDList - Contains a pointer to an ISiteIDList object, which contains the connected sites in the system

METHODS SUPPORTED

METHOD	PURPOSE
GetAgentPerformance(BSTR SiteID, long lAgentID, IAgentPerformance** ppAgentPerformance)	Retrieves an IAgentPerformance object containing performance data for the specified site and agent ID. Only performance data for a single agent may be retrieved. To retrieve performance data for multiple agents, use GetAgentPerformanceList or GetAgentPerformanceListEx. Parameters: SiteID - Specifies the site of the agent for which performance data should be retrieved lAgentID - Specifies the unique ID of the agent for which performance data should be retrieved ppAgentPerformance - Contains a pointer to an IAgentPerformance object, which contains performance data for the indicated agent
GetAgentPerformanceList(BSTR SiteID, long lAgentID, IAgentPerformanceList** ppAgentPerformanceList)	Retrieves an IAgentPerformanceList object containing a list of performance data for the specified site and agent ID. Parameters: SiteID - Specifies the site of the agent(s) for which performance data should be retrieved. If SiteID is empty, performance data for all agents is retrieved lAgentID - Specifies the unique ID of the agent for which performance data should be retrieved. If agent ID is 0, performance data for all agents specified by SiteID is retrieved. If SiteID is empty, this parameter is ignored. ppAgentPerformanceList - Contains a pointer to an IAgentPerformanceList object, which contains a list of performance data for the specified site and agent ID
GetAgentPerformanceListEx(BSTR SiteID, VARIANT AgentIDs, IAgentPerformanceList** ppAgentPerformanceList)	Retrieves an IAgentPerformanceList object containing a list of performance data for the specified site and agent IDs. Parameters: SiteID - Specifies the site of the agent(s) for which performance data should be retrieved. If SiteID is empty, performance data for all agents is retrieved AgentIDs □ Specifies a VARIANT containing a list of agent IDs for which performance data should be retrieved. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, performance data for all agents specified by SiteID is retrieved. If SiteID is empty, this parameter is ignored. ppAgentPerformanceList - Contains a pointer to an IAgentPerformanceList object, which contains a list of performance data for the specified site and agent IDs
GetAgentPerformanceListExByString(BSTR SiteID, BSTR AgentIDs, IAgentPerformanceList** ppAgentPerformanceList)	Similar to GetAgentPerformanceListEx, but supports applications that are unable to use the VARIANT type. In this method, the AgentIDs parameter is passed as a BSTR, with each agent ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceGroupPerformance(BSTR SiteID, long lGroupID, IServiceGroupPerformance** ppGroupPerformance)	Retrieves an IServiceGroupPerformance object containing performance data for the specified service group ID. Only performance data for a single service group may be retrieved. To retrieve performance data for multiple service groups, use GetServiceGroupPerformanceList or GetServiceGroupPerformanceListEx. Parameters: SiteID - Specifies the site of the service group for which performance data should be retrieved lGroupID - Specifies the unique ID of the group for which performance data should be retrieved ppGroupPerformance - Contains a pointer to an IServiceGroupPerformance object, which contains performance data for the indicated service group
GetServiceGroupPerformanceList(BSTR SiteID, long lGroupID, IServiceGroupPerformanceList** ppServiceGroupPerformanceList)	Retrieves an IServiceGroupPerformanceList object containing a list of performance data for the specified site and group ID. Parameters: SiteID - Specifies the site of the service group(s) for which performance data should be retrieved. If SiteID is empty, performance data for all service groups is retrieved lGroupID - Specifies the unique ID of the service group for which performance data should be retrieved. If group ID is 0, performance data for all service groups specified by SiteID is retrieved. If SiteID is empty, this parameter is ignored. ppServiceGroupPerformanceList - Contains a pointer to an IServiceGroupPerformanceList object, which contains a list of performance data for the specified site and group ID
GetServiceGroupPerformanceListEx(BSTR SiteID, VARIANT GroupIDs, IServiceGroupPerformanceList** ppServiceGroupPerformanceList)	Retrieves an IServiceGroupPerformanceList object containing a list of performance data for the specified site and group IDs. Parameters: SiteID - Specifies the site of the service group(s) for which performance data should be retrieved. If SiteID is empty, performance data for all service groups is retrieved GroupIDs □ Specifies a VARIANT containing a list of service group IDs for which performance data should be retrieved. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, performance data for all service groups specified by SiteID is retrieved. If SiteID is empty, this parameter is ignored. ppServiceGroupPerformanceList - Contains a pointer to an IServiceGroupPerformanceList object, which contains a list of performance data for the specified site and group IDs
GetServiceGroupPerformanceListExByString(BSTR SiteID, BSTR ServiceGroupIDs, IServiceGroupPerformanceList** ppServiceGroupPerformanceList)	Similar to GetServiceGroupPerformanceListEx, but supports applications that are unable to use the VARIANT type. In this method, the ServiceGroupIDs parameter is passed as a BSTR, with each group ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceAccessPerformance(BSTR SiteID, long lServiceAccessID, IServiceAccessPerformance** ppServiceAccessPerformance)	Retrieves an IServiceAccessPerformance object containing performance data for the specified service access ID. Only performance data for a single service access may be retrieved. To retrieve performance data for multiple service accesses, use GetServiceAccessPerformanceList or GetServiceAccessPerformanceListEx. Parameters: SiteID - Specifies the site of the service access for which performance data should be retrieved lServiceAccessID - Specifies the unique ID of the service access for which performance data should be retrieved ppServiceAccessPerformance - Contains a pointer to an IServiceAccessPerformance object, which contains performance data for the indicated service access
GetServiceAccessPerformanceList(BSTR SiteID, long lServiceAccessID, IServiceAccessPerformanceList** ppServiceAccessPerformanceList)	Retrieves an IServiceAccessPerformanceList object containing a list of performance data for the specified site and service access ID. Parameters: SiteID - Specifies the site of the service access(s) for which performance data should be retrieved. If SiteID is empty, performance data for all service accesses is retrieved lServiceAccessID - Specifies the unique ID of the service access for which performance data should be retrieved. If service access ID is 0, performance data for all service accesses specified by SiteID is retrieved. If SiteID is empty, this parameter is ignored. ppServiceAccessPerformanceList - Contains a pointer to an IServiceAccessPerformanceList object, which contains a list of performance data for the specified site and service access ID
GetServiceAccessPerformanceListEx(BSTR SiteID, VARIANT ServiceAccessIDs, IServiceAccessPerformanceList** ppServiceAccessPerformanceList)	Retrieves an IServiceAccessPerformanceList object containing a list of performance data for the specified site and service access IDs. Parameters: SiteID - Specifies the site of the service access(s) for which performance data should be retrieved. If SiteID is empty, performance data for all service accesses is retrieved ServiceAccessIDs □□Specifies a VARIANT containing a list of service access IDs for which performance data should be retrieved. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, performance data for all service accesses specified by SiteID is retrieved. If SiteID is empty, this parameter is ignored. ppServiceAccessPerformanceList - Contains a pointer to an IServiceAccessPerformanceList object, which contains a list of performance data for the specified site and service access IDs

