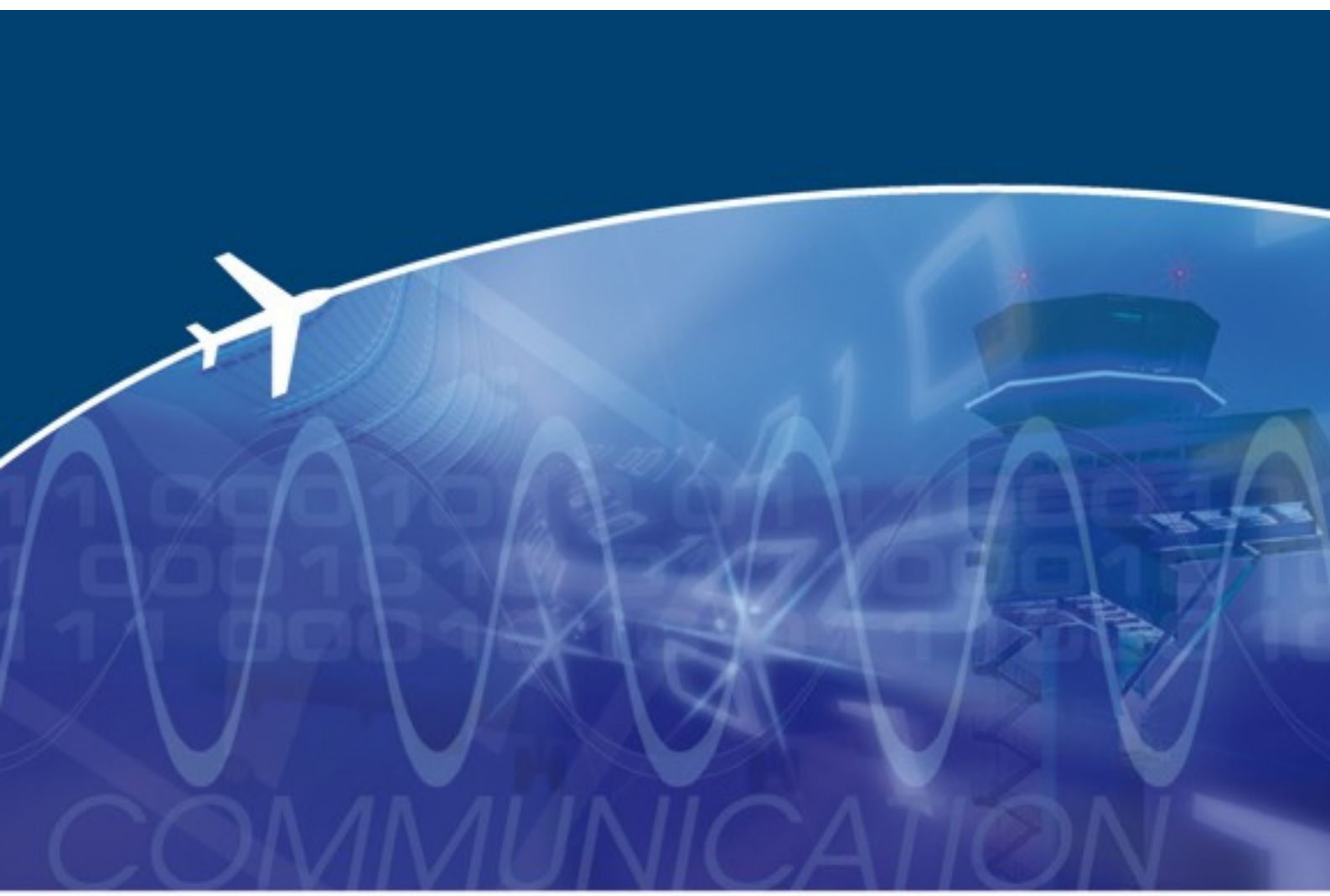


Communications Systems
and Programmes

ETIC User Guide

COM Reference No.:	**NETWORK-ETIC-UG-01**
CDMS Reference No.:	
Edition Number:	**3.2.2**
Edition Date:	**28/10/2008**
Status:	**Issued**
Classification:	**Restricted**
Author:	**Eivan Cerasi**
Intended For:	**Stakeholders**

EUROCONTROL

Copyright Notice

Document Approval

Name	Comments	Signature
Eivan Cerasi	Work Package Leader	17/10/08
Jacky Pouzet	Communications Domain Manager	21/10/08.
Robert Stewart	Head of DAP/CSP Communications Systems & Programmes	24/10/08
Guido Kerkhofs	Director ATM Programmes	27/10/08

Edition History

Edition No.	Date	Comments	section/pages affected
1.0 to 3.0	06/93 to 06/05	Releases of ETIC version 1.0 to 3.0	All
3.0.1	07/05	Internal release of ETIC version 3.0.1	7.7, 8.2, 8.5, 8.8, 8.10, 8.11, 8.12, 8.14
3.0.2	03/06	Release of ETIC version 3.0.2	5.3, 6, 8.9.5, 8.16, 7.3.2.3, 9.2, 9.3, 10.1, 13.1
3.1.0	02/07	Release of ETIC version 3.1.0	3.1, 5.3.3, 6, 7.6, 8.6.3, 8.9, 8.14.2, 10, 11
3.2.0	11/07	Release of ETIC version 3.2.0	3, 5.3, 6, 7.7, 8.2, Fig 9, 8.6.6, 8.7, 8.9, Fig 17, Fig 25-26, 8.13, 8.14, Fig 35, 10, 11.3, 12.1
3.2.1	05/08	Release of ETIC version 3.2.1	3.1, 5.2, 5.3, 6, 7.4.2, 7.5, 7.7, 8.6.3, Fig 9, Fig 15, Fig 19, 8.10.1, 8.10.5, 8.11, Fig 26, 8.12.6, 8.13, Fig 37-38, 11, 12.1
3.2.2	10/08	Release of ETIC version 3.2.2	5.3, 6, 7.3, 7.4.3, 7.5, 7.7, Fig 9, 8.6.2, 8.10.1, 8.12.2, 8.12.3, 8.12.5, Fig 37, 8.17, 9.2, 9.3, 12.1

Distribution List

Name	Representing
All ETIC users	National Administrations, ANSPs and Industry Stakeholders

Status, Audience and Accessibility

Status		Intended for		Accessible via	
Working Draft	☐	EATM	☐	Project Server	☑
Proposed Issue	☐	Stakeholder	☑	One Sky Team	☑
Issued	☑	General Public	☐		

Contact Point

Contact Person(s)	Telephone	Unit
Eivan Cerasi	+32 2 729 3791	DAP/CSP

Table of Contents

List of Tables

List of Figures

1 Purpose

This document is the ETIC version 3.2 User Guide. It provides a general description of ETIC, installation procedures and its functions. It also provides a description of the internal ETIC behaviour and details user interaction.

The purpose of ETIC is to provide a user-friendly cross-platform software tool to test implementations of the Flight Message Transfer Protocol (FMTP) over TCP/IP and the Flight Data Exchange Interface Control Document Part 1 (FDE ICD Part 1) over X.25.

2 Document Audience

This User Guide is intended to support developers, system engineers and system testers that are involved in the implementation of a FMTP/FDE ICD Part 1 or their validation. As a result, knowledge of the FMTP and FDE ICD Part 1 protocols is assumed.

3 Documents

3.1 Reference documents

It is advised to consult the EUROCONTROL web server http://www.eurocontrol.int/communications for the most current information on ETIC and the associated documentation.

[RDn]	Reference / Edition	Title
[RD1]	(FDE ICD Part 1), Edition 1.0	EUROCONTROL Standard Document for Flight Data Exchange Interface Control Document Part 1
[RD2]	(FMTP Guidelines), Edition 1.0	EUROCONTROL Guidelines for Implementation Support, Part 5, Chapter 13: Flight Message Transfer Protocol
[RD3]	(FMTP Specification), Edition 2.0	EUROCONTROL SPECIFICATION of Interoperability and Performance Requirements for the Flight Message Transfer Protocol (FMTP), EUROCONTROL-SPEC-0100, 14/06/07
[RD4]	(OLDI) Edition 4.1	EUROCONTROL SPECIFICATION for On-Line Data Interchange (OLDI), EUROCONTROL-SPEC-0106, 16/01/08
[RD5]	(ICAO Document 4444), 14th Edition	ICAO Procedures for Air Traffic Services Doc 4444, ATM/501, APPENDIX 3. AIR TRAFFIC SERVICES MESSAGES
[RD6]	(ADEXP), Edition 3.0	EUROCONTROL SPECIFICATION for ATS Data Exchange Presentation (ADEXP), EUROCONTROL-SPEC-0107, 20/10/07

Table 1: Reference documents

The FMTP Guideline (Reference [RD2]) is referred by EUROCONTROL ECIP Objective ITY-FMTP. The European Union has published the FMTP Implementing Rule for the Single European Sky as Commission Regulation (EC) No. 633/2007. The EUROCONTROL Specification for FMTP (Reference [RD3]) is recognised as a Community Specification to support the Implementing Rule. There is full equivalence between the all mandatory technical provisions of the FMTP Specification and the FMTP Guideline (Reference [RD2]).

4 Abbreviations

The following abbreviations are used in this User Guide:

Abbreviation	Signification
ADEXP	ATS Data Exchange Presentation
CP	Communication Partner
ECIP	European Convergence and Implementation Plan
FDE ICD Part 1	Flight Data Exchange Interface Control Document Part 1
FMTP	Flight Message Transfer Protocol
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ICAO	International Civil Aviation Organisation
IP	Internet Protocol
OLDI	On-Line Data Interchange
NSAP	Network Service Access Point
JRE	Java Runtime Environment
JVM	Java Virtual Machine
SuT	System under Test
TCP	Transmission Control Protocol
NSAP	Network Service Access Point
XML	Extensible Markup Language

Table 2: Abbreviations

5 ETIC Presentation

5.1 Background

EUROCONTROL previously released version 2.1 of the ETIC test tool. It supported the FDE ICD Part 1 standard (Reference [RD1]) for the exchange of flight data between air traffic control centres. ETIC had the capability to simulate one or more X.25 peer implementations, known as communication partners (CPs). The System under Test (SuT) interfaced to ETIC CPs which simulated remote FDE ICD Part 1 implementations. With ETIC, the connection status could be monitored and it was also possible to force the CPs to operate abnormally thereby verifying SuT compliance.

With the advent of the TCP/IP protocol suite, EUROCONTROL has prepared a new version of the flight data exchange specification to operate over TCP/IP and named it Flight Message Transfer Protocol (Reference [RD2]).

To support this new specification, a new version of ETIC (version 3) has been developed. While having the same functionality as the previous version, the new version includes support for FMTP operating over IPv6 or IPv4. The new version is also a stand-alone application but can operate on several operating systems. User interaction is through a

graphical user interface (GUI) and a key feature is its user friendliness. Cross-platform independence has been made possible by developing in the Java programming language. Nevertheless, the FDE ICD Part 1 module depends on non-portable C libraries that interface to X.25 network cards.

5.2 New features of ETIC

This section is primarily intended for existing ETIC version 2.1 (or older) users. ETIC version 3 is not an evolution of ETIC version 2.1 but a completely rewritten product.

Its main features are:

1. It supports the following protocols:
    a. FMTP over TCP/IPv4,
    b. FMTP over TCP/IPv6,
    c. FDE ICD Part 1 over X.25.

2. The ETIC implementation of FMTP is developed on the Java® platform (http://java.sun.com) of Sun Microsystems and uses de-facto standard and cross-platform open-source libraries. Therefore, it can run on any platform on which a Java Virtual Machine (JVM) 5.0 has been installed.

3. The ETIC implementation of FDE ICD Part 1 requires a DIALOGIC® Eiconcard™ http://www.dialogic.com and their associated non-portable C libraries. Currently, FDE ICD Part 1 support is limited to Microsoft® Windows® (98, ME, NT, XP, 2000, 2003, Vista), SUSE®, Novell® Linux Desktop, CentOS and RED HAT® Linux® platforms.

4. It displays an intuitive GUI using the Java Swing technology. As a user selectable native look-and-feel could introduce cross platform concerns, the *metal* look-and-feel has been hard-coded.

5. Whilst conformant to both FMTP and FDE ICD Part 1 protocols, ETIC can also behave in a non-conformant way in order to check the behaviour of the SuT.

6. All the functional capabilities of the previous ETIC versions have been maintained but have been upgraded and configurable in user-friendly manner. In particular:
    a. The notion of the ini file and database is superseded by the combination of:
        i. System-wide parameters, and
        ii. Specific ETIC sessions.

    b. There is a no hard-coded limitation on the number of CPs that can be defined. This number is only limited by the performance and memory size of the platform. For instance, on a standard PC, ETIC version 3 has been tested with up to 256 CPs continuously exchanging messages at an aggregate rate of 60 per second.

    c. Three working modes are available:
        i. Normal session mode: all ETIC functions are available,
        ii. Locked session mode: restricted editing and deletion of session entities to distribute pre-defined test scenarios to operators or third parties, and
        iii. Conformance session mode: restricted to the execution of a semi-automated conformance test procedure to validate SuT compliance to the FMTP protocol.

d. Every event that occurs is logged to disk for future retrieval. Live monitoring is available from the main GUI. Both come with extended event filtering tools to display specific events.

e. “Lists” are replaced by “sequences”. Sequences should be understood as a means to simulate operational scenarios. In particular, sequences can:
   i. Execute actions such as connecting with the SuT, sending a message at a given time, etc.
   ii. Watch for events, such as connection establishment, message receipt, etc.
   iii. Be nested in other sequences,
   iv. Be iterated or even looped forever, and
   v. Run concurrently with other sequences.

f. Pattern matching using either wildcards or regular expressions is available for:
   i. CP identification,
   ii. Message content, and
   iii. Filtering.

g. A relay function that allows external non-interoperable FDE ICD and/or FMTP systems to establish associations and exchange messages between each other; which can be extended to support message scripting.

h. There is an operational On-Line Data Interchange (OLDI) message builder supporting ICAO (Reference [RD5]) and ADEXP (Reference [RD6]) message formats.

i. “Batch mode” allows the processing of commands from a batch file located in a user defined directory. Available commands include sending messages and writing information to the log file.

5.3 System requirements

5.3.1 Hardware

An x86 platform with the following characteristics:

- 1GHz CPU
- 256MB RAM
- Network interface card
- 20G Hard Disk
- Optional EICON X.25 Eiconcard C/S-series for FDE ICD Part 1 support.

ETIC also operates on alternative platforms as long as Java 1.5 is available e.g. Sun SPARC platforms.

5.3.2 Reported compatible hardware platforms

The following list is purely for informational purposes. ETIC is known to operate on this hardware but this list is by no means a recommended hardware list. Consult the EUROCONTROL OneSkyTeam (https://extranet.eurocontrol.int) devoted to FMTP/ETIC to discover other compatible configurations.

This ETIC version has been tested for compatibility on the following CPU platforms running Windows XP SP2:

- Fujitsu-Siemens Lifebook S6410 (Intel® Core™2 Duo 2.2GHz/2GB)
- HP xw6000 (Dual-Core Intel® Xeon® 2.8GHz/2GB) with Eiconcard C90, C91, C91v2, S91v2
- HP xw4000 (Intel Pentium® 4 2.66GHz/2GB) with Eiconcard C90, C91, C91v2, S91v2
- HP xw4000 (Intel Pentium® 4 2.66GHz/512MB) with Eiconcard C90, C91, C91v2, S91v2
- Toshiba Tecra® S1 (Mobile Intel® Pentium® 1.4GHz/512MB)
- Compaq Evo N800w (Mobile Intel® Pentium® 4 2.2Ghz/1GB)
- Clone PC (AMD Athlon™ 1.2GHz/512MB)

This ETIC version has been tested for compatibility on the following CPU platforms running Solaris 9 and 10:

- HP xw6000 (Dual-Core Intel® Xeon® 2.8GHz/2GB)
- Compaq Evo N800w (Mobile Intel Pentium® 4 2.2Ghz/1GB)
- Sun Blade 1000 (Ultra SPARC III – 64-bit)

This ETIC version has been tested for compatibility on the following CPU platforms running SUSE Linux 8.2, 9.x, 10:

- HP xw6000 (Dual-Core Intel® Xeon® 2.8GHz/2GB) with Eiconcard C90, C91, C91v2, S91v2
- HP xw4000 (Intel Pentium® 4 2.66GHz/256MB) with Eiconcard C90, C91, C91v2, S91v2
- Compaq Evo N800w (Mobile Intel Pentium® 4 2.2Ghz/1GB)

This ETIC version has been tested for compatibility on the following CPU platforms running RED HAT Linux 9:

- HP xw6000 (Dual-Core Intel® Xeon® 2.8GHz/2GB) with Eiconcard C90, C91, C91v2, S91v2
- HP xw4000 (Intel Pentium® 4 2.66GHz/256MB) with Eiconcard C90, C91, C91v2, S91v2
- Compaq Evo N800w (Mobile Intel Pentium® 4 2.2Ghz/1GB)

This ETIC version has been tested for compatibility on the following CPU platforms running Novell Linux Desktop 9 and RED HAT Enterprise Linux Workstation 3.0:

- HP xw4000 (Intel Pentium® 4 2.66GHz/256MB) with Eiconcard C90, C91, C91v2, S91v2

This ETIC version has been tested for compatibility on the following CPU platforms running Mac OS 10.4.2+ (Tiger):

- Apple PowerBook G4 (PowerPC™ 1.0 Ghz/1GB)
- Apple Mac mini (Intel® Core™ Duo 1.66GHz/512MB)

This ETIC version has been tested for compatibility on the following CPU platforms running Mac OS 10.5 (Leopard):

- Apple Mac mini (Intel® Core™2 Duo 2.16GHz/2GB)

5.3.3 Operating systems

The appropriate choice of operating system highly depends on the required ETIC FMTP/FDE ICD Part 1 protocol stacks.

The following table gives an illustration of ETIC protocol stack support per operating system. Other operating systems may support the FMTP protocol stack once the correct java run-time environment is available.

