TEMS Automatic Trace Functionality

White Paper

ERICSSON

Contents

1 Trace Functionality

The Trace function has the effect of logging some activities of various TEMS Automatic application components. These logs are useful for fault tracing, and you may be asked to use the Trace function when contacting customer support. You should be aware that the Trace function also lowers performance; therefore, use it only when necessary.

For the Trace function to be available, it must have been initialized. This is always the case if the file "<application name>Trace.ini" is present in the same directory as the application executable. These are the precise names of the .ini files:

- CallGeneratorTrace.ini
- COMServerTrace.ini
- LogfileHandlerTrace.ini

1.1 Trace INI File Structure

The Trace .ini file consists of several sections. The first section is composed as follows (the application in this example is the Call Generator):

```
[CallGenerator]
outputtype=file
output=c:\temp\CallGenerator.log
maxentries=50000
fullaction=delete
backupextension=".BUP"
execute=notepad.exe
timeformat=""
```

The keys specified are:

- **outputtype:** By default set to “file”; no other options.
  - Specifies that all traces shall be written to file.
- **output:** Set by TEMS Automatic system administrator.
  - Specifies the path to the trace log file (e.g. “c:\temp\CallGeneratorTrace.log”).
- **maxentries:** By default set to 50,000 lines; editable by TEMS Automatic system administrator.
  - Specifies the maximum number of lines in the trace log file (see **fullaction**).

- **fullaction:** By default set to “delete”; editable by TEMS Automatic system administrator to “backup”.
  - Specifies the action taken when the trace log file has reached the size **maxentries**. If **fullaction** = “delete”, the trace log file is truncated at this point; if **fullaction** = “backup”, the trace is continued to a new log file, and the extension of the old one is changed to **backupextension** (see below).
- **backupextension:** By default set to “.BUP”; no other options.
  - Specifies the backup trace log file extension (see **fullaction**).
- **execute:** By default set to “notepad.exe”; no other options.
- **timeformat:** By default set to “” (nothing); no other options.

To change any of the above parameters, a restart of the application is needed. Changing parameter values at runtime has no effect.

The Trace .ini file continues with the following sections, governing which components are traced for the application in question. See the example below for the Call Generator.

```
[CallGen Main]
#tracelevel=MESSAGE

[CallGen Channel]
#tracelevel=MESSAGE

[CallGen Scheduler]
#tracelevel=MESSAGE

[CallGen PSQM]
#tracelevel=MESSAGE

[CallGen DSP]
#tracelevel=MESSAGE
```

The **tracelevel** key describes which trace messages shall be written to the trace log file. It can take the values MESSAGE, WARNING or FATAL. All types of messages with *equal or higher* severity are written to the trace log file (FATAL = highest severity). You can edit the **tracelevel** sections to change the tracing behavior at runtime.

NOTE: By default, no tracing is activated (all **tracelevel** lines are commented out with a #). To start tracing for a component, uncomment the corresponding **tracelevel** line and save the file.

1.2 TEMS Automatic Application Components

Here are listed all components that generate trace messages for each TEMS Automatic application.

Call Generator	Communication Server	Logfile Handler
[CallGen Main] *tracelevel*=MESSAGE	[ComServer] *tracelevel*=MESSAGE	[DBWRITER] *tracelevel*=MESSAGE
[CallGen Channel] *tracelevel*=MESSAGE	[FTPCLIENT] *tracelevel*=WARNING	[DECODER] *tracelevel*= MESSAGE
[CallGen Scheduler] *tracelevel*=MESSAGE	[FTPMirror] *tracelevel*=MESSAGE	[FRONTEND] *tracelevel*= MESSAGE
[CallGen PSQM] *tracelevel*=MESSAGE	[FileWriter] *tracelevel*=MESSAGE	[EVENTHANDLER] *tracelevel*= MESSAGE
[CallGen DSP] *tracelevel*=MESSAGE	[FileReader] *tracelevel*=MESSAGE	

2 MTU Trace Functionality

The cables used for trace and configuration of the MTUs are of standard type null modem cable. The cable has 9 pin D-sub female connectors on both ends. The pin configuration of the cable is shown on page 11.

MTU 600: The connectors are behind the cover on the right side of the MTU. **COM A is the left connector and is used for tracing.**

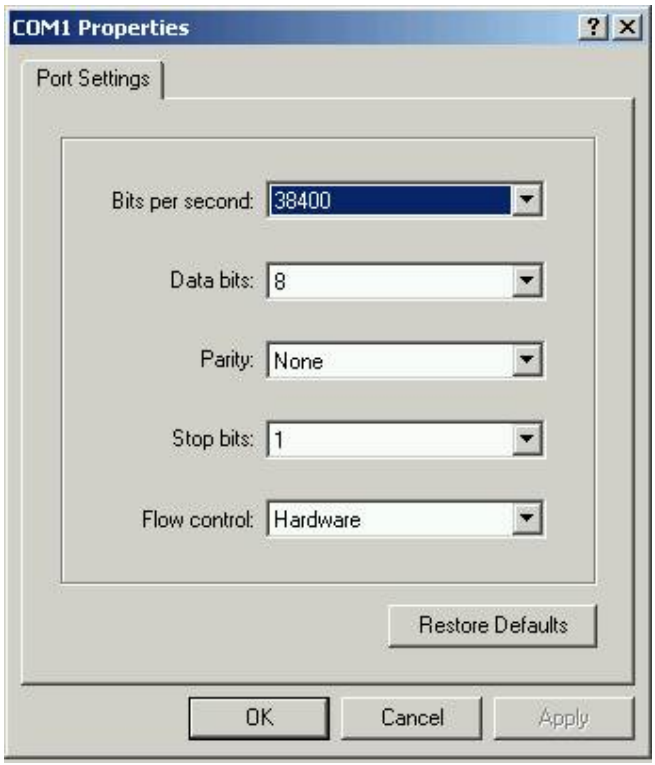
MTU 700: The connectors are found behind the cover in the middle on the front of the MTU. **COM 1 is the bottom connector and is used for tracing.**