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceAccessPerformanceListExByString(BSTR SiteID, BSTR ServiceAccessIDs, IServiceAccessPerformanceList** ppServiceAccessPerformanceList)	Similar to GetServiceAccessPerformanceListEx, but supports applications that are unable to use the VARIANT type. In this method, the ServiceAccessIDs parameter is passed as a BSTR, with each service access ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.
SetAgentSubscription(BSTR SiteID, VARIANT AgentIDs)	Sets the list of agents to receive performance data events for. Performance data events will only be received for agents in the subscription list. Configuration events for all agents will be received regardless of subscription. Parameters: SiteID - Specifies the site of the agent(s) to set in the subscription list. If SiteID is empty, all agents will be set in the subscription list as well as any new agents added to the system in the future AgentIDs Specifies a VARIANT containing a list of agent IDs to set in the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all agents specified by SiteID are set in the subscription list as well as any new agents added to the site in the future. If SiteID is empty, this parameter is ignored.
SetAgentSubscriptionByString(BSTR SiteID, BSTR AgentIDs)	Similar to SetAgentSubscription, but supports applications that are unable to use the VARIANT type. In this method, the AgentIDs parameter is passed as a BSTR, with each agent ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.
AddAgentSubsciption(BSTR SiteID, VARIANT AgentIDs)	Adds to the list of agents to receive performance data events for. Performance data events will only be received for agents in the subscription list. Configuration events for all agents will be received regardless of subscription. Parameters: SiteID - Specifies the site of the agent(s) to add to the subscription list. If SiteID is empty, all agents will be added to the subscription list as well as any new agents added to the system in the future AgentIDs Specifies a VARIANT containing a list of agent IDs add to the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all agents specified by SiteID are added to the subscription list as well as any new agents added to the site in the future. If SiteID is empty, this parameter is ignored.
AddAgentSubsciptionByString(BSTR SiteID, BSTR AgentIDs)	Similar to AddAgentSubscription, but supports applications that are unable to use the VARIANT type. In this method, the AgentIDs parameter is passed as a BSTR, with each agent ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.