Note that:

- EICON PCMCIA Eiconcards are only supported on Windows
- SUSE 9.x refers to SUSE 9.0, 9.1 and 9.3, SUSE LE to SUSE Enterprise Linux and SUSE E to SUSE Enterprise
- SUSE 9.x refers to SUSE 9.0, 9.1 and 9.3, SUSE LE to SUSE Enterprise Linux and SUSE E to SUSE Enterprise

OS ETIC Stack	Windows		SUSE 8.2 SUSE 9.2 Novell LD 9 SUSE E 9.0 SUSE LE	SUSE 9.x SUSE 10	RH 9 CentOS RHEL	Mac OS 10.4.2+ 10.5	Solaris 9/10 x86 and SPARC
	98, ME, NT, 2k, XP, 2003	Vista					
FMTP (IPv6)	No*	Not yet**	Yes	Yes	Yes	Yes	Yes
FMTP (IPv4)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FDE ICD Part 1	Yes	Yes	Yes	No	Yes	No	No

*: See section 6.1.1.1 of this User Guide.
**: IPv6 support for ETIC is expected with the release 1.7 of Java Runtime Environment (JRE) v1.7.

5.3.4 Java runtime environment

The ETIC CD-ROM contains ready-made installation routines. In particular for Windows and Linux, these installation routines can install both ETIC and the appropriate Java Runtime Environment (JRE). However, the ready made installation routines allow ETIC to be installed by itself, as long as a JRE 1.5 is already present on the target platform. In particular, Swing and NIO must be available.

6 Installation

Prior to the installation of the ETIC software, the platform must be properly configured. The following section gives the details of the preparation for each supported operating system.

The instructions for your target OS should be read in full prior to any actions.

6.1 ETIC platform preparation

6.1.1 Microsoft Windows (98 2nd Ed, ME, NT, 2000 SP3+, 2003, XP SP1+)

6.1.1.1 Enabling IPv6

Microsoft Windows 2000, XP and 2003 support IPv6 but ETIC requires the java NIO class which is not available for the current Java Runtime Environments (JRE) within an IPv6 environment. As a result, ETIC cannot make use of IPv6 when running Windows 2000, XP and 2003. SUN does not plan to support NIO for these Windows platforms for IPv6.

More details regarding NIO support can be found on the Java support web site at: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6230761

Note: If installation problems are encountered when using rarer versions of Windows that are not listed here (e.g. 'Windows Server 2003, Enterprise Edition') users are advised to consult the web sites of Microsoft, DIALOGIC and Sun.

6.1.1.2 Installing Java

- plan to make use of the ETIC installer for Windows with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java -version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp
- open the "Java" console from the "Control Panel" to ensure that J2SE 5.0 is the default JRE for the "Java Application Runtime Settings"

6.1.1.3 Installing the Eiconcard

Windows NT requires Eiconcard Connections for Windows NT V4R5, Windows 98 and Windows ME require Eiconcard Connections for Windows 98&ME. These drivers are no longer maintained by DIALOGIC but may be still part of the installation CD-ROM or available for download at http://www.dialogic.com/products/serial_protocol_adapters/eol.htm (e-mail etic@eurocontrol.int for further information). DIALOGIC provides different Eiconcard drivers for 32 or 64-bit operating systems.

- Fit the Eiconcard network interface card hardware into the computer
- Windows will automatically detect that new hardware has been fitted
- Insert the Eiconcard Universal Connections Suite CD
- Access the 'quick.pdf' file to read the 'readme' information in the 'quickstart' document (If necessary, access the 'Install.pdf' file to read further information in the 'Getting Started' document)
- Choose for Windows to search the Eiconcard Universal Connections Suite CD for drivers (Note, if Windows finds the same driver at two different locations on the CD, either of them can be chosen)
- Alternatively, download the latest Eiconcard Connections driver, appropriate for the type of Microsoft Windows being used, from the DIALOGIC website (e.g. http://www.dialogic.com/products/serial_protocol_adapters/Eiconcard_Connections_for_W2K.htm?dl=1&regID=5801)
- Install the chosen Eiconcard driver
- Click 'continue' on several pop-ups. (This is necessary because the Eiconcard driver is unsigned)
- Wait approximately 20 seconds for the installation to complete
- Click 'finish'
- Configure the Eiconcard using the Eiconcard driver settings described in section 6.2 of this User Guide

6.1.2 Microsoft Windows Vista

6.1.2.1 Enabling IPv6

Microsoft Windows Vista supports IPv6 but ETIC requires the java NIO class which is not available for the current Java Runtime Environment (JRE) within an IPv6 environment. As a result, ETIC does not currently support IPv6 when running Windows Vista.

However, in the course of 2009, SUN does plan to support NIO for Windows Vista for IPv6 with the release of JRE 1.7.

More details regarding NIO support can be found on the Java support web site at: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6230761

6.1.2.2 Installing Java

- plan to make use of the ETIC installer for Windows with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java -version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp
- open the "Java" console from the "Control Panel" to ensure that J2SE 5.0 is the default JRE for the "Java Application Runtime Settings"

6.1.2.3 Installing the Eiconcard

Windows Vista requires Eiconcard Connections for Windows V6R6. Refer to the installation procedure described in section 6.1.1.3. DIALOGIC provides different Eiconcard drivers for 32 and 64-bit operating systems.

6.1.3 Solaris 9 and 10

6.1.3.1 Enabling IPv6

- during OS installation, if you enabled IPv6 support a `/etc/hostname6.interface` file exists
- if IPv6 is not enabled, type `'touch hostname6.interface'` (where interface is the LAN interface such as eri0 or iprb0)
- edit the file and enter `'addif ipv6_address/prefix'`
- you may need to disable/enable networking or just reboot
- to display the interface addressing type `'ifconfig -a'`

6.1.3.2 Installing Java

- check the java version and path setting by typing `java -version` in a terminal window
- if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp
- SUN delivers different JREs for x86, SPARC, 32-bit and 64-bit CPUs

6.1.3.3 Installing the Eiconcard

There are no Solaris drivers available for the Eiconcard, ETIC does not support X.25 when running Solaris.

6.1.4 Mac OS X 10.4.2+ and 10.5

6.1.4.1 Enabling IPv6

- open "System Preferences", select "Network" and highlight the required interface
- select the "Configure…" (10.4.x) or the "Advanced..." (10.5) button
- under the "TCP/IP" tab, select "Configure IPv6"

6.1.4.2 Installing Java

- open the “Utilities” folder within “Applications” of a Finder window
- open “Java Preferences” to ensure J2SE 5.0 is installed
- if required, install the JRE from the CD-ROM or download a version from http://www.apple.com
- within “Java Preferences” ensure that J2SE 5.0 is the top listed JRE under “Java Application Versions”

6.1.4.3 Installing the Eiconcard

There are no Mac OS X drivers available for the Eiconcard, ETIC does not support X.25 when running Mac OS X.

6.1.5 SUSE 9.0, 9.1, 9.3 and 10

6.1.5.1 Enabling IPv6

SUSE provides a helpfile for the sysconfig files. It is recommended to consult this helpfile as there could be slight deviations for each version of the OS.

- SUSE supports IPv6 addressing through YAST
- Launch YAST modules/network devices/Network Card
- Add an additional IPv6 address to the required interface
- Restart the network service with the command `rcnetwork restart`

It is recommended to configure IPv6 manually:

- Edit the `/etc/sysconfig/network/config` file and set `USE_IPV6=yes`
- Edit the `/etc/sysconfig/network/ifcfg-interface` file and set `IPADDR_='ipv6_address/prefix'`

6.1.5.2 Installing Java

- plan to make use of the ETIC installer for Linux with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java –version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp

6.1.5.3 Installing the Eiconcard

DIALOGIC has not released drivers for SUSE 9.0, 9.1, 9.3 nor 10.

6.1.6 SUSE 8.2, Enterprise 9.0, Professional 9.2 and Novell Linux Desktop 9

6.1.6.1 Enabling IPv6

SUSE provides a helpfile for the sysconfig files. It is recommended to consult this helpfile as there could be slight deviations for each version of the OS.

- Edit the `/etc/sysconfig/network/config` file and set `USE_IPV6=yes`
- Edit the `/etc/sysconfig/network/ifcfg-interface` file and set `IPADDR_='ipv6_address/prefix'`
- Restart the network service with the command `rcnetwork restart`

6.1.6.2 Installing Java

- plan to make use of the ETIC installer for Linux with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java -version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp

6.1.6.3 Installing Linux Streams and the Eiconcard

SUSE Linux 8.2 requires Eiconcard Connections for Linux V4R4; SUSE Enterprise Linux 9.0 and Professional 9.2 require V5R1. These drivers are no longer maintained by DIALOGIC but the RPM package may be still part of the installation CD-ROM or available for download at http://www.dialogic.com/products/serial_protocol_adapters/eol.htm (e-mail etic@eurocontrol.int for further information). DIALOGIC provides different Eiconcard drivers per operating system release and CPU architecture. The multi-processor release of the driver is required for platforms with dual-core processors.

The instructions in the Eiconcard "Readmefirst.txt" file should be strictly followed. These instructions describe the installation procedure of GCOM Linux Streams (LIS) and the Eiconcard services. **DO NOT** install routing or PAD services.

For SUSE 8.2, ensure that you are using the latest Build 136 of Eiconcard drivers. When configuring the Eiconcard with the eiconcfg utility you may need to run the utility within a "Linux Console" terminal for proper display or function-key support. Alternatively, type `xhost +` before invoking the `eiconcfg` utility in order to allow the root or super-user to properly display the configuration window in a new terminal.

For Novell Linux Desktop 9, follow the instructions for SUSE Enterprise Linux 9.

The Eiconcard patch must be applied to all version 2 cards (S91v2 and C91v2). The patch is available on the CD-ROM; the two files must be copied to `/usr/lib/eicon/cardsw` overwriting the existing files.

Refer to section 6.2 for the Eiconcard driver settings.

6.1.7 RED HAT 9.0

6.1.7.1 Enabling IPv6

The configuration files for network interfaces are located in the `/etc/sysconfig/` directory and the corresponding `/etc/sysconfig/network-scripts/ifcfg-<interface-name>` script per interface. For a manual static configuration apply the following settings:

- Edit the `/etc/sysconfig/network` file and set `NETWORKING_IPV6=yes` and you can also disable IPv6 autoconfiguration with `IPV6_AUTOCONF=no`
- Edit the `/etc/sysconfig/network-scripts/ifcfg-interface` file and set `IPV6INIT=yes` and `IPV6ADDR='ipv6_address/prefix'`

6.1.7.2 Installing Java

- plan to make use of the ETIC installer for Linux with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java -version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from

the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp

6.1.7.3 Installing Linux Streams and the Eiconcard

RED HAT 9.0 requires Eiconcard Connections for Linux V4R4 which is no longer maintained by DIALOGIC but the RPM package may be still part of the installation CD-ROM or available for download at http://www.dialogic.com/products/serial_protocol_adapters/eol.htm (e-mail etic@eurocontrol.int for further information). DIALOGIC provides different RED HAT 9.0 drivers per CPU architecture; the multi-processor release of the driver is required for platforms with dual-core processors.

The instructions in the "Readmefirst.txt" file should be strictly followed however you must follow the installation instructions for RED HAT 8.0. These instructions describe the installation procedure of GCOM Linux Streams (LIS) and the Eiconcard services. **DO NOT** install routing or PAD services.

If you need to recompile the RED HAT kernel for GCOM Linux Streams (LIS) installation, it is recommended to edit the `/usr/src/linux/Makefile`. This file contains the parameter `EXTRAVERSION` set to `<A number>custom`. It is advised to remove the appended term custom. Refer to the RED HAT documentation on kernel compilation for a detailed explanation of RED HAT kernel versioning.

The Eiconcard patch must be applied to all version 2 cards (S91v2 and C91v2). The patch is available on the CD-ROM; the two files must be copied to `/usr/lib/eicon/cardsw` overwriting the existing files.

Refer to section 6.2 for the Eiconcard driver settings.

6.1.8 RED HAT 3.0 EL

6.1.8.1 Enabling IPv6

The configuration files for network interfaces are located in the `/etc/sysconfig/` directory and the corresponding `/etc/sysconfig/network-scripts/ifcfg-<interface-name>` script per interface. For a manual static configuration apply the following settings:

- Edit the `/etc/sysconfig/network` file and set `NETWORKING_IPV6=yes`
- Edit the appropriate `/etc/sysconfig/network-scripts/ifcfg-<interface-name>` script, set `IPV6INIT=yes` and `IPV6ADDR="ipv6_address/prefix"` e.g. `IPV6ADDR="2001:4b50:c622:1::1/64"`
- IPv6 autoconfiguration can be disabled by adding `IPV6_AUTOCONF=no` to `/etc/sysconfig/network` or to the specific interface script `/etc/sysconfig/network-scripts/ifcfg-<interface-name>`

6.1.8.2 Installing Java

- plan to make use of the ETIC installer for Linux with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java -version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp

6.1.8.3 Installing the Eiconcard

RHEL 3.0 requires Eiconcard Connections for Linux V5R1 which is no longer maintained by DIALOGIC but the RPM package may be still part of the installation CD-ROM or available for download at http://www.dialogic.com/products/serial_protocol_adapters/eol.htm (e-mail etic@eurocontrol.int for further information). DIALOGIC provides different RHEL drivers per CPU architecture; the multi-processor release of the driver is required for platforms with dual-core processors.

The instructions in the "Readmefirst.txt" file should be strictly followed. Unlike RED HAT 9.0 there is no need to recompile the kernel to prepare for GCOM Linux Streams (LIS) installation. These instructions describe the installation procedure of the Eiconcard services. **DO NOT** install routing or PAD services.

Refer to section 6.2 for the Eiconcard driver settings.

6.1.9 RED HAT 4.0 EL, RED HAT 5.0 EL, CentOS 4, CentOS 5

6.1.9.1 Enabling IPv6

The configuration files for network interfaces are located in the `/etc/sysconfig/` directory and the corresponding `/etc/sysconfig/network-scripts/ifcfg-<interface-name>` script per interface. For a manual static configuration apply the following settings:

- Edit the `/etc/sysconfig/network` file and set `NETWORKING_IPV6=yes`
- Edit the appropriate `/etc/sysconfig/network-scripts/ifcfg-<interface-name>` script, set `IPV6INIT=yes` and `IPV6ADDR="ipv6_address/prefix"` e.g. `IPV6ADDR="2001:4b50:c622:1::1/64"`
- IPv6 autoconfiguration can be disabled by adding `IPV6_AUTOCONF=no` to `/etc/sysconfig/network` or to the specific interface script `/etc/sysconfig/network-scripts/ifcfg-<interface-name>`

6.1.9.2 Installing Java

- plan to make use of the ETIC installer for Linux with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java -version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp

6.1.9.3 Installing the Eiconcard

RHEL 4.0 and RHEL 5.0 require Eiconcard Connections for Linux V5R7. CentOS releases are aligned with the respective RHEL releases. The drivers are RPM packages and are available at: http://www.dialogic.com/products/serial_protocol_adapters/Eiconcard_Connections_for_Linux.htm?dl=1&regID=26145. DIALOGIC provides different RHEL drivers per CPU architecture; the multi-processor release of the driver is required for platforms with dual-core processors.

The instructions in the "Readmefirst.txt" file should be strictly followed. Unlike prior versions of the driver, GCOM Linux Streams (LIS) installation is no longer required. These instructions describe the installation procedure of the Eiconcard services. **DO NOT** install routing or PAD services.

Refer to section 6.2 for the Eiconcard driver settings.

6.1.10 SUSE Enterprise Linux 9.0 and 10.0

6.1.10.1 Enabling IPv6

SUSE provides a helpfile for the sysconfig files. It is recommended to consult this helpfile as there could be slight deviations for each version of the OS.

- Edit the `/etc/sysconfig/network/config` file and set `USE_IPV6=yes`
- Edit the `/etc/sysconfig/network/ifcfg-interface` file and set `IPADDR_='ipv6_address/prefix'`
- Restart the network service with the command `rcnetwork restart`

6.1.10.2 Installing Java

- plan to make use the ETIC installer for Linux with JRE (see section 6.3.1)
- alternatively, check the existence of a pre-installed java by typing `java –version` in a terminal window, if the version is lower or higher than 1.5, install the JRE 1.5 from the CD-ROM or download a version from http://java.sun.com/javase/downloads/index_jdk5.jsp

6.1.10.3 Installing the Eiconcard

SUSE Enterprise Linux requires Eiconcard Connections for Linux V5R7. The drivers are RPM packages and are available at http://www.dialogic.com/products/serial_protocol_adapters/Eiconcard_Connections_for_Linux.htm?dl=1&regID=26145. DIALOGIC provides different SUSE Enterprise drivers per CPU architecture; the multi-processor release of the driver is required for platforms with dual-core processors.

The instructions in the “Readmefirst.txt” file should be strictly followed. Unlike prior versions of the driver, GCOM Linux Streams (LIS) installation is no longer required. These instructions describe the installation procedure of the Eiconcard services. **DO NOT** install routing or PAD services.

Refer to section 6.2 for the Eiconcard driver settings.

6.2 Recommended Eiconcard settings

WINDOWS and LINUX – Recommended values: ETIC users are recommended to use the following values for the Eiconcard configuration parameters:

- Default Packet Sizes = 256 (bytes)
- [Number of] Two-way Virtual Circuits (TVC) = 30
- Idle Probe Timer T3 = 99999
- Sync/Direct = POINT-TO-POINT (at both ends of the communication channel)
- Serialisation Protocol = X.21/RS232 (at both ends of the communication channel)
- Node Type = ‘DTE’ or ‘DCE’

Note: The computer running ETIC usually provides the ‘DTE’ node. On a simple communication channel it is necessary to have ‘DTE’ at one end of the channel and ‘DCE’ at the other. However, it is possible to create a more complex communication channel containing three (or more) nodes to form a ‘DTE’ to ‘DTE’ channel which includes a ‘DCE’ node (e.g. DTE <-> DCE <-> DTE). A ‘DCE’ node must be present because the ‘DCE’ provides the clock on layer 2 of a channel.

- Clock Setting (on a ‘DTE’ computer) = EXTERNAL
- Clock Setting (on a ‘DCE’ computer) = INTERNAL/64000

- The existing (default) settings should be acceptable for the other parameters

WINDOWS – Setting the values: Microsoft Windows provides a simple graphical interface window for the setting of Eiconcard configuration parameter values. Users must click the 'Save' button to save any changes to the configuration parameter values.

LINUX – Setting the values: The figure below shows the 'flow' of the Graphical User Interface used during configuration of Eiconcards in a Linux environment. It also shows an example Eiconcard configuration for Linux. In the following figure:

Each box represents a window displayed on the screen.