MTU 750: The connectors are found behind the cover on the right side of the MTU. **COM 1 is the bottom connector and is used for tracing.**

To trace the MTU the cable is connected between the correct COM-port on the MTU and a serial communication port on the PC. Use the Hyper terminal application to see the trace. The hyper terminal can be found in programs, accessories, and communications.

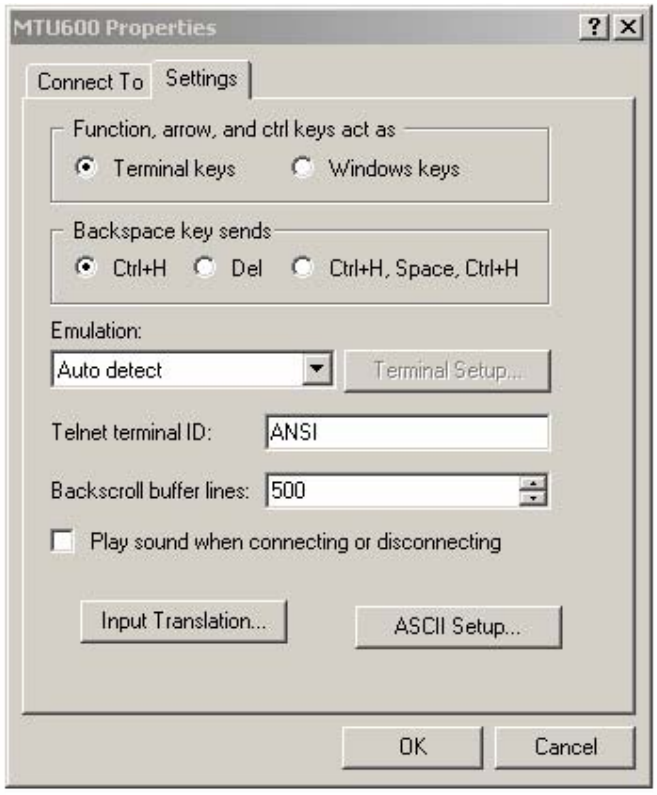
Name your connection. Click OK

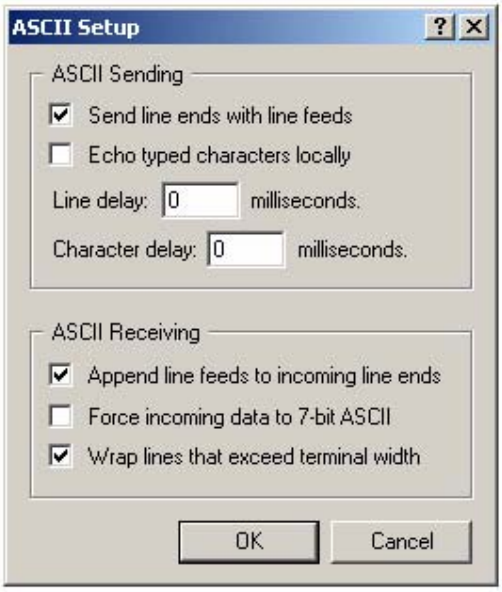
Fill in country code and area code and which COM port you are using. Click OK.



Fill in port settings as shown in the table below. Click OK.

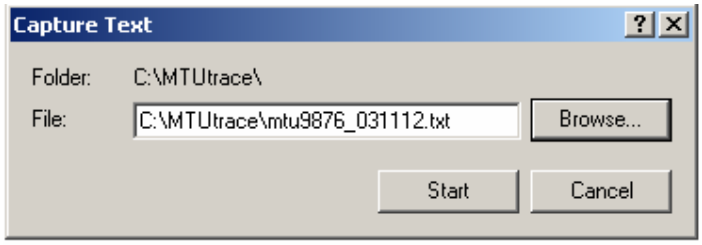
MTU type	**MTU600**	**MTU700**	**MTU700**	**MTU750**
SW version	All versions	Up to 4.0.x.x	After 4.1.x.x	All versions
Bits per sec	9600	9600	38400	38400





Once you have the terminal window click File and Properties. In the properties dialogue click ASCII setup. Fill in as shown below click OK to come back to the terminal window.

The trace log can be saved. This is achieved by going to the transfer menu in the terminal window and choosing Capture text. Click on browse to choose where the text file is to be saved. Name the file and click Start.



Null Modem Pin Connetion:

	25 Pin	9 Pin		9 Pin	25 Pin	
FG (Frame Ground)	1	-	X	-	1	FG
TD (Transmit Data)	2	3	-	2	3	RD
RD (Receive Data)	3	2	-	3	2	TD
RTS (Request To Send)	4	7	-	8	5	CTS
CTS (Clear To Send)	5	8	-	7	4	RTS
SG (Signal Ground)	7	5	-	5	7	SG
DSR (Data Set Ready)	6	6	-	4	20	DTR
DTR (Data Terminal Ready)	20	4	-	6	6	DSR