METHODS SUPPORTED

METHOD	PURPOSE
RemoveAgentSubsciption (BSTR SiteID, VARIANT AgentIDs)	Removes from the list of agents to receive performance data events for. Performance data events will only be received for agents in the subscription list. Cconfiguration events for all agents will be received regardless of subscription. Parameters: SiteID - Specifies the site of the agent(s) to remove from the subscription list. If SiteID is empty, all agents will be removed from the subscription list AgentIDs Specifies a VARIANT containing a list of agent IDs to remove from the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all agents specified by SiteID are removed from the subscription list. If SiteID is empty, this parameter is ignored
RemoveAgentSubsciption ByString(BSTR SiteID, BSTR AgentIDs)	Similar to RemoveAgentSubscription, but supports applications that are unable to use the VARIANT type. In this method, the AgentIDs parameter is passed as a BSTR, with each agent ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.
SetServiceGroupSubscription(BSTR SiteID, VARIANT GroupIDs)	Sets the list of service groups to receive performance data events for. Performance data events will only be received for service groups in the subscription list. Configuration events for all service groups will be received regardless of subscription. Parameters: SiteID - Specifies the site of the service group(s) to set in the subscription list. If SiteID is empty, all service groups will be set in the subscription list as well as any new service groups added to the system in the future GroupIDs Specifies a VARIANT containing a list of service group IDs to set in the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all service groups specified by SiteID are set in the subscription list as well as any new service groups added to the site in the future. If SiteID is empty, this parameter is ignored
SetServiceGroupSubscriptionByString(BSTR SiteID, BSTR GroupIDs)	Similar to SetServiceGroupSubscription, but supports applications that are unable to use the VARIANT type. In this method, the GroupIDs parameter is passed as a BSTR, with each group ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.

METHODS SUPPORTED

METHOD	PURPOSE
AddServiceGroupSubscription(BSTR SiteID, VARIANT GroupIDs)	Adds to the list of service groups to receive performance data events for. Performance data events will only be received for service groups in the subscription list. Configuration events for all service groups will be received regardless of subscription. Parameters: SiteID - Specifies the site of the service group(s) to add to the subscription list. If SiteID is empty, all service groups will be added to the subscription list as well as any new service groups added to the system in the future GroupIDs Specifies a VARIANT containing a list of service group IDs add to the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all service groups specified by SiteID are added to the subscription list as well as any new service groups added to the site in the future. If SiteID is empty, this parameter is ignored.
AddServiceGroupSubscriptionByString(BSTR SiteID, BSTR GroupIDs)	Similar to AddServiceGroupSubscription, but supports applications that are unable to use the VARIANT type. In this method, the GroupIDs parameter is passed as a BSTR, with each group ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.
RemoveServiceGroupSubscription(BSTR SiteID, VARIANT GroupIDs)	Removes from the list of service groups to receive performance data events for. Performance data events will only be received for service groups in the subscription list.Configuration events for all service groups will be received regardless of subscription. Parameters: SiteID - Specifies the site of the service group(s) to remove from the subscription list. If SiteID is empty, all service groups will be removed from the subscription list GroupIDs Specifies a VARIANT containing a list of service group IDs to remove from the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all service groups specified by SiteID are removed from the subscription list. If SiteID is empty, this parameter is ignored
RemoveServiceGroupSubscriptionByString(BSTR SiteID, VARIANT GroupIDs)	Similar to RemoveServiceGroupSubscription, but supports applications that are unable to use the VARIANT type. In this method, the GroupIDs parameter is passed as a BSTR, with each group ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.

METHODS SUPPORTED

METHOD	PURPOSE
SetServiceAccessSubscription(BSTR SiteID, VARIANT ServiceAccessIDs)	Sets the list of service accesses to receive performance data events for. Performance data events will only be received for service accesses in the subscription list. Configuration events for all service accesses will be received regardless of subscription. Parameters: SiteID - Specifies the site of the service access(s) to set in the subscription list. If SiteID is empty, all service accesses will be set in the subscription list as well as any new service accesses added to the system in the future ServiceAccessIDs Specifies a VARIANT containing a list of service access IDs to set in the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all service accesses specified by SiteID are set in the subscription list as well as any new service accesses added to the site in the future. If SiteID is empty, this parameter is ignored
SetServiceAccessSubscriptionByString(BSTR SiteID, BSTR ServiceAccessIDs)	Similar to SetServiceAccessSubscription, but supports applications that are unable to use the VARIANT type. In this method, the ServiceAccessIDs parameter is passed as a BSTR, with each service access ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.
AddServiceAccessSubscription(BSTR SiteID, VARIANT ServiceAccessIDs)	Adds to the list of service accesses to receive performance data events for. Performance data events will only be received for service accesses in the subscription list. Cconfiguration events for all service accesses will be received regardless of subscription. Parameters: SiteID - Specifies the site of the service access(s) to add to the subscription list. If SiteID is empty, all service accesses will be added to the subscription list as well as any new service accesses added to the system in the future ServiceAccessIDs Specifies a VARIANT containing a list of service access IDs to add to the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all service accesses specified by SiteID are added to the subscription list as well as any new service accesses added to the site in the future. If SiteID is empty, this parameter is ignored.
AddServiceAccessSubscriptionByString(BSTR SiteID, BSTR ServiceAccessIDs)	Similar to AddServiceAccessSubscription, but supports applications that are unable to use the VARIANT type. In this method, the ServiceAccessIDs parameter is passed as a BSTR, with each service access ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.