Each box lists the parameters shown in the window. Within each window, focus can be moved from one parameter value to the next so that parameter values can be changed.

Each flow arrow marked <F4> shows how window focus can be changed to display the 'next' window when the function key 'F4' is pressed.

The horizontal flow arrows leaving the third box indicate that the focused parameter ('X.25' or 'Direct') determines which window will be displayed when 'F4' is pressed.

Each flow arrow marked <F3> shows how window focus can be changed to display the 'previous' window when the function key 'F3' is pressed.

Although not shown in the figure below, the function key 'F2' must be pressed to save the configuration settings.

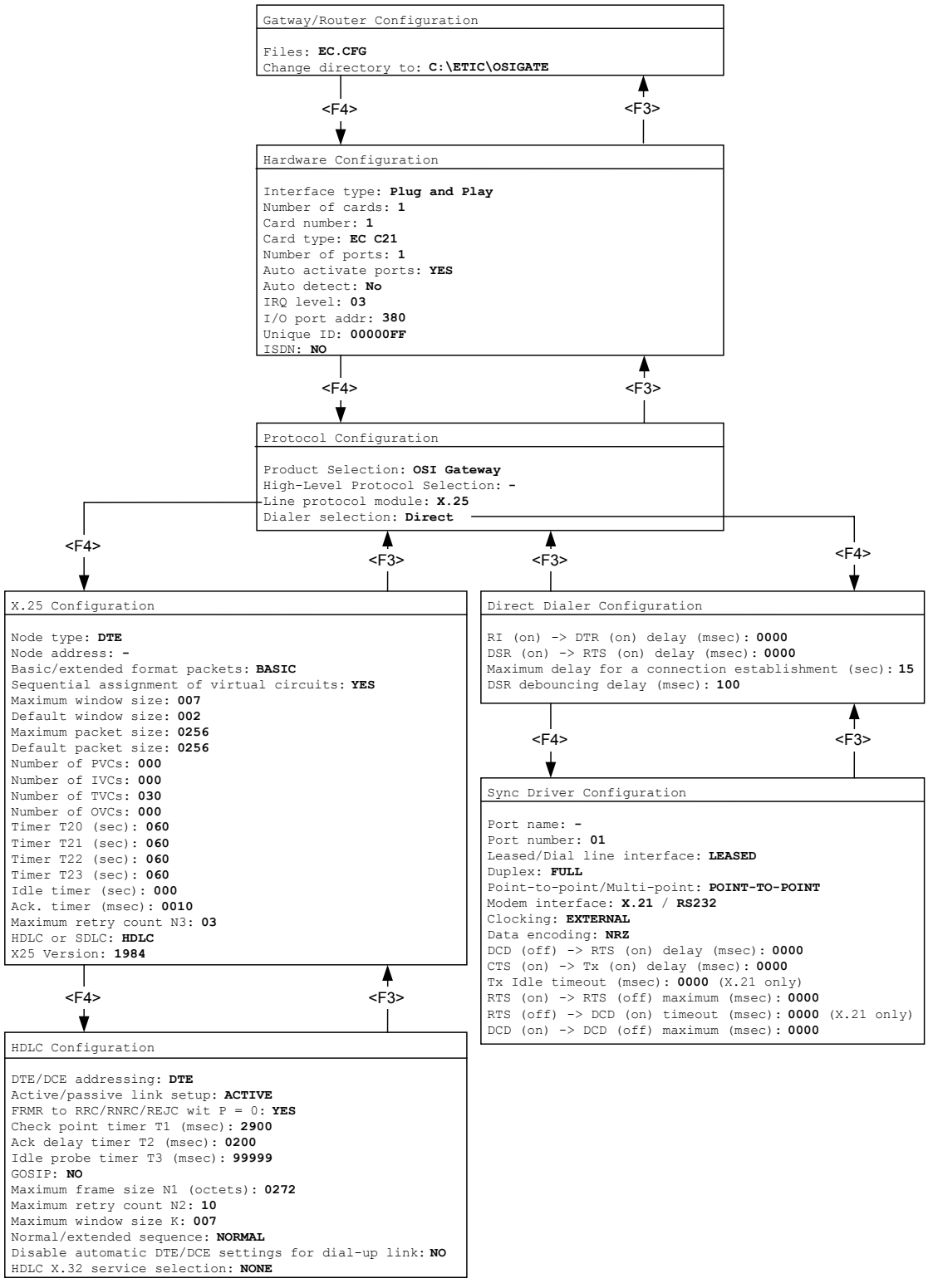


Figure 1: Eiconcard Setting Windows for Linux

6.3 ETIC Installation

There are two methods to install ETIC version 3.2:

1. Using the ETIC automated installation routine, the installer, or
2. Manual installation.

Whilst the first option is preferred, ETIC automated installers are only available for Windows, Linux, Mac OS X and Solaris operating systems. Manual installation will be required for all other compatible operating systems.

6.3.1 The ETIC installer

When using the installer, ensure the appropriate networking interfaces are available on your platform (IPv4, IPv6 and optionally X.25).

Insert the ETIC CD-ROM:

- Windows: the standalone flash player menu will appear
- Mac OSX: manually launch the flash player menu by opening 'menu-mac'
- Linux: open a shell, enter the CD-ROM root folder and type './menu'
- For all other systems (including Linux systems that failed to run the menu), the shockwave/flash player plug-in needs to be installed into the web browser used to view the file 'menu.swf'. This plug-in is available from the CD-ROM or from the http://www.adobe.com web site.

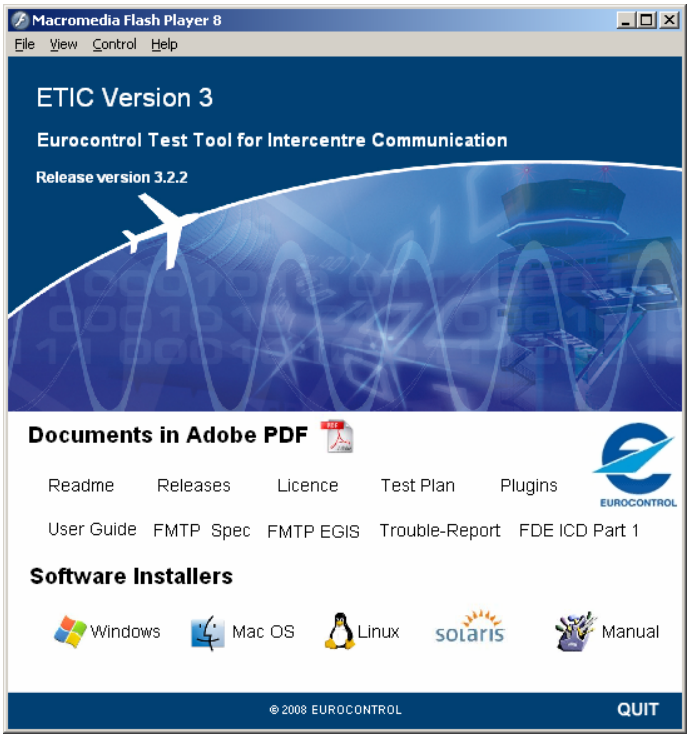


Figure 2: ETIC CD-ROM menu

Select the appropriate ETIC software installer page for Windows, Mac OS X, Linux or Solaris and follow the specific instructions. For Windows and Linux one can also include installation of the Java run-time for ETIC. **Windows Vista users are required to proceed to manual installation as the installer will fail to launch.**

You will then be presented with the main installation menu illustrated in Figure 3.

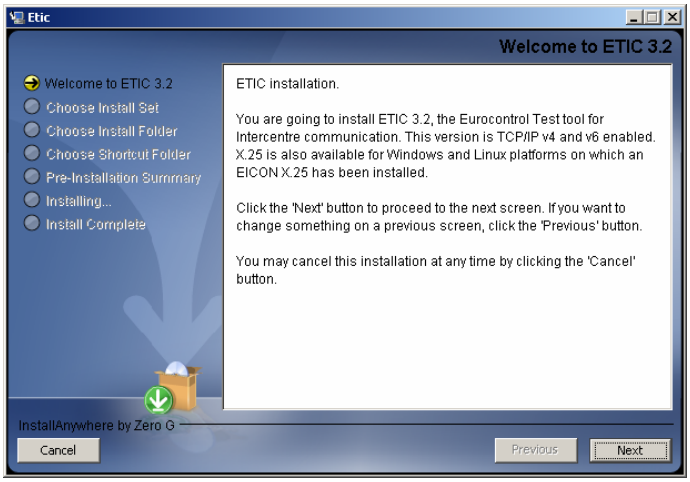


Figure 3: ETIC installation menu

Proceed with the installation by following the instructions indicated by the installer:

- Under Windows or Linux, there is an option to include X.25 support (which will result in the copying of the eicon.dll file or the compilation of an eiconcard library file)
- The target installation directory can also be specified
- Shortcut(s) can be defined

Once installed, simply use the shortcut to start ETIC. During Windows and Mac OS X installation, you can have the installer add a desktop icon or a quick launch button to start ETIC. On Linux/Solaris, you will need to execute the shortcut from a terminal window or manually create a desktop icon to launch the application.

If any warnings or errors are encountered during the ETIC installation process, details can be found in the 'etic_InstallLog.log' file in whichever installation directory was selected for the installation.

The ETIC installers can also be launched directly by browsing the CD-ROM:

- Windows

  The installer is a self-extracting executable file named `etic.exe`

- Mac OS X

  The installer is an archived package named `etic.zip`

- Linux and Solaris

  The installer is a self-extracting executable files are named `etic.bin`. From a console or terminal window enter the command `sh <path>/etic.bin`, where `<path>` is the absolute or relative path of the directory where `etic.bin` is located. For example, if your CD-ROM's mount point is `/media/cdrom/`, then an example path would be `/media/cdrom/installers/linux/with_java`. You do not require root permissions to install ETIC on Linux or Solaris.

6.3.2 Manual installation

In the event of manual installation or the use of an installer without JRE installation, ensure that JVM 5.0 is installed and is present in the path. From a terminal or console window, enter the command `java -version` to verify that the default JRE is suitable.

With the exception of the X.25 C library (for Windows and Linux only), ETIC is a pure Java application. As a result, ETIC can run virtually on any platform on which JVM 5.0 is installed. Therefore, even if the target platform is not Windows, Linux, Mac OS X or Solaris, it is possible to install the ETIC software once the suitable JVM is installed.

Installing ETIC is simply a question of copying the jar files and two configuration files from the CD-ROM and then starting the JVM to run `ETIC.gui.ETIC` with the right java classpath.

For manual installation ETIC needs the following files which are available on the CD-ROM:

- etic.jar
- log4j-1.2.8.jar
- commons-beanutils.jar, commons-digester.jar, commons-logging.jar
- groovy-all-1.5.4.jar
- jh.jar
- help.jar
- log.ini (initialization file for log4j)
- conformance.dat (conformance test session)

Copy these files into a directory on the target platform and launch ETIC with the right classpath set. For example, on Windows, it is possible to start ETIC by typing in a command window:

```
java –classpath "etic.jar;log4j-1.2.8.jar;commons-beanutils.jar;
commons-digester.jar;commons-logging.jar;groovy-all-1.5.4.jar;
jh.jar;help.jar" etic.gui.Etic
```

For Linux/Unix systems, the above semi-colon is to be replaced by a colon. If you are not running ETIC from the installation directory, you will need to add an absolute or a relative path to the filenames.

It is also possible to give an extra-argument corresponding to a session file that is to be loaded during ETIC start-up.

7 ETIC Functions

This section describes ETIC from a functional point of view. The following section describes how to configure and interact with these functions.

7.1 Building blocks

ETIC is a Java application that has been designed to assist the testing and debugging of operational systems implementing the FMTP protocol (IPv4 and IPv6) and the FDE ICD Part 1 protocol (X.25).

ETIC hardware platforms can scale from laptops to servers. A file system must be available for ETIC to save parameters, sessions and log data.

Figure 4 is a high level overview of ETIC simulating several communication partners interfacing to a SuT using FMTP or FDE ICD Part 1. ETIC can simulate full conformance to FMTP/FDE ICD Part 1 but can also deliberately simulate a non-conformant communication partner to better test SuT behaviour.

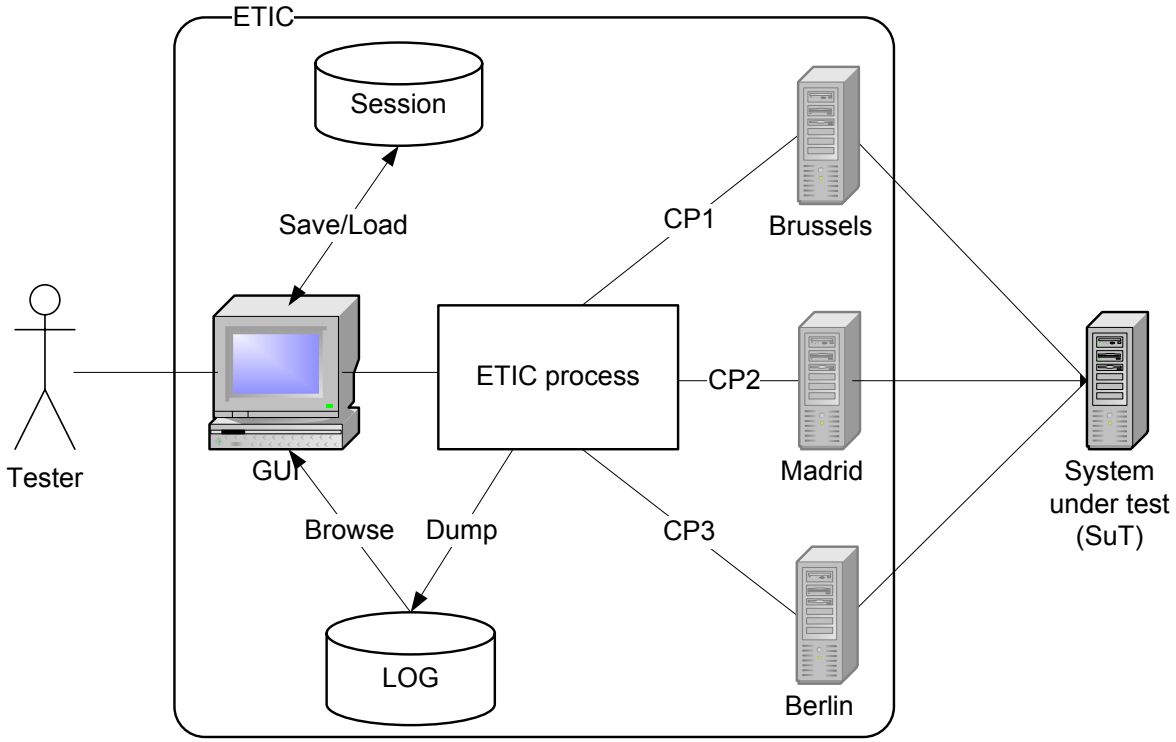


Figure 4: Overview of ETIC

7.2 Session

Sessions play a central role in ETIC. A session file contains all the essential parameters to define a CP and an associated environment in which a particular instance of ETIC runs.

The following entities are key components of a session file:

- The CP(s) ETIC has to simulate
- Message patterns
- Relays
- Smart responses
- Sequences

Furthermore, there are three types of session files:

1. **Normal** session file(s): the ETIC user has full control over the session file content. Users can add, edit or deleted any type of entity.
2. **Locked** session file(s): editing session file content is restricted to a specific number of CP parameters. Users cannot add or delete session entities but can see all session file content. Thereby, locked sessions are a means to share session files with other users and to avoid their modification.
3. **Conformance** session file: a conformance session is designed to assess the conformance of a SuT to the FMTP protocol. Only two predefined sequences can be executed.

ETIC can open session files created by any previous version. A previous version of ETIC can open the session files created with a more recent version but will ignore the definitions relating to unknown features. For example, ETIC 3.1.0 would ignore relay definitions of an ETIC 3.2.2 file, if the 3.2.2 file is saved with ETIC 3.1.0, the relay definitions will be lost.

Unless explicitly stated in this User Guide, subsequent use of the term “session” refers to a “normal session”.

7.3 Communication Partners (CPs)

7.3.1 Communication Partner types

As illustrated in Figure 4 above, ETIC can simulate FMTP systems with which the SuT establishes a connection.

Within a given session many CPs can be defined. The total number of CPs is only constrained by the available memory of the ETIC platform.

To define a CP, five connection types are supported:

1. **TCP/IP client**: in this case the ETIC CP will simulate a FMTP client that connects to a SuT acting as FMTP server.
2. **TCP/IP server**: in this case the ETIC CP will simulate a FMTP server that passively listens for incoming connections from a SuT acting as FMTP client.
3. **X.25 caller**: in this case the ETIC CP will simulate a FDE ICD Part 1 peer that calls a FDE ICD Part 1 SuT.
4. **X.25 listener**: in this case the ETIC CP will simulate a FDE ICD Part 1 peer that waits for incoming calls from a FDE ICD Part 1 SuT.
5. **X.25 both**: in this case the ETIC CP will simulate a FDE ICD Part 1 peer that calls a FDE ICD Part 1 SuT but is also capable of waiting for incoming calls.

For each connection type a set of protocol specific parameters can be configured:

- Physical network interface
- Network addresses (IPv4, IPv6 or X.121)
- TCP port number (FMTP only)
- User data field (FDE ICD Part 1 only)
- Protocol and connection timers

7.3.2 States

7.3.2.1 State machine

The FDE ICD Part 1 (Reference [RD1]) and FMTP (Reference [RD3]) protocols define a state machine and the events that trigger state transitions. The objective of the state machine is to synchronise two peers in reaching the DATA_READY state in which operational data exchange can occur. Each intermediate state represents a finite step towards the achievement of connection or association establishment.

ETIC implements all of these states and state transitions complying with both the FMTP and FDE ICD Part 1. However, for the purpose of this test tool, it is necessary to introduce a number of additional states to provide a better understanding and control of the communications between ETIC and the SuT.

7.3.2.2 Target states

In order to establish a connection or an association between a CP and the SuT, a certain number of actions have to be performed by both peers in a well defined sequence. Every ETIC action can lead to a possible state change and visa versa.

However, there is a fundamental difference between state changes that are a result of ETIC actions and those provoked by the remote SuT. Indeed, the state changes resulting from

ETIC actions are under control, in other words, it is possible not to perform the action. For instance, if the FMTP protocol mandates that an IDENTIFICATION message is to be sent, the user may deliberately override the protocol and not send the message. In such cases, the CP state will not change. On the other hand, if the FMTP protocol mandates that the SuT is to send a given message, there is no way that ETIC can avoid message receipt and a resulting state change.

