IP Communications Applications for Business Advantage

Metreos© Communications Environment 2.1
Call Monitor Reporting

The Call Monitor application records every instance of usage to a database for the purpose of record keeping. In the case that usage records can not be stored in the database, an XML file is used as a temporary alternative. If at anytime the database becomes available again, the contents of the XML file are written to the database, and the usage redundancy file is then deleted.

A command-line utility is provided in order to retrieve the usage records from the database in a human-readable or comma separated format.

Using the Reporting Tool

Packaged with the Application Server is a simple command line tool which can be used to generate HTML and CSV files.

The tool must be run on the Application Server, and can be found at C:\Metreos\PCapService\CallMonitorReport.exe.

CallMonitorReport.exe Usage

The executable takes up to two arguments:

- -s: Report Start Date
- -e: Report End Date

If both are provided, then the range of time between the date specified by –s and the date specified by –e will be used in querying the database for records.

If only –e is provided, then the reporting tool will report the earliest entry through the date specified by –e.

If only –s is provided, then the reporting tool will report all entries from the date specified by –s through the most current date in the database.

If neither is provided, then the tool will display help information.

The valid formats for specifying a date:

- yyyy-mm-dd
- dd-mon-yy
- mm/dd/yyyy

CallMonitorReport.exe Behavior

Once the date range has been determined, the tool will query the database for all usage records in that range, and generate an HTML and CSV file in the directory C:\monitor_call\AuditLog\.

The files will be of the form [StartDate]_[EndDate].csv and [StartDate]_[EndDate].htm, where [StartDate and EndDate] are of the form yyyy-mm-dd.

Usage Reporting Database Schema

The usage reporting database consists of one table. Each row in the following table corresponds to a field of the usage reporting table.

Database Name: monitor_call
Table Name: monitored_call

Field	Type	Null	Key	Default	Extra
mc_monitored_call_id	Int(10) unsigned		PRI	NULL	Auto Increment
mc_government_agent_number	Varchar(255)				
mc_did_number	Varchar(255)				
mc_insurance_agent_number	Varchar(255)				
mc_customer_number	Varchar(255)				
mc_monitored_sid	Varchar(255)				
mc_start_monitor_timestamp	Timestamp	YES		CURRENT_TIMESTAMP	

Table 1: Table Definition

Field	Description
mc_monitored_call_id	Primary key of the table. Auto incremented upon insertion of a new record.
mc_government_agent_number	The DN of the government agent calling in to monitor a conversation.
mc_did_number	The DID DN used to access the Call Monitor application.
mc_insurance_agent_number	The DN of the insurance agent randomly chosen to be monitored.
mc_customer_number	The DN of the customer that the insurance agent is speaking with.
mc_monitored_sid	The device name, or SID, of the insurance agent's SCCP device.
mc_start_monitor_timestamp	The time that the government agent begins monitoring.

Table 2: Table Description

The system comes preloaded with the database and table already in place. By default, the Call Monitor application writes usage records to this database.

XML Schema

The redundancy usage XML file is a one-to-one mapping of the calls_monitored database table.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="CallRecordTable" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="CallRecordTable" msdata:IsDataSet="true">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Records">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Record" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="governmentAgentNumber" type="xs:string" />
                  <xs:attribute name="did" type="xs:string" />
                  <xs:attribute name="insuranceAgentNumber" type="xs:string" />
                  <xs:attribute name="customerNumber" type="xs:string" />
                  <xs:attribute name="monitoredSid" type="xs:string" />
                  <xs:attribute name="startMonitorTime" type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Code Listing 1: XML Schema of the application usage redundancy file

```
<?xml version="1.0"?>
<CallRecordTable xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Records>
    <Record governmentAgentNumber="5126872021" did="6213" insuranceAgentNumber="4742"
customerNumber="9892132342" monitoredSid="SEP000011112222" startMonitorTime="2005-09-08
22.27.43:756" />
    <Record governmentAgentNumber="5126872035" did="6217" insuranceAgentNumber="4742"
customerNumber="9892152314" monitoredSid="SEP000011112222" startMonitorTime="2005-09-08
22.28.28:389" />
  </Records>
</CallRecordTable>
```

Code Listing 2: XML example of the application usage redundancy file

The XML redundancy file is written to in the event that the database can not be used to keep usage records. The system will automatically maintain this file. This file can be found at C:\monitor_call\backup\pendingrecords.xml. However, the file will only exist in the case that there is currently an issue in writing records to the database; in the case that the database comes back online, the current contents of the file are inserted into the database, and the file is deleted.

If for some reason neither the file nor the database can be written to, the application will log error messages into the logging system of the Application Server.