METHODS SUPPORTED

METHOD	PURPOSE
RemoveServiceAccessSubscription(BSTR SiteID, VARIANT ServiceAccessIDs)	Removes from the list of service accesses to receive performance data events for. Performance data events will only be received for service accesses in the subscription list. Configuration events for all service accesses will be received regardless of subscription. Parameters: SiteID - Specifies the site of the service access(s) to remove from the subscription list. If SiteID is empty, all service accesses will be removed from the subscription list ServiceAccessIDs Specifies a VARIANT containing a list of service access IDs to remove from the subscription list. This VARIANT must contain an array of 4 byte integers. If there are no elements in the array, all service accesses specified by SiteID are removed from the subscription list. If SiteID is empty, this parameter is ignored
RemoveServiceAccessSubscriptionByString(BSTR SiteID, BSTR ServiceAccessIDs)	Similar to RemoveServiceAccessSubscription, but supports applications that are unable to use the VARIANT type. In this method, the ServiceAccessIDs parameter is passed as a BSTR, with each service access ID separated by the | character. For example, 1|2|3|. Note that there should be a | character after the last ID provided.
GetLocalSiteID(BSTR * pSiteID)	Retrieves the ID of the local site.

INETMONCLIENTOBJNOTIFY

This is the object used to receive notifications when performance data is updated.

Table 8:

EVENTS SUPORTED

EVENT	PURPOSE	ACTION
OnAgentPerformanceUpdate(BSTR SiteID, long lAgentID)	Called when performance data for an agent has changed. Parameters: SiteID - Specifies the site of the agent that has been updated. lAgentID - Specifies the unique ID of the agent that has been updated	GetAgentPerforma nce() should be called on the INetMonClientObj with the indicated site and agent ID in order to retrieve the current performance data for the agent.

EVENTS SUPORTED

EVENT	PURPOSE	ACTION
OnServiceGroupPerformanceUpdate (BSTR SiteID, long lGroupID)	Called when performance data for a service group has changed. Parameters: SiteID - Specifies the site of the service group that has been updated. lGroupID - Specifies the unique ID of the service group that has been updated	GetServiceGroupP erformance() should be called on the INetMonClientObj with the indicated site and service group ID in order to retrieve the current performance data for the group.
OnServiceAccessPerformanceUpdate(BSTR SiteID, long lServiceAccessID)	Called when performance data for a service access has changed. Parameters: SiteID - Specifies the site of the service access that has been updated. lServiceAccessID - Specifies the unique ID of the service access that has been updated	GetServiceAccess Performance() should be called on the INetMonClientObj with the indicated site and service access ID in order to retrieve the current performance data for the service access.
OnAgentConfigurationUpdate(long lAction, BSTR SiteID, long lAgentID, BSTR NewName)	Called when configuration data for an agent has changed. Parameters: lAction - Indicates the type of update that occurred as follows: 1: Agent added 2: Agent deleted 3: All agents deleted 4: Agent name changed SiteID - Specifies the site of the agent, which has been updated. If lAction is 3, this parameter will be empty if all agents in the system are deleted. Otherwise, all agents at the specified site are deleted lAgentID - Specifies the unique ID of the agent which has been updated. If lAction is 3, this parameter is undefined NewName □ If lAction is 1 or 4, this specifies the new name of the agent	If agents are deleted, all performance data for the agents should be removed. Individual performance data events will not be received.

EVENTS SUPORTED

EVENT	PURPOSE	ACTION
OnServiceGroupConfigurationUpdate (long lAction, BSTR SiteID, long lGroupID, BSTR NewName)	Called when configuration data for a service group has changed. Parameters: lAction - Indicates the type of update that occurred as follows: 1: Service group added 2: Service group deleted 3: All service groups deleted 4: Service group name changed SiteID - Specifies the site of the service group, which has been updated. If lAction is 3, this parameter will be empty if all service groups in the system are deleted. Otherwise, all service groups at the specified site are deleted lGroupID - Specifies the unique ID of the service group which has been updated. If lAction is 3, this parameter is undefined NewName □ If lAction is 1 or 4, this specifies the new name of the group	If service groups are deleted, all performance data for the groups should be removed. Individual performance data events will not be received.

EVENTS SUPORTED

EVENT	PURPOSE	ACTION
OnServiceAccess ConfigurationUpda te(long lAction, BSTR SiteID, long lServiceAccessID, BSTR NewName)	Called when configuration data for a service access has changed. Parameters: lAction - Indicates the type of update that occurred as follows: 1: Svc access added 2: Svc access deleted 3: All svc accesses deleted 4: Svc access name changed SiteID - Specifies the site of the svc access, which has been updated. If lAction is 3, this parameter will be empty if all svc accesses in the system are deleted. Otherwise, all svc accesses at the specified site are deleted lServiceAccessID - Specifies the unique ID of the svc access which has been updated. If lAction is 3, this parameter is undefined NewName If lAction is 1 or 4, this specifies the new name of the svc access	If service accesses are deleted, all performance data for the service accesses should be removed. Individual performance data events will not be received.
OnSiteIDUpdate(long lAction, BSTR OldName, BSTR NewName)	Called when configuration data for a site in the system has changed. Parameters: lAction - Indicates the type of update that occurred as follows: 1: Site added 2: Site deleted 3: Site name changed OldName If lAction is 2 or 3, specifies the current or old name of the site. Otherwise undefined NewName If lAction is 1 or 3, specifies the new name of the site	If a site is deleted from the system, all configuration and performance data for that site should be discarded. Individual performance and configuration events for service groups, agents and service accesses will not be received.
OnDisconnect()	Called when a disconnect from the Real Time Interface Service is detected. Note: If a call to Disconnect() was made, this event is not signaled.	