In view of the above, ETIC has defined the notion of target states. Target states are states in which an ETIC CP can remain until the user triggers the action that results in a state transition. During the establishment of a connection, it is possible to instruct ETIC to perform all the necessary actions to reach a user-selected target state. For example it is possible to instruct ETIC to remain forever in the READY state; regardless of the incoming SuT messages, ETIC will remain in READY. In the following state diagrams, target states are shaded in green.

Note: When a CP is configured part of a relay (see section 7.7) between two external peers, the available range of target states is limited.

7.3.2.3 Transient states

When in a transient state, ETIC cannot control state transitions. Indeed, the characteristic of a transient state is that an external event, usually generated by the SuT will determine a state transition. For example, an ETIC CP cannot remain forever in ASSOCIATION_PENDING as an incoming STARTUP message will force it to enter the DATA_READY state. In the following state diagrams, these states are shaded in yellow or in grey. **'Grey' states are internal to ETIC and are not visible to the user.**

7.3.2.4 CP state mechanism

ETIC can establish a connection on the basis of a user-selected target state. **When a target state is assigned to a CP, ETIC performs all the necessary FMTP (or FDE ICD Part 1) protocol actions to ensure the CP changes from its current state to the target state.**

Once their target state is reached, ETIC CPs do not perform any action i.e. if the current state = the target state. However, if the SuT or any other external event puts the CP out of its target state, it will immediately attempt to reach its originally assigned target state.

At any time, the set of available targets depends on the CP's current state with regard to the FMTP (or FDE ICD Part 1) state machine. In practice, with the exception of the transition from DATA_READY to READY, available target states are only those that progress towards DATA_READY. A previously reached state is only available by returning via the IDLE state.

The next sections describe the existing states for each kind of CP.

7.3.3 TCP Client

The FMTP TCP client state machine implemented by ETIC is shown in Figure 5. The initial state of an ETIC CP acting as TCP client is DISABLED i.e. the CP definition exists but is inactive. It is to be interpreted as if the CP did not exist. Once activated, the ETIC CP enters the IDLE state. ETIC then attempts to connect to the SuT acting as a TCP server (MT-Connect). Until the connection is established or T_Reply does not expire, the CP remains in the CONNECTION_PENDING state. Indeed, the SuT must accept the TCP connection within T_Reply, otherwise the connection attempt will be abandoned and a new TCP connection will be issued by ETIC.

Once the connection is established, the CP state changes to TCP_ESTABLISHED. According to the FMTP protocol, ETIC must send an identification message and preferably before the server's Ti timer expires. The CP enters the ID_PENDING state.

The SuT must reply with its identification message preferably before the client's Ti timer expires. If the SuT does not reply within Ti or closes the connection, the CP enters the REJECTED state and will attempt to re-connect after a new Ti period[1].

Upon receipt of a proper identification message in the ID_PENDING state, the CP enters the PRE_READY state. Following this, ETIC sends an ACCEPT message to the SuT enters the READY state.

Finally, to reach the DATA_READY state, ETIC will need to send a STARTUP message to the SuT and receive a STARTUP message from the SuT (MT-Associate). While remaining in the ASSOCIATION_PENDING state, the CP will continue sending STARTUP messages every Tr seconds until the maximum number of STARTUP messages is reached.

From the DATA_READY state, ETIC can return to READY by sending a SHUTDOWN message (MT-Stop). Receipt of a SHUTDOWN message in the DATA_READY state (MT-Stop), will make the ETIC CP return to the ASSOCATION_PENDING state from which it will attempt to send STARTUP messages.

In all of the following state diagrams, target states are shaded in green and transient states are shaded in yellow or in grey ('grey' states are internal to ETIC and are, therefore, not visible to the user).

[1] This is implementation specific to avoid ETIC looping after failed connections

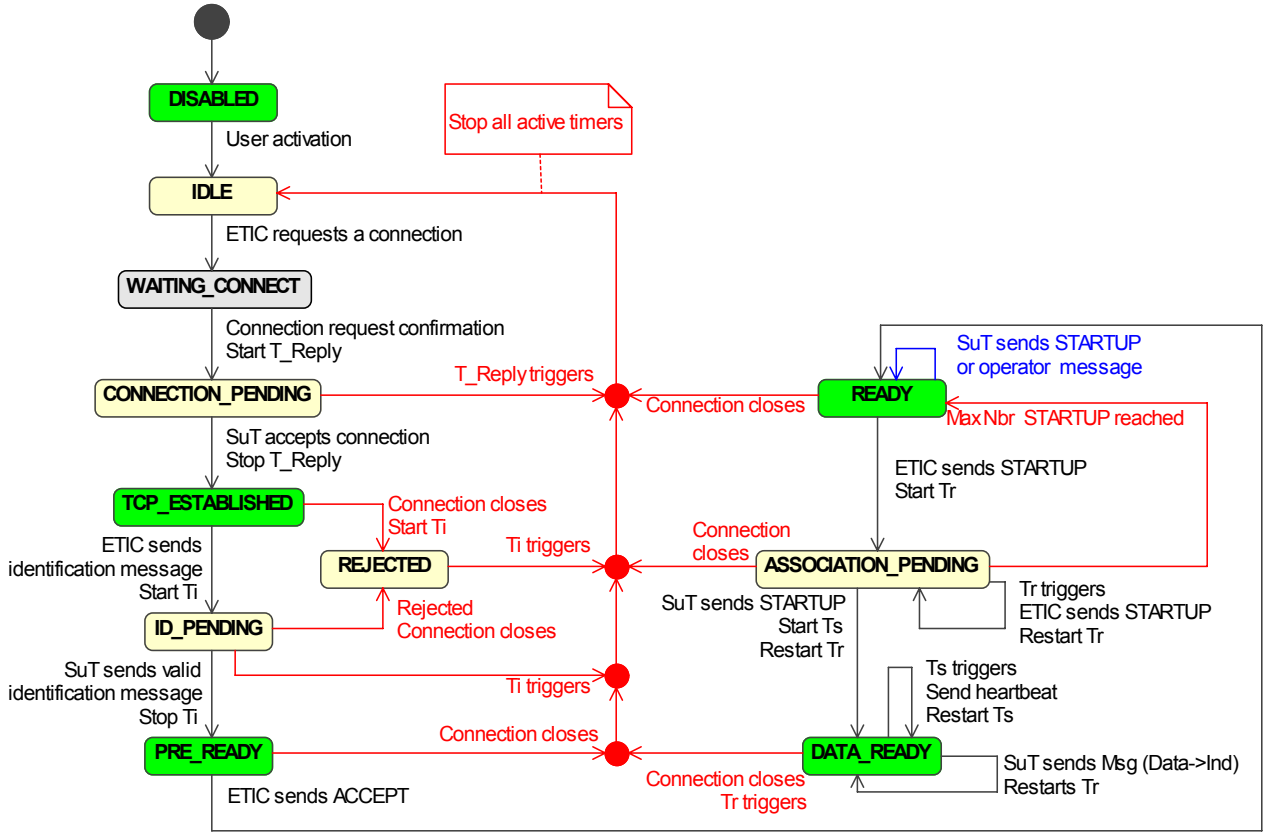


Figure 5: ETIC FMTP TCP client state machine

The target states that can be reached from a given current state are shown in the Table 3 below. If a target state is not directly accessible, it can be reached by transiting via the IDLE state and rebuilding the connection. For example, if ETIC is in the READY state, it cannot return to TCP_ESTABLISHED state directly. The connection must first be closed and then re-established in order to reach the TCP_ESTABLISHED state.

In Table 3, green indicates a target state, yellow indicates a transient state, white indicates that a direct transition is available and blue indicates that it is necessary to start from the IDLE state.

Current state ↓ / Target state →	DISABLED	TCP_ ESTABLISHED	PRE_READY	READY	DATA_READY
DISABLED					
IDLE					
CONNECTION_PENDING					
TCP_ESTABLISHED					
ID_PENDING					
REJECTED					
PRE_READY					
READY					
ASSOCIATION_PENDING					
DATA_READY					

Table 3: State-specific TCP client target states

7.3.4 TCP Server

The FMTP TCP server state machine implemented by ETIC is shown in Figure 6 below. The initial state of an ETIC CP acting TCP client is DISABLED i.e. the CP definition exists but is inactive. It is to be interpreted as if the CP did not exist. Once activated, the ETIC enters the IDLE state and starts listening for client connections on a predefined TCP port. When a SuT connects to this port, the CP state changes to SYSTEM_ID_PENDING and receipt of an identification message within Ti seconds is expected. If an invalid identification message or none is received, ETIC closes the connection and returns to the IDLE state. If a valid identification message is received, the ETIC CP enters the READY_SEND_ID state.

By sending its identification message, ETIC can then enter the ID_PENDING state. When the SuT returns an ACCEPT message before Ti expires, the CP enters the READY state (M-Connect). Otherwise, the connection is closed and the CP returns to the IDLE state. From the READY state, the transitions are the same as in the case of the TCP client.

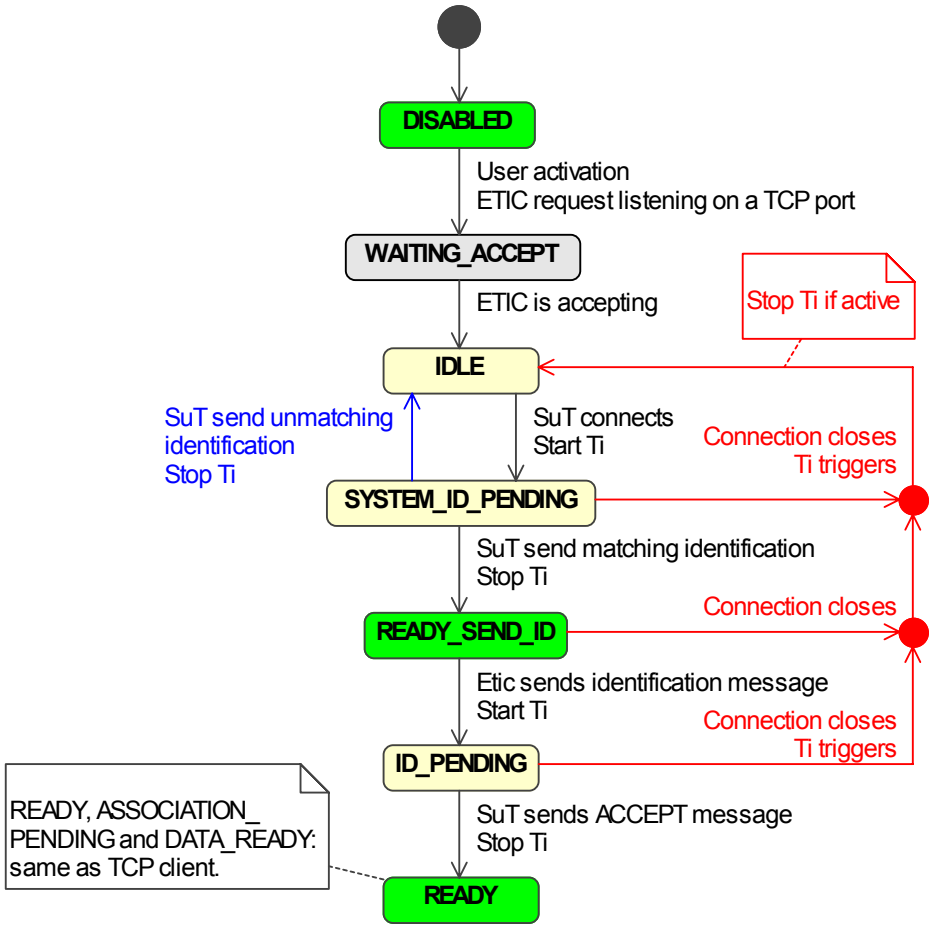


Figure 6: ETIC FMTP TCP server state machine (up to READY)

The target states that can be reached from a given current state are shown in Table 4 below. If a target state is not directly accessible, it can be reached by transiting via the IDLE state and rebuilding the connection. For example, if ETIC is in the READY state, it cannot return to READY_SEND_ID directly. The connection must first be closed and then re-established in order to reach the READY_SEND_ID state.

In Table 4, green indicates a target state, yellow indicates a transient state, white indicates that a direct transition is available and blue indicates that it is necessary to start from the IDLE state.

Current state ↓ / Target state →	DISABLED	READY_SEND_ID	READY	DATA_READY
DISABLED				
IDLE				
SYSTEM_ID_PENDING				
READY_SEND_ID				
ID_PENDING				
READY				
ASSOCIATION_PENDING				
DATA_READY				

Table 4: State-specific TCP server target states

7.3.5 X.25 Caller

The FDE ICD Part 1 caller state machine implemented by ETIC is shown in Figure 7. The initial state of an ETIC CP acting FDE ICD Part 1 caller is DISABLED i.e. the CP definition exists but is inactive. It is to be interpreted as if the CP did not exist. Once activated, the ETIC CP enters the IDLE state. ETIC then issues an X.25 call request and the CP enters the CALLING state. If the X.25 call confirmation is not received within the time period T_Reply, the CP returns to the IDLE state and will issue a new call request. When the X.25 call confirmation is received in time, the CP enters the READY state. From the READY state, the transitions to DATA_READY are the same as in the case of the TCP client.

A notable difference between FMTP and FDE ICD Part 1 is the expiry of timer Tr in DATA_READY. In the case of FDE ICD Part 1, the CP returns to the ASSOCATION_PENDING state.

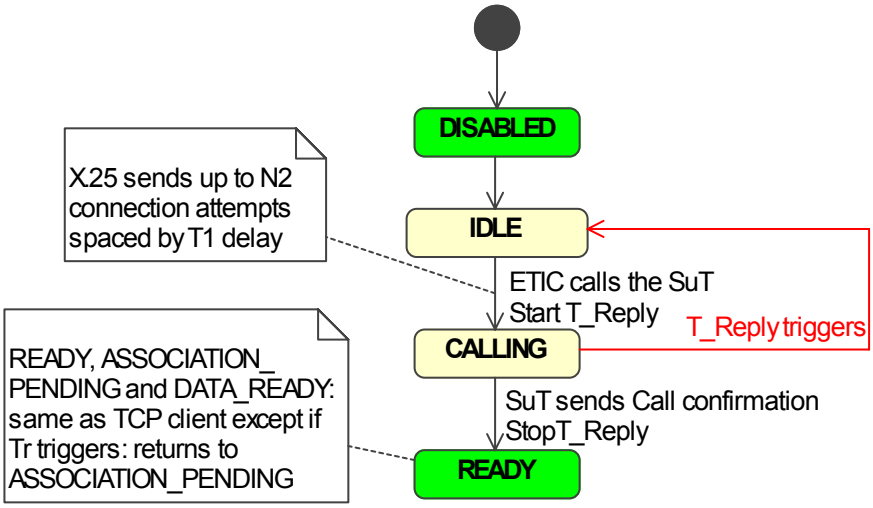


Figure 7: ETIC FDE ICD Part 1 X.25 caller state machine

The target states that can be reached from a given current state are shown in Table 5 below. Green indicates a target state, yellow indicates a transient state, white indicates that a direct transition is available and blue indicates that it is necessary to start from the IDLE state.

Current state ↓	Target state →	DISABLED	READY	DATA_READY
DISABLED				
IDLE				
CALLING				
READY				
ASSOCIATION_PENDING				
DATA_READY				

Table 5: State-specific X.25 caller target states

7.3.6 X.25 Listener

The FDE ICD Part 1 listener state machine implemented by ETIC is shown in Figure 8. The initial state of an ETIC CP acting FDE ICD Part 1 listener is DISABLED i.e. the CP definition exists but is inactive. It is to be interpreted as if the CP did not exist. Once activated, the ETIC CP enters the IDLE state and listens on the predefined X.25 port. When a SuT connects to this port, the X.25 call indication is detected and the CP enters the PRE_READY state. By sending a X.25 call confirmation, ETIC can then enter the READY state.

From the READY state, the transitions are the same as in the case of the X.25 caller.

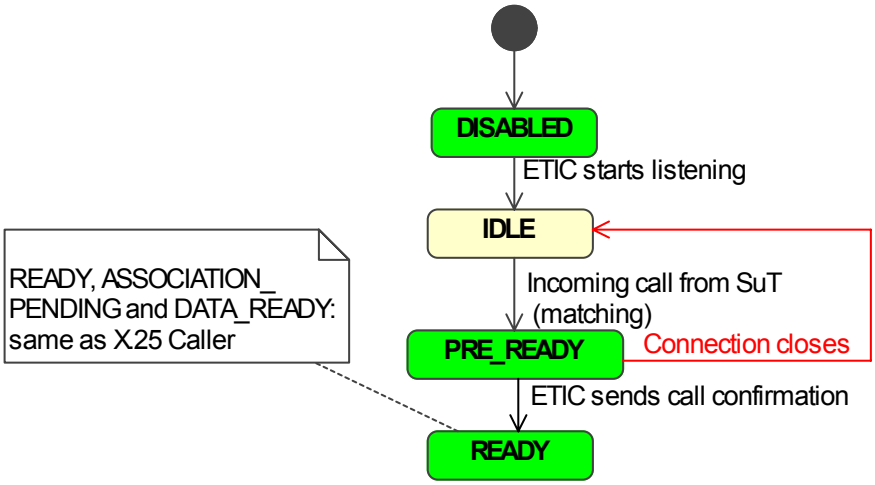


Figure 8: ETIC FDE ICD Part 1 X.25 listener state machine (up to READY)

The target states that can be reached from a given current state are shown in Table 6 below. Green indicates a target state, yellow indicates a transient state, white indicates that a direct transition is available and blue indicates that it is necessary to start from the IDLE state. If a target state is not directly accessible, it can be reached by transiting via the IDLE state and rebuilding the connection. For example, if ETIC is in the READY state, it cannot return to PRE_READY directly. The connection must first be closed and then re-established in order to reach the PRE_READY state.