EVENTS SUPORTED

EVENT	PURPOSE	ACTION
OnConfigurationServiceConnection(long lAction)	Called when a change in connection status is detected between the Real Time Interface Service and the Configuration Service. Parameters: lAction □ Indicates the current connection status: 1: Connection broken 2: Connection established	
OnEventServiceConnection(long lAction)	Called when a change in connection status is detected between the Real Time Interface Service and the Event Service. Parameters: lAction □ Indicates the current connection status: 1: Connection broken 2: Connection established	

ISERVICEGROUPPERFORMANCE

This object contains performance data for a single service group.

Table 9:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
BusyAgents	long	Read	Specifies the number of agents busy handling calls for this service group.
BusyOther Agents	long	Read	Specifies the number of agents skilled to handle calls for this service group, but currently handling calls for another service group.
CallsInQueue	long	Read	Specifies the current number of calls queued for this service group.
CurrentSkillChoice	long	Read	Specifies the current skill choice configured.
EstimatedWaitTimeChoice1	long	Read	Specifies the estimated wait time in seconds for skill choice 1.
EstimatedWaitTimeChoice2	long	Read	Specifies the estimated wait time in seconds for Skill Choice 2.

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
EstimatedWaitTimeChoice3	long	Read	Specifies the estimated wait time in seconds for Skill Choice 3.
FreeAgents	long	Read	Specifies the number of agents skilled to handle calls for this service group and currently idle.
SiteID	BSTR	Read	Specifies the site this group belongs to.
GroupID	long	Read	Specifies the unique group identifier for this service group.
GroupName	BSTR	Read	Specifies the service group name.
LoggedOnAgents	long	Read	Specifies the number of logged on agents skilled to handle calls for this service groups.
LongestWaitTime	long	Read	Specifies the longest time in seconds that a call has been waiting in queue for this service group.
ServiceLevel	long	Read	Specifies the current service level percentage value for this service group.
TimeStamp	DATE	Read	Specifies the time that performance data was last updated for this service group.
UnavailableAgents	long	Read	Specifies the number of agents not ready to answer calls, but skilled to handle calls for this service group.
AnsweredCallCount	long	Read	Specifies the number of calls answered for this service group.
AbandonedCallCount	long	Read	Specifies the number of calls abandoned for this service group.
OverflowedOutCallCount	long	Read	Specifies the number of calls overflowed out for this service group.
AbandonedRate	long	Read	Specifies the current abandoned rate.
AcceptedCallbacks	long	Read	Specifies the number of accepted callbacks.
FailedCallbacks	long	Read	Specifies the number of failed callbacks.
CallbacksInQueue	long	Read	Specifies the number of callbacks currently in the queue.

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
ScheduledCallbacks	long	Read	Specifies the number of callbacks currently scheduled.

IAGENTPERFORMANCE

This object contains performance data for a single agent.

Table 10:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
AgentID	long	Read	Specifies the unique agent identifier for this agent.
ServiceGroupCount	long	Read	Specifies the number of service group performance objects for this agent.
LogonName	BSTR	Read	Specifies the logon name of the agent.
SiteID	BSTR	Read	Specifies the site this agent belongs to.
LogonExtension	BSTR	Read	Specifies the current extension the agent is logged onto.
AgentState	long	Read	Specifies the current agent state where: 0 = Logged On 1 = Logged Off 2 = Ready 3= Not Ready
CallState	long	Read	Specifies the current voice call state where: 0 = Undefined 1 = Idle 2 = Ringing Service Call 3 = Talking Service Call 4 = Parked Service Call 5 = Service Call in Clerical 6 = Callback Service Call 7 = Setting Up Service Call 8 = Personal Incoming Call 9 = Personal Outgoing Call
CallSubState	long	Read	Specifies the current voice call sub-state where: 0 = Talking 1 = Assisting 2 = Assisted 3 = Monitoring 4 = Monitored

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
VoiceReady	long	Read	Indicates if the agent is ready to receive voice calls where: 0 = No 1 = Yes
EmailReady	long	Read	Indicates if the agent is ready to receive e-mail sessions where: 0 = No 1 = Yes
CSRReady	long	Read	Indicates if the agent is ready to receive CSR sessions where: 0 = No 1 = Yes
NotReadyReason	BSTR	Read	Returns the Reason entered when changing state to Not Ready for voice calls, if available.
TimeStamp	DATE	Read	Specifies the time that performance data was last updated for this agent.