Current state ↓	Target state →	DISABLED	PRE_READY	READY	DATA_READY
DISABLED					
IDLE					
PRE_READY					
READY					
ASSOCIATION_PENDING					
DATA_READY					

Table 6: State-specific X.25 listener target states

7.3.7 X.25 Both

ETIC can support CPs acting as both X.25 caller and X.25 listener. The FDE ICD Part 1 both state machine implemented by ETIC is shown in Figure 9. The initial state of an ETIC CP acting FDE ICD Part 1 both is DISABLED i.e. the CP definition exists but is inactive. It is to be interpreted as if the CP did not exist. Once activated, the ETIC CP enters the IDLE state and listens on the predefined X.25 port. Right after this event, ETIC then issues an X.25 call request and the CP enters the CALLING state. If the X.25 call confirmation is not received before the T_Reply timer expires, the CP returns to the IDLE state, and will issue a new call request.

When the X.25 call confirmation is received in time, the CP enters the READY state. If ETIC receives a call indication from the same Network Service Access Point (NSAP) with whom there is an outstanding X.25 call request, the CP enters the COLLISION state. The call collision is then resolved according to the FDE ICD Part 1 specification:

- If the ETIC CP NSAP address is greater than the SuT's, then ETIC clears the incoming call and the CP returns to the CALLING state.
- Otherwise, ETIC accepts the incoming call and the CP enters the POST_COLLISION state until the SuT clears the outstanding call request. If the SuT does not clear the call before timer T_Reply expires or data is received by the SuT, then ETIC clears its outstanding call and issues a warning. Then the CP enters the READY state.

From the READY state, the transitions are the same as in the case of the X.25 caller.

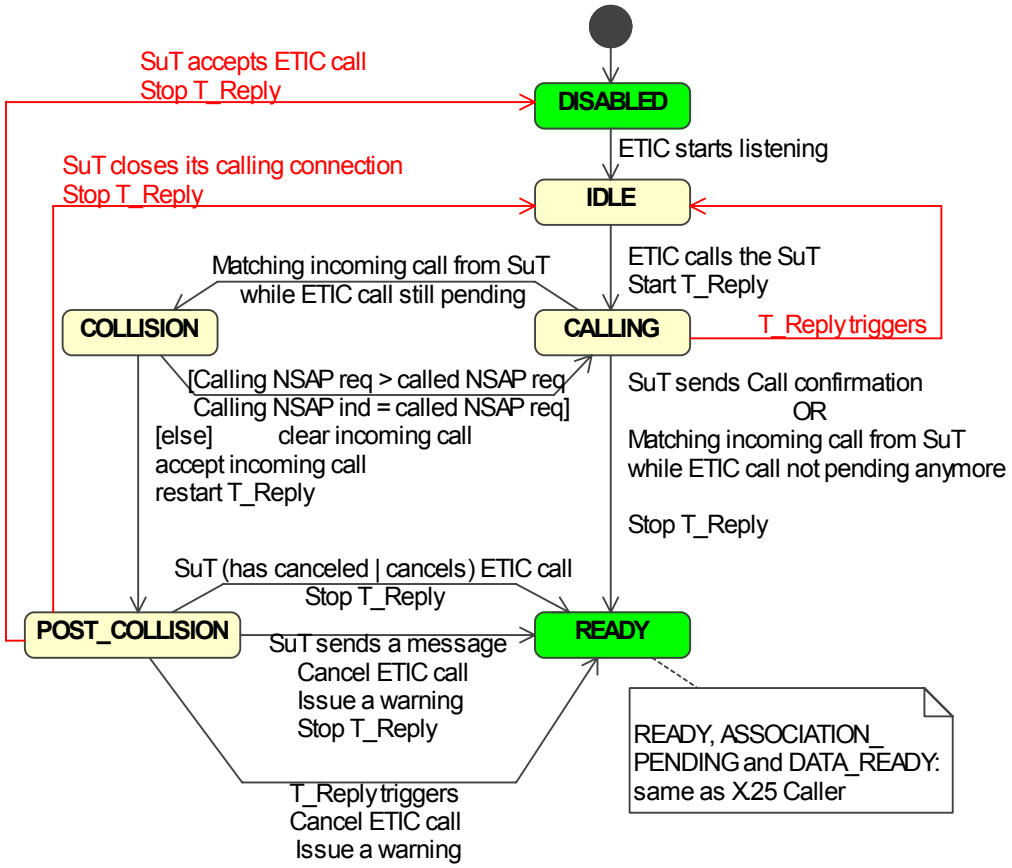


Figure 9: ETIC FDE ICD Part 1 X.25 both state machine (up to READY)

The target states that can be reached from a given current state are shown in Table 7 below. Green indicates a target state, yellow indicates a transient state, white indicates that a direct transition is available and blue indicates that it is necessary to start from the IDLE state. If a target state is not directly accessible, it can be reached by transiting via the IDLE state and rebuilding the connection.

Current state ↓ / Target state →	DISABLED	READY	DATA_ READY
DISABLED			
IDLE			
CALLING			
COLLISION			
POST_COLLISION			
READY			
ASSOCIATION_PENDING			
DATA_READY			

Table 7: State-specific X.25 both mode target states

7.4 Message patterns: macros and regular expressions

7.4.1 Messages types and patterns

In addition to connection establishments/disconnections, a CP and the SuT interact together by exchanging data messages. A data message is characterised by its type and content.

FMTP (Reference [RD2]) defines five messages types (FDE ICD Part 1 (Reference [RD1]) supports four):

- Identification (FMTP only)
- Operational
- Operator
- Status
- System

Users can create, edit and delete messages named message patterns. Internally, ETIC operates with message patterns. For example, consider the CP acting as FMTP client with identification “FOO” connecting to a SuT with identification “BAR”. One of the steps in the connection establishment process is that the client must send an identification message with content: “FOO-BAR” i.e. <local id>-<remote id>. However, the SuT must reply with his <local id>-<remote id> i.e. “BAR-FOO”. The use of such message patterns facilitates ETIC coding and provides a very powerful tool for users.

Indeed, for users it is also very useful to be able to monitor messages of a given type and whose content matches a specific pattern. For instance, it is possible with ETIC to monitor an operator message containing the substring “hello” or “Hello”.

Thus a message pattern is the combination of a message type and a content pattern.

7.4.2 Patterns for outgoing messages

In order to send a message from a CP to the SuT, a message pattern is used. A message pattern is defined by the combination of its type and content.

The available message types are the five types defined by the FMTP specification. When such a message pattern is to be sent to the SuT, ETIC appends the appropriate FMTP or FDE ICD Part 1 message header.

The message pattern content is composed of:

- **Fixed text:** it is also possible to insert non-printable ASCII characters using the “^” character followed by 2 hexadecimal digits. For example, ^0D corresponds to the carriage return (CR) character.
- **Macros:** which are substituted by a meaningful string before the message is sent. ETIC provides 7 macros:
  - $l will be substituted by the ETIC CP’s FMTP identifier
  - $r will be substituted by the SuT’s FMTP identifier
  - $L will be substituted by the ETIC CP’s ATS unit name
  - $R will be substituted by the SuT’s ATS unit name
  - $N will be substituted by an internal number that is incremented after each message transmission. This message number may be referred to by message responses such as a Logical Acknowledgment Message (LAM)
  - $n is only used in the context of smart responses (see section 7.5). It will be substituted by the message number of the incoming message
  - $t will be substituted by the ETIC system time, a time offset can be specified with the syntax $t or $t+hhmm or $t-hhmm where hhmm represents the hours and minutes. Note that ‘hh’ can be in the range 00 to 99 and that ‘mm’ can also be in the range 00 to 99

The use of macros can save a lot of time. For example, all ETIC FMTP identification messages simply refer to the single pattern: $l-$r.

Further to the above message types, ETIC provides a new message type named RAW. When such a message pattern is to be sent to the SuT, no header is added by ETIC. For this message type, the header has to be part of the message content and becomes user editable. The RAW type is to be used with caution as ETIC performs no syntax or semantic checks when sending the message to the SuT. The benefit of this message type is that it allows users to validate SuT behaviour upon receipt of a message with an invalid header.

7.4.3 Patterns for incoming messages

ETIC can trigger specific actions when an incoming message matches a message pattern. "Matching" means:

- the type of the incoming message and the type of the message pattern are the same[2], and
- the content of the incoming message matches with the content of the pattern.

As for outgoing messages (section 7.4.2), fixed text and macro[3] patterns can be defined for incoming messages. Once the macros are substituted by real values and combined with any fixed text, the resulting string is interpreted as being a regular expression. Indeed a regular expression (regex or regexp for short) is a special text string for describing a search pattern which is widely supported by software applications and programming languages such as Java. ETIC users are invited to consult www.regular-expressions.info as ETIC makes use of regular expressions for the definition of message pattern content. Their use will make finding complex information a lot easier.

For instance, assuming that the CP identification value is "FOO" and that the SuT identification value is "BAR", the following pattern:

```
Type: operator
Content: .*$l.*$r.*
```

will be interpreted as the following regular expression:

```
Type: operator
Content: .*FOO.*BAR.*
```

This matches with any incoming operator message that contains

1. 0, 1 or many characters, followed by
2. FOO, followed by
3. 0, 1 or many characters, followed by
4. BAR, followed by
5. 0, 1 or many characters

Thus, `FOOBAR`, `FOO-BAR` and `you are FOO, I am BAR` match, but `Foo-Bar` doesn't.

Although it is not the purpose of this User Guide to describe regular expressions, the following regex special characters would be the most commonly used by ETIC users:

- a dot (.) means any character;
- x* means zero, one or many of character x;
- x+ means one or many of character x;
- x? means zero or one of character x;

[2] For incoming messages, the raw type is not used. Indeed, if the incoming message type does not comply to FMTP or FDE ICD Part 1, the message is considered invalid and is dumped.
[3] For incoming messages, $N is not used.

- Characters within square brackets means “any of the character(s) inside” e.g. [aA] matches with the two characters ‘a’ and ‘A’;
- To use the ‘.’, ‘*’, ‘+’, ‘?’, ‘]’ and ‘[‘ as characters, it is necessary to use the escape special character backslash e.g. 3 * 5 = 15 matches with the string “3 * 5 = 15”.

7.5 Smart responses

The smart response mechanism allows the definition of an automatic message to be sent in response to a specific incoming message. Each smart response can by enabled or disabled. In the latter case, the smart response is still defined but is simply ignored. In addition, ETIC CPs can also be smart response enabled or disabled. In the latter case, the CP doesn’t make use of the smart response mechanism.

A smart response is created by associating an input message pattern with an output message pattern (which will be the response).

When a “smart response” enabled CP receives a message from the SuT, it compares the incoming message with the input pattern of all the enabled smart responses. When a match is found, it creates the response message on the basis of the output pattern and sends it to the SuT with the specified response delay. Even though a first match is found, ETIC may continue searching for additional matches depending on the smart response configuration. As a result, a single incoming message can trigger more than one response if more matches are found.

To summarise, the following six pre-requisites are necessary for an ETIC CP to send message ‘B’ as a smart response to incoming message ‘A’ from a System under Test:

- a CP is defined in ETIC to enable two way communications with the SuT
- the CP is ‘Smart Response Enabled’ in the ‘Modify Communication Partners’ window
- message pattern ‘A’ is defined in ETIC
- message pattern ‘B’ is defined in ETIC
- a Smart Response is defined and enabled in ETIC where ‘A’ is the ‘In’ message and ‘B’ is the ‘Out’ message in the ‘Smart Response’ window
- the SuT sends message ‘A’

7.6 Sequences

Sequences are an advanced concept of ETIC. With the help of sequences, a multitude of scenarii can be prepared for several ETIC CPs and then played against the SuT(s) in an automatic fashion.

Naturally, a sequence is composed of a number of items that are executed sequentially. Each item being:

1. **Actions:** compulsory statements that a CP has to execute. There are five kinds of actions:
   a. Setting the target state of an ETIC CP, e.g. set the target of the CP “Brussels” to READY
   b. Sending a message from an ETIC CP to a SuT, e.g. send operator message hello_world from CP “Brussels” to the SuT
   c. Displaying a user-defined text in the log (mostly for documentation purposes), e.g. “Starting test #3”
   d. Displaying a pop-up window, e.g. display a pop-up with the following message “Disconnect the SuT”
   e. Inserting a time wait, e.g. wait 10s

2. **Event watches:** events that are to be monitored by a CP. A specific time period is associated for the event to occur. If the event occurs within the time period, the event watch is considered successful, otherwise it is a failure. There are two kinds of events:
   a. The ETIC CP should receive a given message from the SuT, e.g. CP “Brussels” should receive a matching message within the next 5s
   b. The ETIC CP should enter a given state, e.g. CP “Brussels” should become READY within the next 10s

   These events can be further characterised by two criteria:
   a. An event watch can be blocking or non-blocking. If it is blocking, the sequence will wait until the event occurs or the associated time period expires. If it is non-blocking, the sequence will continue running and the event watcher will stay active in the background
   b. An event watch can be negated. In this case, the watch is successful if the event watch does not occur within the associated time period.

3. **Sub-sequence:** is the term describing a nested sequence within a sequence. A sub-sequence can contain another sub-sequence and there is no limitation on the number of sub-sequences within a sub-sequence.
   To prevent a loop between sequence definitions, a sequence (A) may only nest another sequence (B) if neither sequence (B) nor any of its sub-sequences nests sequence (A). For the same reason, it is not possible for a sequence to refer to itself.

Sequences can be executed concurrently and it is also possible to iterate a sequence a finite or infinite number of times.

Sequences have to be written with due care because synchronisation between actions and event watchers is critical. Consider the following scenario, upon receipt of the “Hello !” message, a SuT has to respond with the “Good day !” message within 10s.

Let’s consider the below sequence and check its behaviour:

1. Send the “Hello !” message
2. Watch for a “Good day !” message within the next 10s

If the SuT is very fast and ETIC is busy on another task right after having sent the message, the expected “Hello !” message could be received before the second step of the sequence is executed. In other words, the event watch becomes active after ETIC has received the “Hello !” message. As the event watch will never detect an incoming “Good day !” message, the event watch will fail after the 10s time period.

Therefore, the correct way to define such a sequence is:

1. Non-blocking event watch for a “Good day !” message within the next 10s
2. Send the “Hello !” message

This sequence starts the event watch before running the action that will trigger the event i.e. perform setup before taking action. Of course, it is important here that the watcher be non-blocking in order to perform the action.

As a general rule for sequences: when an ETIC action “immediately” triggers an event from the SuT which is to be monitored by ETIC, the associated non-blocking event watch should be placed just before the action.

7.7 Relays

Within a session, internal links can be made between ETIC CPs by defining them as being part of a relay. A relay allows incoming messages from one CP to be relayed to one or more CPs in the form of outgoing messages. As a result, ETIC can act as a relay between an FDE ICD Part 1 implementation and an FMTP implementation as the headers of incoming and outgoing messages are managed by ETIC (see section 8.13).

Message scripts can also be associated with a relay. User-defined message scripts can be created to process incoming messages for analysis and modification followed by a decision to relay or drop the message.

Note: the existence of a relay will alter the range of target states available to the Communications Partners associated with that relay. For example, the target state of ‘DATA_READY’ is not available to a CP that forms part of a Basic relay so the highest target state available to such a CP is ‘READY’.

7.8 Batch processing

The batch capability of ETIC allows processing commands from a batch file named “etic.in”. This file is searched for in a user defined directory (see section 8.2).

The file is first parsed. Any problem found during parsing is logged into memory. Then, each correctly parsed command is executed in sequence, a new command starting only when the previous one is done. Any problem found at this stage is also logged into memory. Note that the batch processing keeps running if a command cannot be parsed or if it fails at execution time. When all commands have been successfully processed, an empty file named “etic.out” is created to indicate that the batch processing is done. If the processing of certain command(s) fail(s), the “etic.out” file contains a report of the identified problem(s).

The batch processor supports 4 commands:

- Send a message with a predefined CP
- Wait an amount of time
- Log an entry in the event archive
- Exit ETIC

ETIC checks every 100 ms for the presence of a batch file to process.

7.8.1 Batch processing conditions

ETIC starts processing a batch file if and only if the following conditions are met:

- A ‘normal’ session is active (not a Conformance session),
- Batch mode is turned on (see section 8.2),
- The batch directory exists (see section 8.2),
- A file named “etic.in“ is found in the batch directory,
- There is no “etic.out” file found in the batch directory (to prevent overwriting previous results).

In order to use the batch mode, it is recommended to follow the below steps:

- In the batch directory, create a file with a different name than “etic.in”. Edit this file to insert the required commands. Alternatively, copy or FTP this file from somewhere else.
- Rename the file into “etic.in”. Renaming is an atomic operation for operating systems. This ensures that the whole content of the file is available at once under the name “etic.in”[4].
- Wait for “etic.out” to appear.
- Check the content of “etic.out” for possible problems.
- Delete “etic.out”.

7.8.2 Batch file syntax

The “etic.in” file must follow strict syntax rules. Every line is considered as a separate command; only white lines are skipped by the parser. The command keywords are always case insensitive (e.g. "log test” is the same as “LOG test”).

Communication partner names, message content or name need to be enclosed in between two identical delimiters. The delimiter is always the first non blank character of the string. While natural delimiters such as " or ' are best choices, it is possible to use any other character if needed. For instance, all the following strings are equivalent:

```
"hello !"
'hello !'
@hello !@
zHello !z
```

But, `eHello !e` is not valid as the character `e` is contained in the text itself.

Note: In the following sections, delimiters are always identified using double-quotes and keywords are underlined.

7.8.2.1 Comments

A comment does nothing and is simply ignored.

Syntax: #comment

- comment: the skipped text line

Example: # this is a comment line

Note: a command cannot be followed by a comment on the same line. Thus, the following line is invalid:

```
Log "Log this string" # This is not a valid comment
```

7.8.2.2 Send command

Send message content or a predefined message to an existing CP.

Syntax 1: send “local-remote” msg_type “content”
Syntax 2: send “local-remote” predefined “msg name”

- local-remote: the local and remote name of the Communication Partner separated by a minus sign. The CP name is case sensitive.

[4] Copying is not an atomic operation. So, ETIC could see the file while it is still being copied.

- msg_type: the message type can be one of the six following types: raw, operational, operator, identification, system, status. The message type is case insensitive.
- content: the message content. The ^xx notation can be used to represent not printable characters.
- msg_name: the name of a predefined message pattern.

Examples:
```
send "Brussels-Paris" raw "^02^00^00^11^02Raw message."
send "Brussels-Paris" predefined "msg_34"
```

7.8.2.3 Wait command

Pause the batch processor for a given time.

Syntax: wait seconds

- seconds: a number specifying how much seconds to wait.

Example: `wait 5.350`

7.8.2.4 Log command

Log a text in the archiver.

Syntax: log "text"

- text: the text to be logged in the archiver. The originator is "Batch".

Example: `log "This is a marker in the system log to find the start of a batch file"`

7.8.2.5 Exit command

This command closes the ETIC application.

Syntax: exit

8 ETIC User Interface

The following user interface screenshots have been made on a Windows platform. Slight deviations may be noted on other platforms as Swing can be operating system dependent.

8.1 Main screen

At ETIC application launch, the main screen appears. A screenshot of the ETIC main screen is shown in Figure 10.

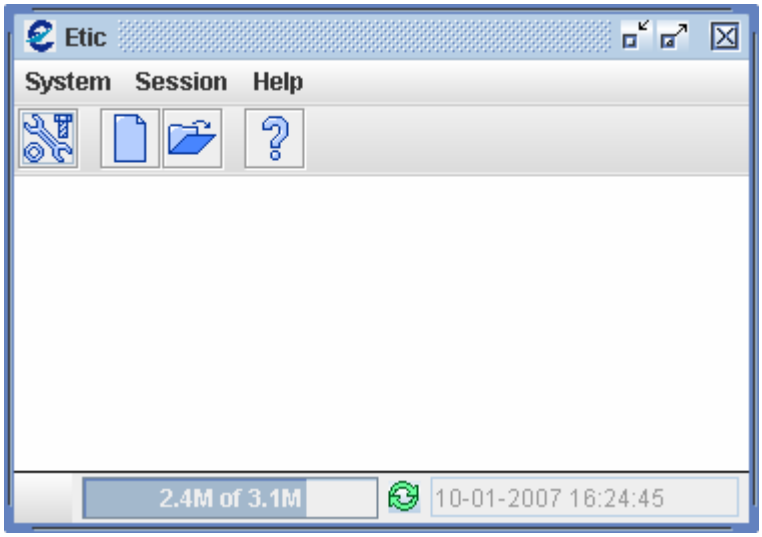


Figure 10: ETIC main screen

8.2 Changing or reviewing the system Settings

ETIC system Settings can be viewed or edited by clicking on the system Settings button  in the toolbar or by using the menu option *System/Settings*. A screenshot of all the system parameters is shown in Figure 11 below and each parameter is described in Table 8 below.

Some of these parameters may be read-only (or hidden). Indeed, all parameters are configurable when there is no active session. But once a session is loaded, the parameters for the log system, the message data section size and the automatic CP start-up cannot be edited. Within a Conformance session, several system parameters are fixed and are thus made invisible to the user.

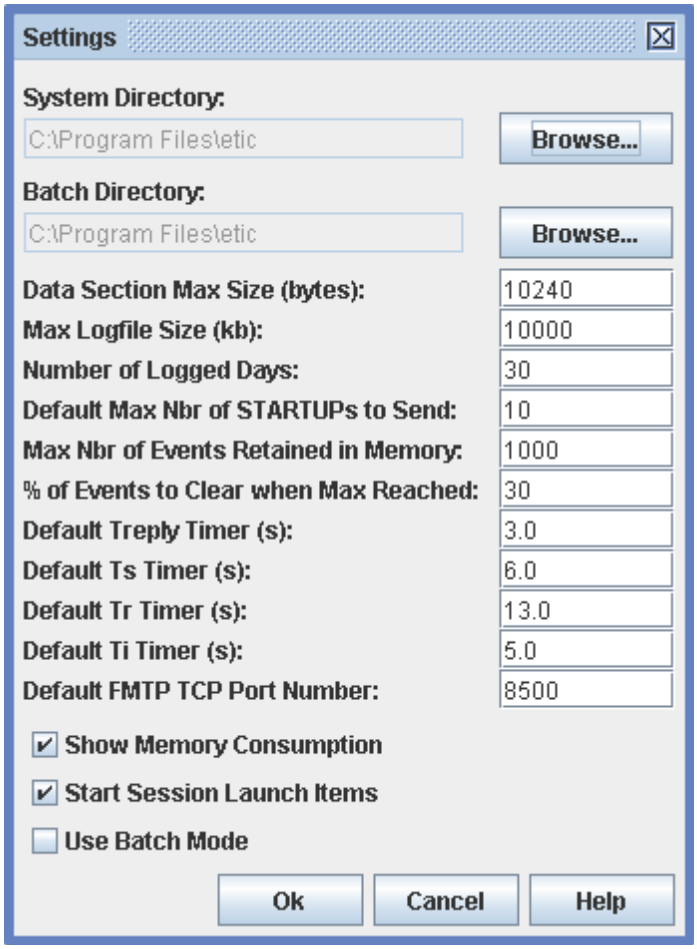


Figure 11: Settings window

Parameter	Description
System Directory	This specifies the directory for all log files.
Batch Directory	This specifies the directory for the batch processing.
Data Section Max Size	Maximum size in bytes of message user data. The default is 10240 bytes according to the FMTP specification (Reference [RD2]).
Max Logfile Size	Maximum size in kilo-bytes of a session log file split into several parts (see Event logging system). The default is 10000 kb.
Number of Logged Days	Maximum age in days of session logs and the system log content after a purge log system (see Event logging system). The default is 30 days.
Default Max Nbr of STARTUPs to Send	Maximum number of STARTUP messages that will be sent by an ETIC CP in order to reach the DATA_READY state (see Communication Partners). The parameter value range is [1 - 2147483647], empty means infinity and the default is 10.
MAX Nbr of Events Retained in Memory	Maximum number of session events retained in memory cache that can be viewed with Monitor Window (see Monitoring events). The default is 1000.
% of Events to Clear when MAX Reached	Percentage of session events that will be cleared from the memory cache once the maximum number of events retained in memory is reached. The oldest sessions events are removed first. The default is 30%.

Default Treply Timer	Default value in seconds for the Treply timer when creating a new session (see Communication Partners). The default value is 3 seconds, the timer range is [0 - 2147483.647] seconds, empty means infinity.
Default Ts Timer	Default value in seconds for the FMTP Ts timer (see Communication Partners). The default value is 6 seconds, the timer range is [0 - 2147483.647] seconds, empty means infinity.
Default Tr Timer	Default value in seconds for the FMTP Tr timer (see Communication Partners). The default value is 13 seconds, the timer range is [0 - 2147483.647] seconds, empty means infinity.
Default Ti Timer	Default value in seconds for the FMTP Ti timer (see Communication Partners). The default value is 5 seconds, the timer range is [0 - 2147483.647] seconds, empty means infinity.
Default FMTP TCP Port Number	Default port number for FMTP communications (see Communication Partners). The default value is 8500 and the port range is [1 - 65535].
Show Memory Consumption	Enable/disable the display of the Java memory consumption in the status bar below the ETIC main screen.
Start Session Launch Items	Enable/disable the ability to start CPs automatically when opening a session (see the 'Target At Launch' parameter under Communication Partners) and Relays (see the 'Enable at Launch' parameter under Relays).
Use Batch Mode	Enable/disable the engine that processes batch files.

Table 8: Settings parameters

8.3 Checking available network devices

The Info window is accessed via the menu option *System/Information*. It displays operational system information relating to the ETIC platform:

- The ETIC version
- The start date and time
- The JVM and OS features
- The availability of X.25
- The available IP network interfaces

Figure 12: System Information window

8.4 Event logging system

The ETIC log system consists of two completely different sets of files that can be found in the system directory specified in the System Settings (section 8.2, Changing or reviewing the system Settings). All events that occur during an active session are stored in both sets to allow easy future review.

Firstly, the session log files contain all the events of a specific session in an XML based representation. These log files (except the log file currently being assembled by ETIC) can be reviewed by opening them with the View Log window (section 8.5, Reviewing event log files) or Monitor Window (section 8.14, Monitoring events). The use of XML allows for easy retrieval of certain events and the use of the filtering capabilities provided by View Log window. By default, in order to avoid memory issues, session log files are automatically split into parts once they reach a size of 10000 kb (section 8.2, Changing or reviewing the system Settings). As a result, all session log files have the following filename format: `"Session_YYMMDD_HHMMSS_NN.log"` where YYMMDD is the date, HHMMSS is the session start time and NN is the part number.

Secondly, there is a unique system log file named "ETIC_log.txt" that contains all the events in a text based format. Events are simply appended to the file every time a new session is created or opened and a 'new session' marker is inserted. By default, the system log only contains events that are displayed in monitor windows and not session context items such as CP definition. Therefore, the menu option *Session/Export to System Log* makes it possible to export the complete session context to the system log. The advantages of the system log are that it is more readable with a simple external text viewer, it contains the logs of several sessions and occupies less hard disk space than its XML counterparts. The disadvantage is that it cannot be opened inside ETIC and make use of its filtering capabilities.

As both log types will keep on consuming hard disk space indefinitely, it can be purged at any given time by using the menu option *System/Purge Log System*. This will result in ETIC

deleting the session log files and truncating the textual system log on the basis of the number of logged days system parameter (see section 8.2, Changing or reviewing the system Settings).

8.5 Reviewing event log files

Any existing (XML-based) log file, except the log file currently being assembled by ETIC, can be viewed at any time by using the menu option *System/View Log* and then clicking on the 'open' icon button to find, select and open the specific log file that is to be examined. Such a window is shown in Figure 13 and it behaves exactly the same as a Monitor window except that functionality relating to live data is no longer available. For an explanation of the available actions (see section 8.14, Monitoring events).

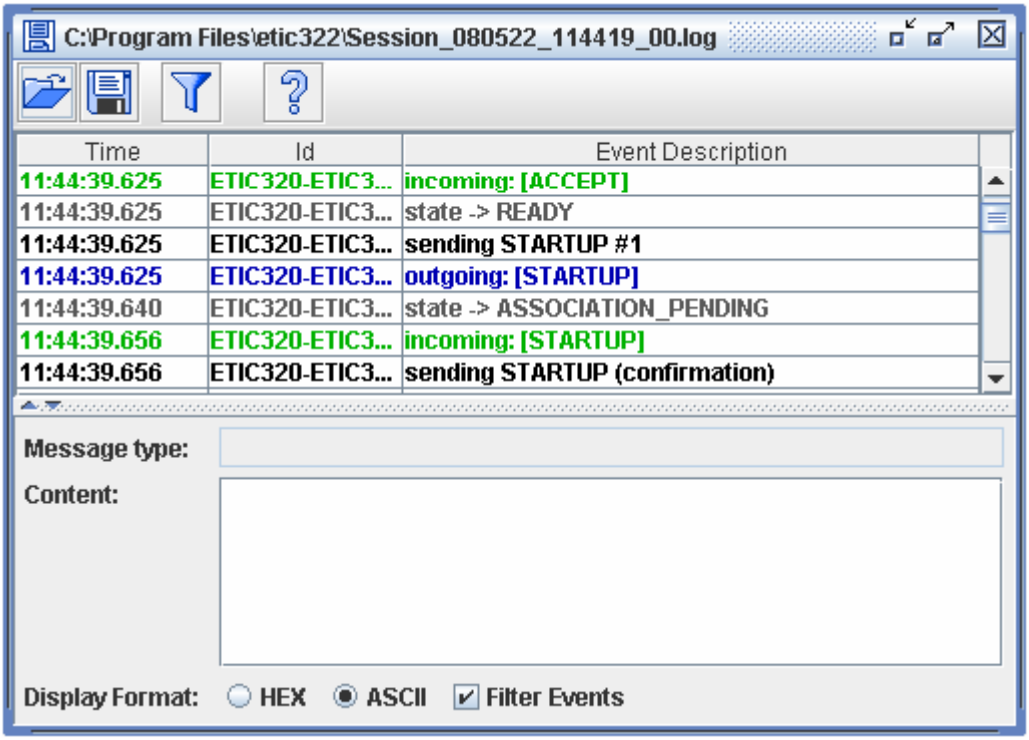


Figure 13: View log window

8.6 Sessions

Sessions can be viewed as ETIC user application files. They contain all the data objects needed to perform tests. Typically, the user starts from a new empty session and adds all the CPs, messages, smart responses and sequences as required.

8.6.1 Open a new session

Use the menu option *Session/New* or the toolbar button  . If this is done when another open session has been modified but not saved, the user will be asked if the changes are to be saved.

8.6.2 Open a previously saved session

From the menu option *Session* select an item from the 'session history list' or use the menu option *Session/Open* or use the toolbar button  . If this is done when another open

session has been modified but not saved, the user will be asked if the changes are to be saved.

8.6.3 Session tree

Use the toolbar button  to display the session tree window. The session tree window provides a hierarchical representation of the session file to browse and easily edit existing session entities.

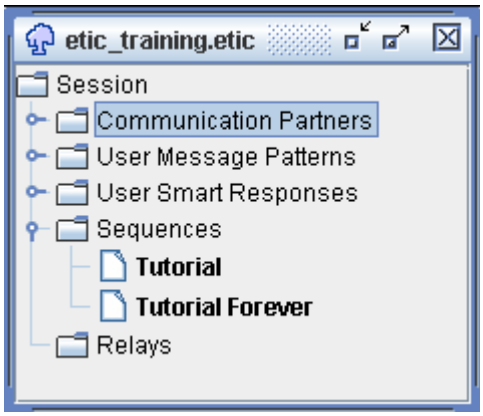


Figure 14: Session tree window

Double click on any item shown in the Session Tree window to open a window that gives access to all the details of that item.

8.6.4 Saving a session

Use the menu option *Session/Save* or *Session/Save as…* or the toolbar button  to display the "Save" window as shown in Figure 15. The session file is saved in pure XML format, except if the session is locked or if it is a conformance session. In these two cases, the content of the file is encrypted to avoid user-tampering. (See sections 8.7 Locked sessions and 8.8 Conformance session).

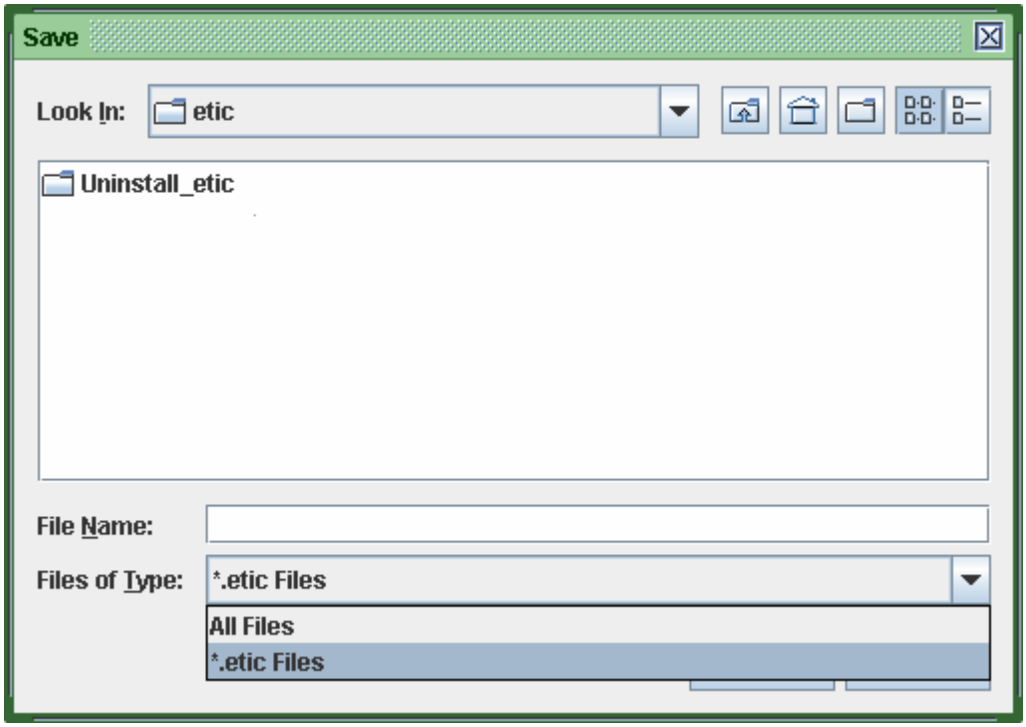


Figure 15: Save session window

If the “Files of Type” drop down menu in the “Save” window is set to “*.etic files”, ETIC will automatically append the .etic extension to the file name chosen when saving the file, and files displayed in the “Save” window will also be filtered on the .etic extension. If the “Files of Type” drop down menu is set to “All Files”, ETIC will not append any extension nor filter the files displayed.

8.6.5 Closing a session

Use the menu option *Session/Close* or the toolbar button  . If the session was modified but not saved, the user is asked if the changes are to be saved.

8.6.6 Export to system log

Use the menu option *Session/Export to System Log*. This will export all the defined session objects to the textual system log file (see section 8.4 Event logging system). With this exported data the session can be rebuilt by copy-pasting the XML data into a new file with the .etic extension.

8.6.7 Export to HTML

Use the menu option Session/Export to HTML. This exports all the defined session objects to a human readable hypertext markup language (HTML) file.

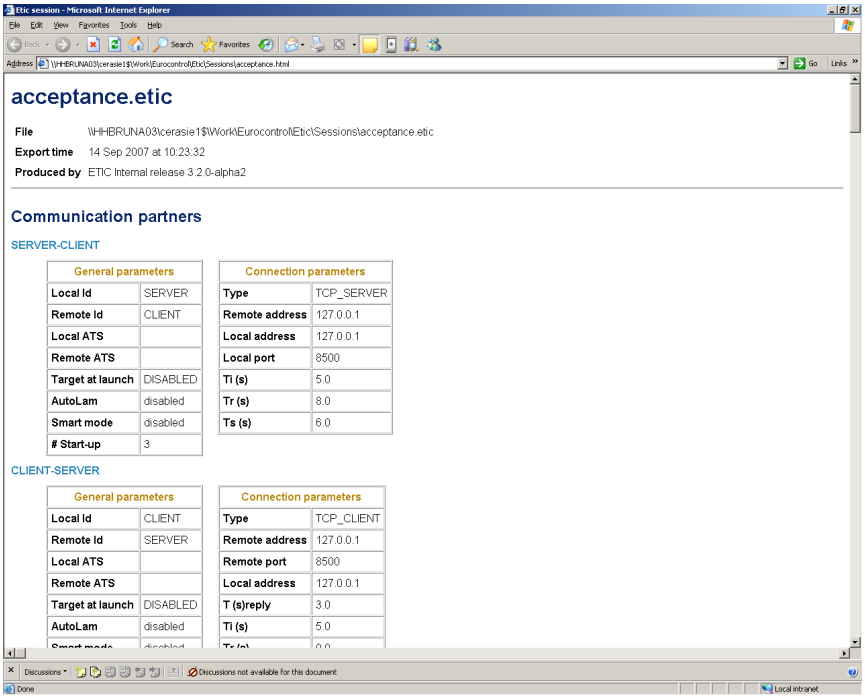


Figure 16: HTML Export

8.7 Locked sessions

A locked session is a normal session that has been locked with a password. This feature has been introduced to allow the distribution of a carefully prepared session without the risk of it being largely re-edited by the final user or operator. When a session is locked:

- CPs definitions can be viewed and edited (except for the locked parameters: Target At Launch, autoLAM, Smart and connection type) but adding or deleting is not allowed.
- Messages and Smart Response definitions can be viewed but adding, modifying or deleting is not allowed.
- Sequences can be viewed and run but adding, modifying or deleting is not allowed.
- Relays can be viewed and enabled but adding, modifying or deleting is not allowed.