Table 11:

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceGroupPerformance(long lGroupIndex, IAgentServiceGroupPerformance** ppGroupPerformance)	Retrieves an IAgentServiceGroupPerformance object for the specified service group index. Parameters: lGroupIndex - Specifies the index for which to retrieve group information. The valid range is from 1 to ServiceGroupCount. ppGroupPerformance - Contains a pointer to an IAgentServiceGroupPerformance object if the method is successful.

IAGENTSERVICEGROUPPERFORMANCE

This object contains performance data for an agent/service group rela- tionship. All data pertains to the performance of the indicated agent for a single service group only.

Table 12:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
AgentID	long	Read	Specifies the unique identifier for the agent.
GroupID	long	Read	Specifies the unique identifier for the service group.
SiteID	BSTR	Read	Specifies the site this agent/service group belongs to.
AbandonedCallCount	long	Read	Specifies the number of calls for the indicated service group which have abandoned (disconnected prior to answer) while ringing at this agent position.
AnsweredCallCount	long	Read	Specifies the number of calls for the indicated service group answered by this agent.
CallbackRejectedCount	long	Read	Specifies the number of callbacks for the indicated service group rejected by this agent.
CallbackTimeoutCount	long	Read	Specifies the number of callback attempts for the indicated service group which timed out at this agent before being acknowledged.
CurrentChoice	long	Read	Specifies the current skill choice for this service group.
DirectCallTimeoutCount	long	Read	Specifies the number of direct calls (not overflowed) to the service group which timed out at this agent before being acknowledged.
DirectRejectedCount	long	Read	Specifies the number of direct calls (not overflowed) to the service group which were rejected by this agent.
OfferedCallCount	long	Read	Specifies the total number of calls offered to this agent for the indicated service group.
TimeStamp	DATE	Read	Specifies the time that performance data was last updated.

ISERVICEGROUPLIST

This object contains a list of all defined service group Ids.

Table 13:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
ServiceGroupCount	long	Read	Specifies the number of service group IDs contained in the list.

Table 14:

METHOD	PURPOSE
METHODS SUPPORTED PURPOSE	
GetServiceGroupID(long lIndex, long* pGroupID)	Retrieves the unique service group identifier at a particular index Parameters: lIndex - Specifies the index for which to retrieve group information. The valid range is from 1 to ServiceGroupCount pGroupID - Contains the unique ID of the service group at the specified index position
GetServiceGroupName(long lIndex, BSTR * pGroupName)	Retrieves the service group name at a particular index Parameters: lIndex - Specifies the index for which to retrieve group information. The valid range is from 1 to ServiceGroupCount pGroupName - Contains the name of the service group at the specified index position
GetServiceGroupSiteID(long lIndex, BSTR * pGroupSiteID)	Retrieves the service group site at a particular index Parameters: lIndex - Specifies the index for which to retrieve group information. The valid range is from 1 to ServiceGroupCount pGroupSiteID - Contains the site ID the service group at the specified index position

IAGENTLIST

This object contains a list of all defined agent Ids.

Table 15:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
AgentCount	long	Read	Specifies the number of agent Ids contained in the list.

Table 16:

METHOD	PURPOSE
METHODS SUPPORTED	
GetAgentID(long lIndex, long* pAgentID)	Retrieves the unique agent identifier at a particular index Parameters: lIndex - Specifies the index for which to retrieve agent information. The valid range is from 1 to AgentCount pAgentID - Contains the unique ID of the agent at the specified index position
GetAgentName(long lIndex, BSTR * pAgentName)	Retrieves the agent name at a particular index Parameters: lIndex - Specifies the index for which to retrieve agent information. The valid range is from 1 to AgentCount pAgentName - Contains the name of the agent at the specified index position
GetAgentSiteID(long lIndex, BSTR * pAgentSiteID)	Retrieves the agent site at a particular index Parameters: lIndex - Specifies the index for which to retrieve agent information. The valid range is from 1 to AgentCount pAgentSiteID - Contains the site ID of the agent at the specified index position

ISERVICEACCESSLIST

Table 17:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
ServiceAccessCount	long	Read	Specifies the number of service access IDs contained in the list.

Table 18:

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceAccessID(long lIndex, long* pServiceAccessID)	Retrieves the unique service access identifier at a particular index Parameters: lIndex - Specifies the index for which to retrieve service access information. The valid range is from 1 to ServiceAccessCount pServiceAccessID - Contains the unique ID of the service access at the specified index position

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceAccessName(long lIndex, BSTR * pServiceAccessName)	Retrieves the service access name at a particular index Parameters: lIndex - Specifies the index for which to retrieve service access information. The valid range is from 1 to ServiceAccessCount pServiceAccessName - Contains the name of the service access at the specified index position
GetServiceAccessSiteID(long lIndex, BSTR * pServiceAccessSiteID)	Retrieves the service access site at a particular index Parameters: lIndex - Specifies the index for which to retrieve service access information. The valid range is from 1 to ServiceAccessCount pServiceAccessSiteID - Contains the site ID the service access at the specified index position

ISERVICEACCESSPERFORMANCE

Table 19:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
CurrentCallCount	long	Read	Specifies the number of calls currently being handled for this service access.
OfferedDirectCallCount	long	Read	Specifies the number of direct calls offered for this service access.
OverflowedInCallCount	long	Read	Specifies the number of calls overflowed into this service access.
OverflowedOutCallCount	long	Read	Specifies the number of calls overflowed out of this service access.
AbandonedCallCount	long	Read	Specifies the number of calls abandoned from this service access.
AvgHandlingTime	long	Read	Specifies the average handling time in seconds for this service access.
SiteID	BSTR	Read	Specifies the site this service access belongs to.
ServiceAccessID	long	Read	Specifies the unique service access identifier for this service access.
ServiceAccessName	BSTR	Read	Specifies the service access name.
TimeStamp	DATE	Read	Specifies the time that performance data was last updated for this service group.