To avoid overriding these restrictions by directly editing the XML session file, a locked session is encrypted upon saving.

8.8 Conformance session

The conformance session is a dedicated session for running an FMTP conformance test.

Before starting a conformance session, the user should read section 9 of this User Guide and ensure that the System_under_Test is configured as described in sections 9.2 / 9.3 of this User Guide.

The conformance session is started by using the menu option *System/Conformance*. This opens the Conformance Test window as shown in Figure 17. When this window is closed, the session is also automatically closed.

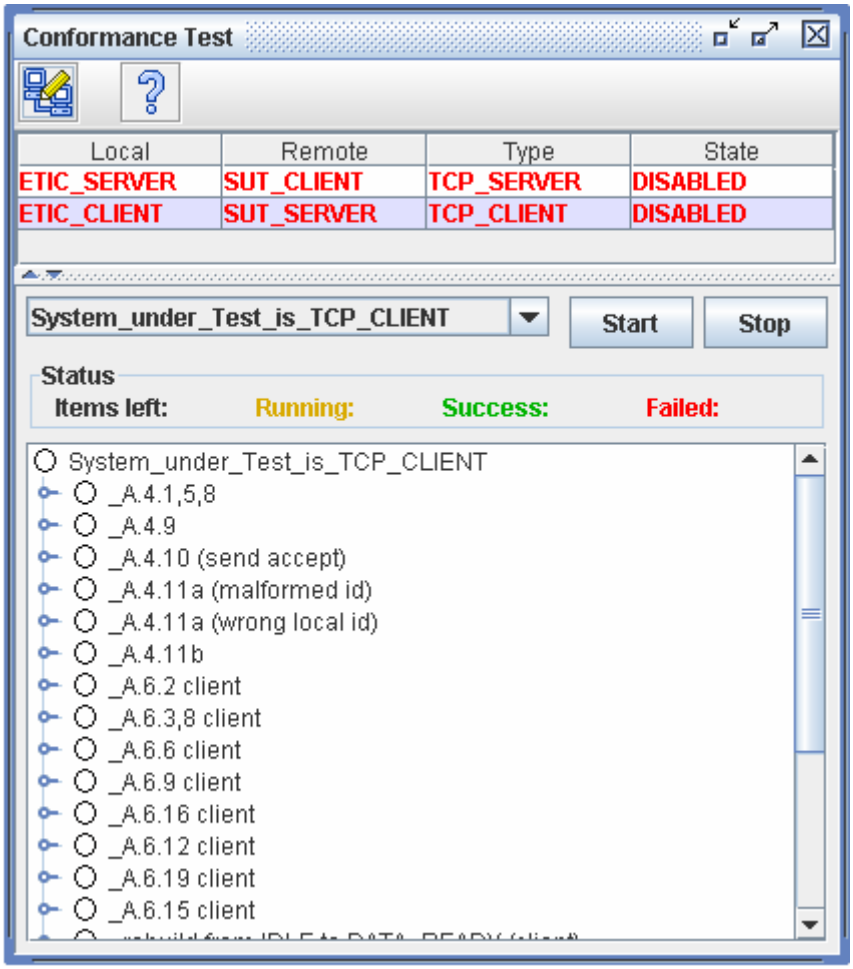


Figure 17: Conformance Test window

For a conformance session, the only required user input is the configuration of the CPs that are to be used in the tests, either by selecting the CP and using the edit button  , or by double clicking on the line entry representing the CP. This opens a CP parameter window as described in section 8.9.2, Editing a Communication Partner, with restricted editing. Indeed, only the remote and local IP addresses can be edited. Once the CP is configured, a conformance test is run by selecting the appropriate item from the dropdown menu and clicking on the start button. During the conformance test, special care must be taken to properly execute the actions defined in the pop-up windows.

If the test succeeds, a conformance certificate file is generated and a popup window appears informing the user where it is located. These certificate files have the following filename format: `"Certificate_TESTNAME_YYMMDD_HHMMSS.crt"` where TESTNAME is the name of the conformance test, YYMMDD is the test date and HHMMSS is the test time. If a conformance test fails, the event log (see section 8.14, Monitoring events) can be consulted to provide a diagnosis.

Just like any other kind of session, conformance sessions can be saved and reloaded.

8.9 Communication Partners

Within a session, CPs are defined and managed with the Communication Partner window as shown in Figure 18. This window can be opened by using the menu option

View/Communication Partners or the toolbar button  . The top of the CP window is made of user friendly buttons and message counters for the selected CP. The CP parameter descriptions are detailed in Table 9 below.

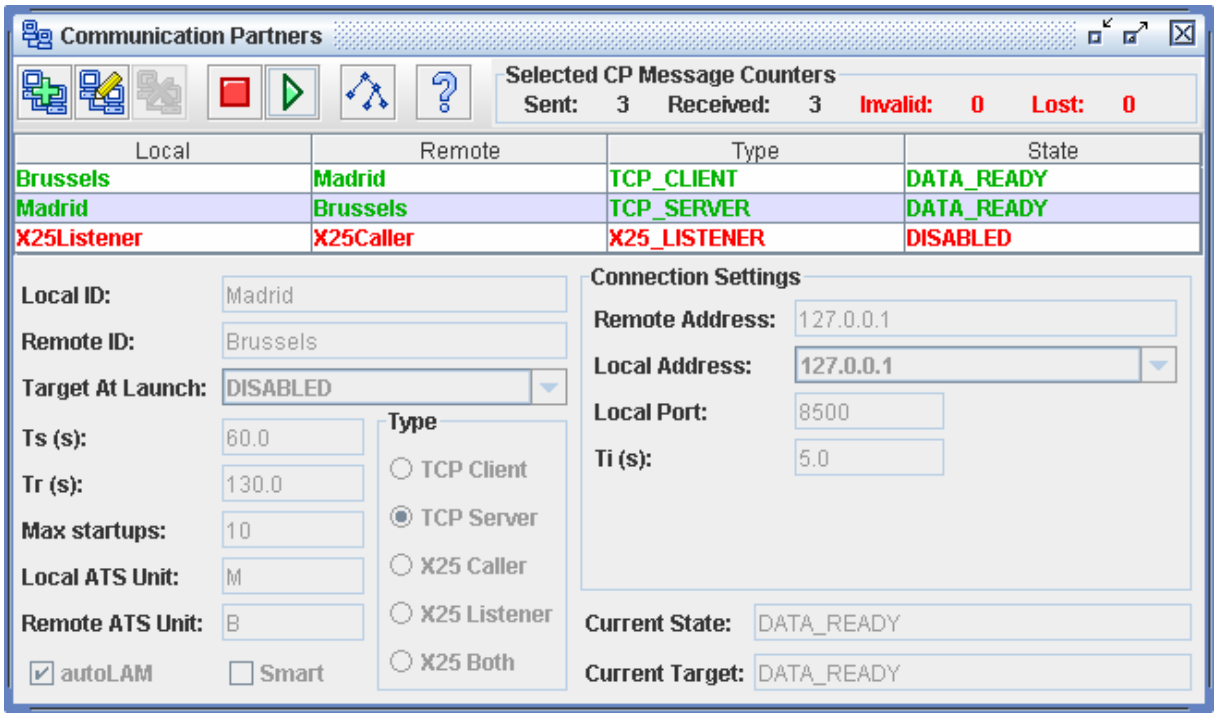


Figure 18: Communication Partners window

Parameter	Description
Local ID or Local Name	A CP's Local ID value corresponds to the FMTP identification value which is used in the identification message. In message patterns it is referred by the $l macro. The Local ID value can be up to 32 printable ASCII characters except the hyphen character and is case sensitive. The Local Name is used to name X.25 CPs. For ease of implementation it has the same syntax as the Local ID.
Remote ID or Remote Name	The Remote ID value corresponds to the FMTP identification value of the SuT which is used to validate incoming identification messages from the SuT. In message patterns it is referred by the $r macro. The Remote ID value can be up to 32 printable ASCII characters except the hyphen character and is case sensitive. The Remote Name is used to name an X.25 SuT. For ease of implementation it has the same syntax as the Remote ID.
Target At Launch	This dropdown menu allows automating the setting of CP's target for when the session is loaded. This feature can be enabled or disabled in the System Settings window. A target at launch can be DISABLED, READY or DATA_READY with the default set to DISABLED.
Ts	The FMTP or FDE ICD Part 1 Ts protocol timer. In the absence of user data being sent, this timer conditions the sending of HEARTBEAT messages. The timer range is [0 - 2147483.647] seconds, empty means infinity. The default value is defined by the System Settings window.
Tr	The FMTP or FDE ICD Part 1 protocol timer Tr. In ASSOCIATION_PENDING this timer conditions the sending of STARTUP messages. In DATA_READY the Tr timer triggers if no HEARTBEAT or operational data messages have been received within the Tr time period. It will close the connection (FMTP) or force a

	return to ASSOCIATION_PENDING (FDE ICD Part 1). The timer range is [0 - 2147483.647] seconds, empty means infinity. The default value is defined by the System Settings window.
Max startups	Maximum number of STARTUP messages that will be sent by an ETIC CP from the READY or ASSOCIATION_PENDING state to try and reach DATA_READY. This is an ETIC implementation specific parameter to avoid an ETIC CP sending STARTUPs forever. Once the maximum number is reached ETIC will automatically invoke the MT-Stop service, sending a SHUTDOWN message and returning to READY. The parameter value range is [1 - 2147483647], empty means infinity. The default value is defined by the System Settings window.
Local ATS Unit	The Local ATS Unit is part of the OLDI operational message (Field 3) and is used to identify the CP in the operational context. The setting of this parameter is optional. However, it is must be defined in order to enable the autoLAM feature. In the message patterns the Local ATS Unit is referred by the $L macro. The Local ATS Unit can be up to 8 upper-case characters of the range [A-Z].
Remote ATS Unit	The Remote ATS Unit is part of the OLDI operational message (Field 3) and is used to identify the SuT in the operational context. The setting of this parameter is optional. However, it is must be defined in order to enable the autoLAM feature. In the message patterns the Remote ATS Unit is referred by the $R macro. The Local ATS Unit can be up to 8 upper-case characters of the range [A-Z].
autoLAM	This checkbox enables or disables the autoLAM mechanism for the CP. When enabled, ETIC will automatically respond with a LAM message to any incoming “LAM-able” operational message that is received from the SuT. Local ATS Unit and Remote ATS Unit parameters must be defined to use the autoLAM feature.
Smart	This checkbox enables or disables the Smart Response mechanism for the CP. When enabled, every incoming message is checked against the list of enabled Smart Responses defined in the session. When one or more matches are found, ETIC sends the associated outgoing messages.
Type	This radio button defines the CP type: FMTP TCP server, FMTP TCP client, FDE ICD Part 1 X.25 caller, FDE ICD Part 1 X.25 listener, or FDE ICD Part 1 X.25 both.
Connection Settings	The connection settings vary according to the CP type. It allows the setting of network interface parameters further detailed in Table 10.
Current State	This indicates the current state of the CP.
Target State	This indicates the target state assigned to the CP. As long as the current and target states differ, ETIC will be performing actions to reach the target state.

Table 9: Communication Partner parameters

Type	Parameter	Description
TCP Client	Remote Address	The IP address of the SuT (behaving as FMTP TCP server). This needs to be a valid IPv4 or IPv6 address. It can also be name if it can be resolved. The IP version must match that of the Local Address.
	Remote Port	The TCP port of the SuT (behaving as TCP server). The port range is [1 - 65535] and the default value is defined by the System Settings window.
	Local Address	A dropdown menu listing a selection of valid local IP address detected by ETIC.
	Treply	This defines the time delay between two successive connection attempts. The timer range is [0 - 2147483.647] seconds, empty means infinity and the default value is defined by the System Settings window.

	Ti	This is the FMTP protocol Ti timer. It defines the maximal delay to receive the SuT's identification message once the CP's identification message has been sent. The timer range is [0 - 2147483.647] seconds, empty means infinity and the default value is defined by the System Settings window.
TCP Server	Remote Address	IP address of the SuT (behaving as TCP client). This needs to be a valid IPv4 or IPv6 address or a resolvable network name. It can also be a regular expression using the wildcards '?' and '*'. For IPv4, '?' means 'a single digit in the range [0-9]' while '*' means 'any number in the range [0-255]'. An IPv6 address is normally written as eight groups (or 'nibbles') of four hexadecimal digits where each group is separated by a colon ':' although consecutive groups containing '0000' can be replaced by '::' (but only one '::' can appear in an address). A simple example could be '30da:f03e:0000:0000:2a11:0101:05ce:ff01' which could also be written as '30da:f03e::2a11:0101:05ce:ff01'. Note also that the final two characters in an IPv6 address used by ETIC (i.e. '%n') are not part of the standard IPv6 address. They are used simply to identify the interface index. For IPv6, '?' means 'a single hexadecimal in the range [0-F]' while '*' means 'between 1 and 32 hexadecimals with each hexadecimal being in the range [0-F]'. Note that '32' is the maximum number of characters (hexadecimals) that can appear in an IPv6 address (eight 'nibbles' separated by colons with a maximum of four characters per 'nibble'). The IP version must match that of the Local Address.
	Local Address	A dropdown menu listing a selection of valid local IP address detected by ETIC.
	Local Port	Local TCP port used by ETIC. The port range is [1 - 65535] and the default value is defined by the System Settings window.
	Ti	This is the FMTP protocol Ti timer. It defines the maximal delay to receive the SuT's identification message once a TCP connection has established. It also defines the maximal delay to receive the SuT's ACCEPT message once the CP's identification message has been sent. The timer range is [0 - 2147483.647] seconds, empty means infinity and the default value is defined by the System Settings window.
X.25 Caller (or both mode)	Local caller DTE	This is an X.121 address which can be up to 15 digits and is used by ETIC when the CP makes a call. When this field is empty a local address is not inserted in the call request packet.
	Remote listener DTE	This is an X.121 address which can be up to 15 digits and is used to define the address of the SuT (behaving as a X.25 listener). This field cannot be empty.
	Treply	This defines the time delay between two successive connection attempts. The timer range is [0 - 2147483.647] seconds, empty means infinity and the default value is defined by the System Settings window.
	Caller UDF	When this checkbox is enabled, data is sent in the User Data Field (UDF) of the X.25 connection request packet when establishing the call. The UDF content is defined by using the Set button next to it. When this checkbox is disabled no UDF is present in the X.25 connection request packet.

X.25 Listener (or both)	Local listener DTE	This is the local X.121 address on which ETIC listens for incoming calls from the SuT. It can be empty or up to 15 digits but wildcards '*' and '?' are also allowed. '*' means void or any number in the range [0-999999999999999] while '?' means a single digit in the range [0-9].
	Remote caller DTE	This is the X.121 address from which ETIC expects the SuT call. It can be empty or up to 15 digits but wildcards '*' and '?' are also allowed. '*' means void or any number in the range [0-999999999999999] while '?' means a single digit in the range [0-9].
	Listener UDF	When this checkbox is enabled, ETIC expects data in the User Data Field (UDF) of the X.25 connection request packet when the SuT calls. The UDF content must match the ETIC UDF which is defined by using the Set button. When this checkbox is disabled any incoming X.25 connection request packet with a UDF field is ignored.
X.25	X25 Port	This is the port number referencing the EICON X.25 Card. According to DIALOGIC Eiconcard documentation there can up to 48 port numbers.

Table 10: Connection Settings parameters

8.9.1 Adding a Communication Partner

A CP is created by using the toolbar button  . This opens the window shown in Figure 19 below. All default values defined in the System Settings are already inserted and can be edited. Refer to Table 9 and Table 10 for a detailed description of all parameters. The autoLAM checkbox will remain hidden until both Local ATS Unit and Remote ATS Unit fields are set.

When working with a locked or conformance session, CPs cannot be added.

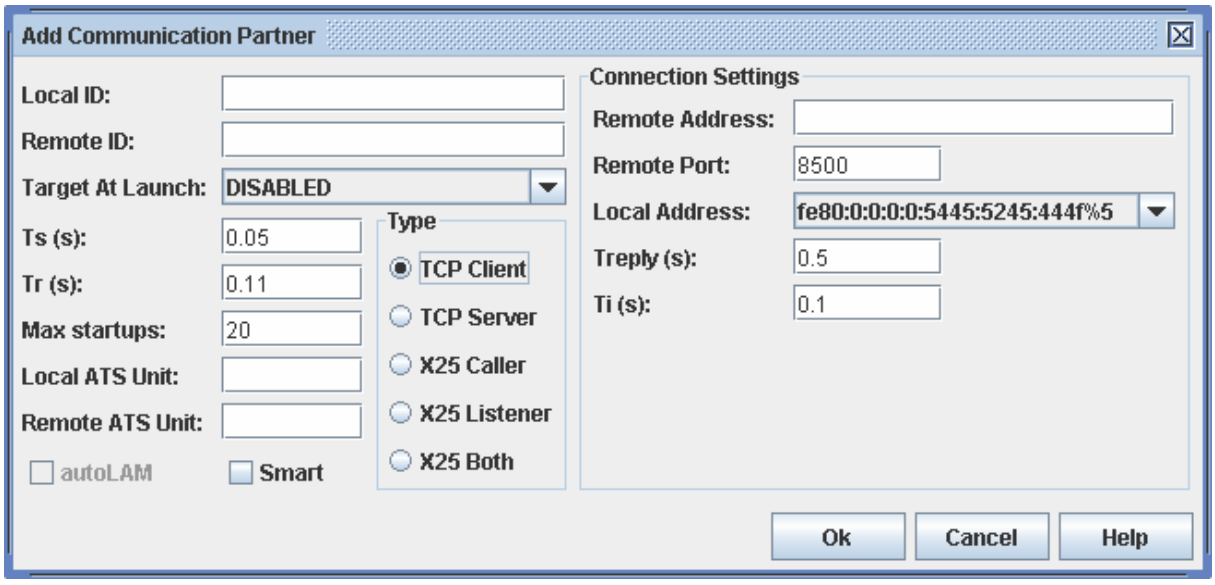


Figure 19: Add Communication Partner window

8.9.2 Modifying a Communication Partner

A CP can be modified by using the toolbar button  , by double clicking on the CP in the CP window or by pressing the keyboard space bar when it is selected in the CP window. This

opens the window shown in Figure 19 below. Refer to Table 9 and Table 10 for a detailed description of all parameters. The autoLAM checkbox will remain hidden until both Local ATS Unit and Remote ATS Unit fields are set.