ISITEIDLIST

Table 20:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
SiteIDCount	long	Read	Specifies the number site Ids contained in the list.

Table 21:

METHODS SUPPORTED

METHOD	PURPOSE
GetSiteID(long lIndex, BSTR * pSiteID)	Retrieves the site ID at a particular index Parameters: lIndex - Specifies the index for which to retrieve site information. The valid range is from 1 to SiteIDCount pSiteID - Contains the site ID at the specified index position

ISERVICEGROUPPERFORMANCELIST

Table 22:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
ServiceGroup PerformanceList	long	Read	Specifies the number of IServiceGroupPerformance objects contained in the list.

Table 23:

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceGroupPerformance(l ong Index, IServiceGroupPerformance*pp GroupInfo)	Retrieves the IServiceGroupPerformance object corresponding to the passed index. Parameters: Index - Specifies the index for which to retrieve performance information. The valid range is from 1 to ServiceGroupPerformanceCount. ppGroupInfo - Pointer to an IServiceGroupPerformance object containing service group information.

IAGENTPERFORMANCELIST

Table 24:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
AgentPerformanceCount	long	Read	Specifies the number of IAgentPerformance objects available.

Table 25:

METHODS SUPPORTED

METHOD	PURPOSE
GetAgentPerformance(long lAgentIndex, IAgentPerformance** ppAgentPerformance)	Retrieves an IAgentPerformance object for the specified agent index. Parameters: lAgentIndex - Specifies the index for which to retrieve agent information. The valid range is from 1 to AgentPerformanceCount. ppAgentPerformance - Contains a pointer to an IAgentPerformance object if the method is successful.

ISERVICEACCESSPERFORMANCELIST

Table 26:

PROPERTIES

PROPERTY	TYPE	ACCESS	DESCRIPTION
ServiceAccessPerformanceCount	long	Read	Specifies the number of IServiceAccessPerformance objects available.

Table 27:

METHODS SUPPORTED

METHOD	PURPOSE
GetServiceAccessPerformance (long lServiceAccessIndex, IServiceAccessPerformance** ppServiceAccessPerformance)	Retrieves an IServiceAccessPerformance object for the specified service access index. Parameters: lServiceAccessIndex - Specifies the index for which to retrieve service access information. The valid range is from 1 to ServiceAccessPerformanceCount. ppServiceAccessPerformance - Contains a pointer to an IServiceAccessPerformance object if the method is successful.

ERROR LOG

The Real Time Interface Service will log critical errors to the Windows Event Log, to be viewed through the Event Viewer. Additionally, an error log file is maintained with warnings and errors. The log is named NetMonError.log, and is located at <InstallDir>\Services\Bin\Log.

DATABASE DESCRIPTION

Following are the database tables created by the Real Time Interface Service in the MiCC Enterprise SQL Server database when started up, if they do not exist.

NET_MONITOR_DATA

The net_monitor_data table contains real-time performance information for each service group currently being monitored.

Note: Groups at the local site, as well as at remote sites, are in the table.

Table 28:

FIELD	TYPE	DESCRIPTION
site_id	VARCHAR (15)	Unique site identifier for the MiCC Enterprise site on which this service group is being monitored
service_grp_id	INT	Unique identifier for the service group
service_grp_name	VARCHAR (20)	Name of the service group
calls_in_que	INT	Current number of calls in queue for this service group
free_agents	INT	Current number of agents free to accept a call for this service group
busy_agents	INT	Current number of agents currently handling a call for this service group
busy_other_agents	INT	Current number of agents skilled to handle calls for this service group, but currently handling a call for another service group
logged_on_agents	INT	Number of logged on agents skilled to handle calls for this service group
unavailable_agents	INT	Number of agents skilled to handle calls for this service group, but currently in Not Ready status
current_skill_choice	INT	Current skill choice configured for the service group

FIELD	TYPE	DESCRIPTION
first_call_time	DATETIME	Time (in local time) at which the first call was placed into the queue for this service group
longest_wait_time	INT	Longest wait time (in seconds) for the calls queued for this service group
service_level	INT	Current service level percentage for this service group
abandoned_rate	INT	Current abandoned rate percentage for this service group
est_wait_time_1	INT	Estimated wait time in seconds for Skill Choice 1 for this service group
est_wait_ti me_2	INT	Estimated wait time in seconds for Skill Choice 2 for this service group
est_wait_time_3	INT	Estimated wait time in seconds for Skill Choice 3 for this service group
answered_cnt	INT	Current answered call count for this service group
abandoned_cnt	INT	Current abandoned call count for this service group
overflowed_out_cnt	INT	Current overflowed out call count for this service group
time_stamp	DATETIME	Time (in local time) at which this record was added or last updated
callback_cnt	INT	Number of callbacks processed for this service group.
callback_fail_cnt	INT	Number of callbacks that have failed for this service group.
callbacks_in_que	INT	Current number of callbacks in the queue for this service group.
scheduled_callback s	INT	Current number of callbacks scheduled.

NET_MONITOR_GROUPS

The net_monitor_groups table contains the list of service groups which should be monitored by the local Real Time Interface Service.

Note: If **Monitor All Groups** is selected from the Real Time Interface Service Configuration, this table is not used by the Real Time Interface Service.

Table 29:

FIELD	TYPE	DESCRIPTION

service_grp_id	INT	Unique identifier for the service group currently being monitored by the local Real Time Interface Service.

NET_MONITOR_AGENTS

The net_monitor_agents table contains the list of agents, which should be monitored by the local Real Time Interface Service.

Note: If **Monitor All Agents** is selected from the Real Time Interface Service Configuration, this table is not used by the Real Time Interface Service.

Table 30:

FIELD	TYPE	DESCRIPTION
agent_id	INT	Unique identifier for the agent currently being monitored by the local Real Time Interface Service.

NET_MONITOR_SVCACCESSES

The net_monitor_svcaccesses table contains the list of service accesses, which should be monitored by the local Real Time Interface Service.

Note: If **Monitor All Agents** is selected from the Real Time Interface Service Configuration, this table is not used by the Real Time Interface Service.

Table 31:

FIELD	TYPE	DESCRIPTION
svcaccess_id	INT	Unique identifier for the service access currently being monitored by the local Real Time Interface Service.

REGISTRY SETTINGS

Following are the registry entries used by the Real Time Interface Service. All entries are located under the key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CCNetworkMonitor\Parameters.

ENTRY	TYPE	DEFAULT	DESCRIPTION
BrokerServerNode	REG_SZ	Local machine name	Machine name of the MiCC Enterprise Broker Service.
BrokerServerPort	DWORD	2600	TCP/IP port of the MiCC Enterprise Broker Service.
DisableCOMInterface	DWORD	0	If set to 1, the COM interface will not be enabled, so clients connecting via the COM interface will not be supported.
LocalPort	DWORD	7500	TCP/IP port of the local Real Time Interface Service.
LocalSiteID	REG_SZ	Local machine name	Unique identifier of the local Real Time Interface Service. Configured through Real Time Interface Service Configuration.
LocalUpdateInterval	DWORD	5000	Interval in milliseconds for updating the Real Time Interface Service Database. Configured through Real Time Interface Service Configuration.
MasterServerNode	REG_SZ	None	Machine name of master Real Time Interface Service. If not set, current Real Time Interface Service is set as master.
MasterServerPort	DWORD	7500	TCP/IP port of master Real Time Interface Service.
MonitorAll Agents	DWORD	1	1 if all agents should be monitored. 0 otherwise. Configured through Real Time Interface Service Configuration.
MonitorAll Groups	DWORD	1	1 if all service groups should be monitored. 0 otherwise. Configured through Real Time Interface Service Configuration.
MonitorAllSvcAccesses	DWORD	1	1 if all service accesses should be monitored. 0 otherwise. Configured through Real Time Interface Service Configuration.
PreferredEventService	REG_SZ	None	If specified, contains the machine name of the preferred MiCC Enterprise Event Service. Otherwise, the name is retrieved from Broker Service.

ENTRY	TYPE	DEFAULT	DESCRIPTION
PreferredEventServicePort	DWORD	2605	If specified, contains the TCP/IP port of the preferred MiCC Enterprise Event Service. Otherwise, the port is retrieved from Broker Service.
RemoteReconnectInterval	DWORD	30000	Interval in milliseconds between connection attempts to the master Real Time Interface Service. Configured through Real Time Interface Service Configuration.
RemoteUpdateInterval	DWORD	5000	Interval in milliseconds between data updates to the master Real Time Interface Service. Configured through Real Time Interface Service Configuration.
ServiceGroupRealTime	DWORD	0	1 if Service Group Real Time Database Update is enabled. 0 otherwise. Configured through Real Time Interface Service Configuration
ServiceReconnectInterval	DWORD	30000	Interval in milliseconds between connection attempts to the local MiCC Enterprise services.
LogLevel	DWORD	0	Level or reporting information stored in the NetMonError.log file. Valid range is 0 to 4 where 0 indicates no logging and 4 indicates the highest level of logging.
AllowAgentDetailMonitoring	DWORD	0	Specified if detailed agent status informatoin is sent over th einterface or not. 0 = No 1 = Yes Configured via SecCfg.exe