It will not be possible to modify certain parameters when the CP is not in the DISABLED state or when working with a locked and conformance sessions.

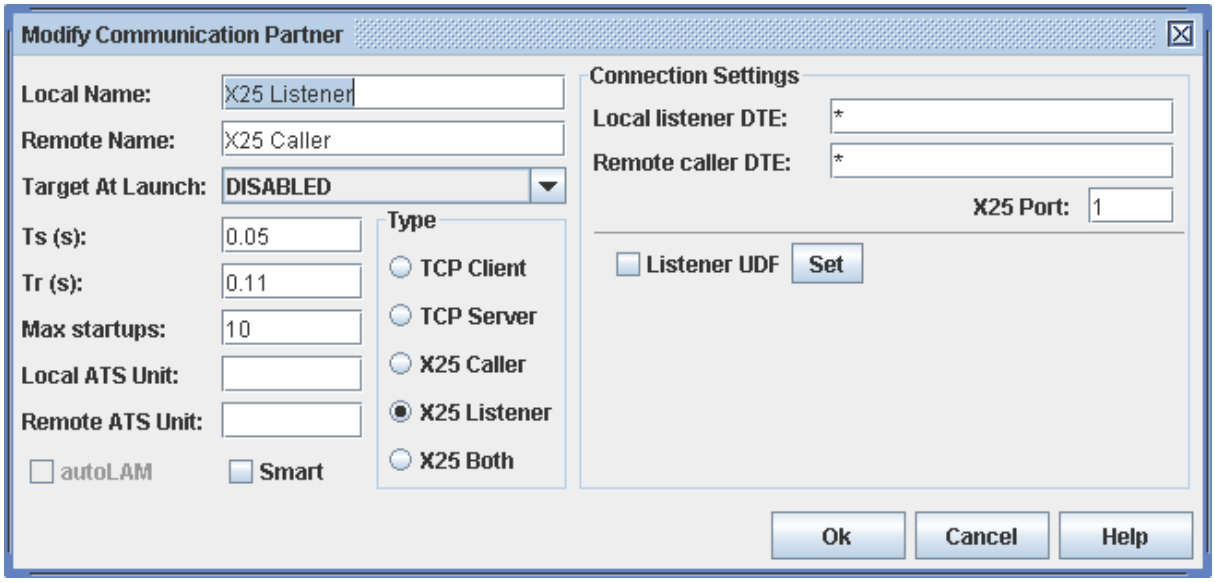


Figure 20: Modify Communication Partner window

8.9.3 Deleting a Communication Partner

CPs can be deleted from the session by using the toolbar button  or by selecting the CPs and pressing the keyboard delete key. A confirmation window appears to prevent accidental deletion.

Multiple CPs can be selected in different ways using standard methods:

- by pressing and holding the left mouse button, and then dragging the mouse over the CPs to be selected
- by selecting the first CP and then selecting the last CP while holding the shift button down (this will select all CPs between the first and last CP)
- By selecting individual CPs while holding the control key down

CPs can not be removed from a locked or conformance session. Furthermore, a CP may not be deleted if other ETIC entities such as sequences or relays depend on it (see section 8.9.5, Check dependencies of a Communication Partner).

8.9.4 Setting the target state of a Communication Partner

A CP becomes active once a target state has been assigned by using the toolbar button  . This causes a dropdown menu to appear with all the reachable states. The CP will then attempt to reach this state according to the FMTP or FDE ICD Part 1 protocol. The CP's state can be quickly changed to DISABLED by using the toolbar button  .

A single target state can be assigned to multiple CPs in one operation by selecting the desired CPs using the method described in section 8.9.3 and then using the  or  buttons.

If the CP belongs to a relay, the list of available target states will be restricted. The target state of 'DATA_READY' is not available to a CP that forms part of a basic relay so the highest target state available to such a CP is 'READY'.

8.9.5 Check dependencies of a Communication Partner

To determine which sequences are using the selected CP, the CP window dependencies button  is to be used. As shown in Figure 21 below, this displays a window containing the names of all the sequences and relays that are using the CP. A CP can not be deleted if it is being used by a sequence or a relay.

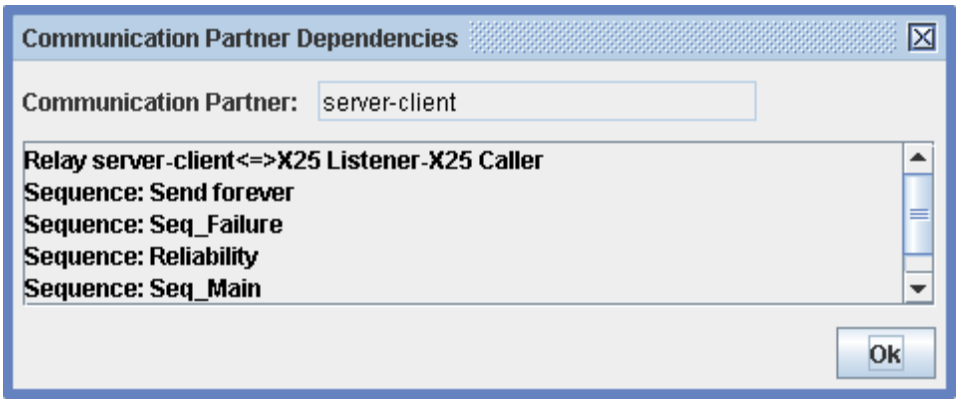


Figure 21: Communication Partner Dependencies window

8.10 Messages

Within a session, message patterns are defined and managed with the Messages window as shown in Figure 22 below. This window is opened through the menu option *View/Messages* or the toolbar button  . Some message patterns are predefined and cannot be removed or edited. They are used for the needs of the FMTP/FDE ICD Part 1 protocols and for the auto-LAM feature. Predefined messages are displayed in blue, user defined messages are displayed in black.

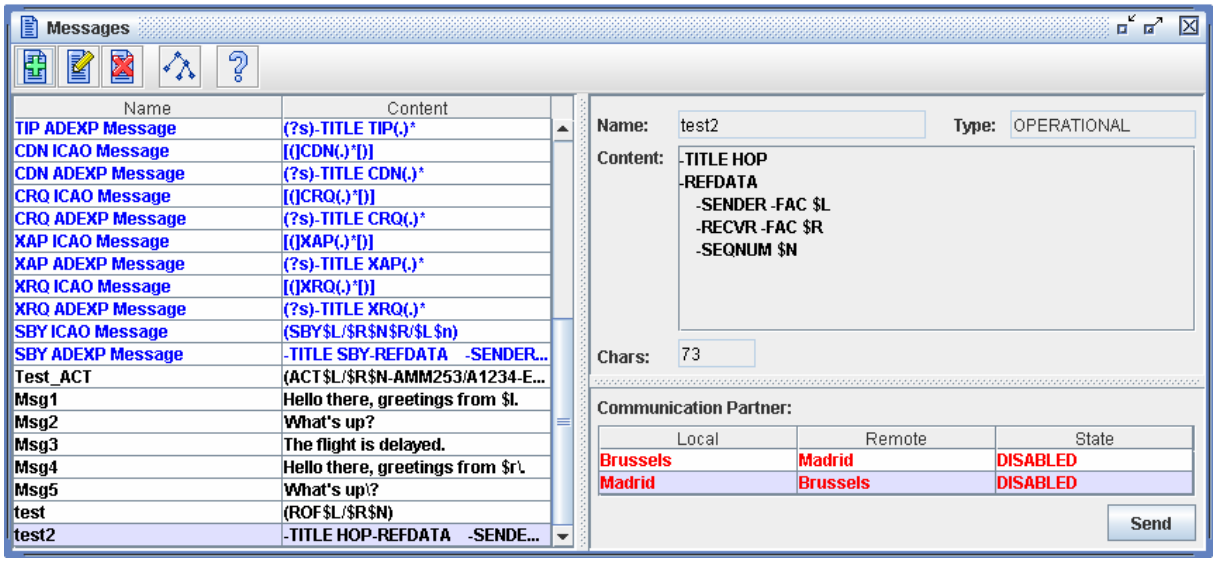


Figure 22: Messages window

8.10.1 Adding a message

A message is created by using the toolbar button  . This opens the window shown in Figure 23 below.

Messages are defined as follows:

1. Enter any name (alphanumeric text string) into the ‘Name’ field.
2. Select the type of message you want to define (e.g. ‘ABI’) using the drop down list.
3. Enter text and/or values into all the active fields that are located inside the ‘Operational Message Builder’ section of the window. Note that many of these values must be entered using drop down lists.
4. Press the ‘Build ICAO’ button or the ‘Build Adexp’ button. This will construct an ICAO or Adexp format message containing the values entered above. ETIC will present the correctly formatted message in the ‘Content’ field.
5. Edit (but only if necessary) the formatted message in the ‘Content’ field. Note that further manual editing of this field is usually not necessary.
6. Press ‘OK’ to save the defined message. The message will be saved under the name entered in step 1 above.

Note that:
The ‘Reset’ button clears all the fields in this window EXCEPT for the ‘Content’ field.
The ‘Clear’ button clears ONLY the ‘Content’ field.

A message consists of a unique name, a message type (IDENTIFICATION, OPERATIONAL, OPERATOR, RAW, SYSTEM or STATUS) and it’s content. With the exception of the RAW type, an FMTP or FDE ICD Part 1 header is pre-pended to the message. The RAW message type is primarily available to allow users define their own header.

Within the content field both printable and non-printable ASCII characters can be entered using the hexadecimal notation ^##, where “##” accepts any hexadecimal value in the range [00-FF]. The seven macros described in section 7.4, Message patterns: macros and regular expressions can also be inserted. These macros will be substituted by their values, when the message pattern is converted into a real message for transmission.

To facilitate the creation of OLDI operational messages in ICAO Document 4444 (Reference [RD5]) or ADEXP (Reference [RD6]) format, an Operational Message Builder is provided. Although the current pre-defined operational message list and field descriptions are based on OLDI (Reference [RD4]) the NEW type can be used to create any type of message content. According to the OLDI specification, the Operational Message Builder will indicate whether mandatory fields are missing but the message can still be built. As a final means of edition, the built message can be fine-tuned by directly editing the 'Content' field.

Message patterns can not be created when working with a locked or conformance session.

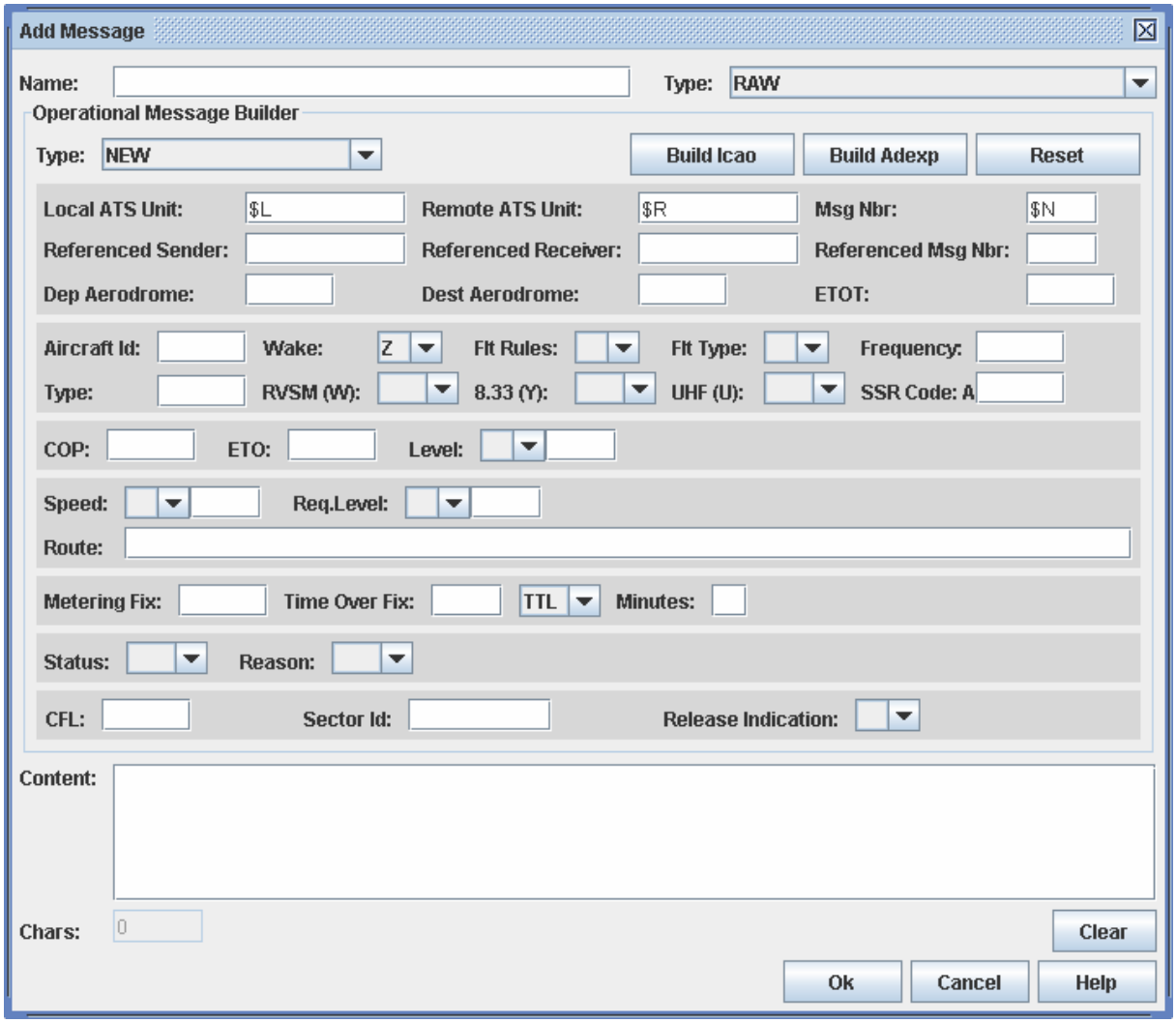


Figure 23: Add Message window

8.10.2 Modifying a message

A message pattern can be modified by using the toolbar button  , by double clicking on the message in the Message window or by pressing the keyboard space bar when it is selected in the Message window. This opens the window shown in Figure 24 below.

ETIC does not interpret OLDI messages; therefore it does not display the message content in a panel similar to the Operational Message Builder.

Within the content field both printable and non-printable ASCII characters can be entered using the hexadecimal notation ^##, where "##" accepts any hexadecimal value in the range [00-FF]. The seven macros described in section 7.4, Message patterns: macros and regular expressions can also be inserted. These macros will be substituted by their values, when the message pattern is converted into a real message to be sent.

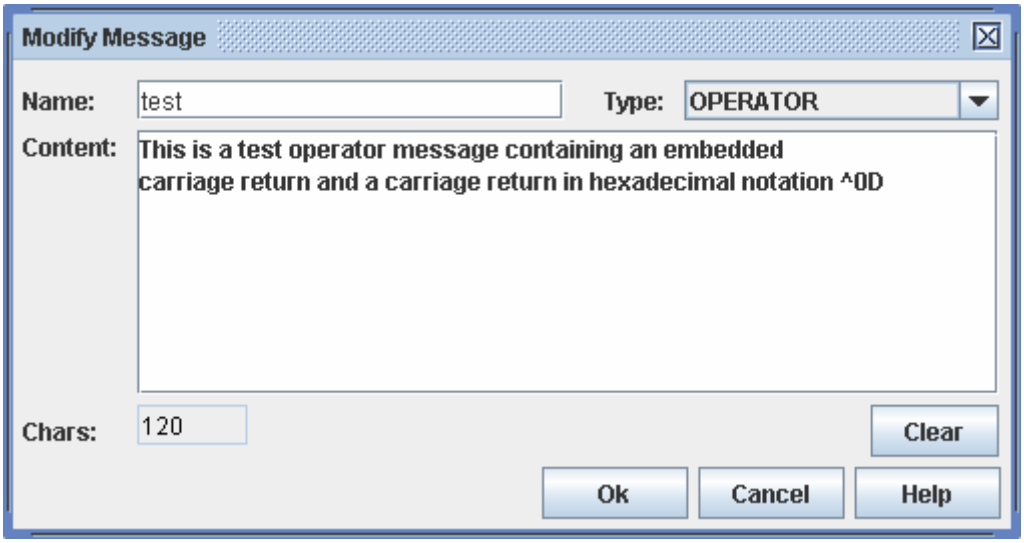


Figure 24: Modify Message window

8.10.3 Deleting a message

A message pattern can be deleted from the session by using the toolbar button  or by selecting the message in the Message window and pressing the keyboard delete key. A confirmation window will appear to prevent accidental deletion.

Message patterns cannot be removed from a locked or conformance session. Furthermore, a message may not be deleted if other ETIC entities such as a sequence depend on it (see section 8.10.4, Check dependencies of a message).

8.10.4 Sending a message

Within a session, a message pattern can be sent to the SuT from the appropriate CP by selecting the Send button within the Messages windows. When multiple messages and CPs are selected at the same time, ETIC will send the first message to all selected CPs, then the second message to all selected CPs etc.

8.10.5 Check dependencies of a message

To determine which sequences or smart responses are using the selected message, the toolbar button  is to be used. As shown in Figure 25 below, this displays a window containing the names of all the sequences or smart responses that are using it. A message pattern cannot be deleted if it is used by a sequence or a smart response.

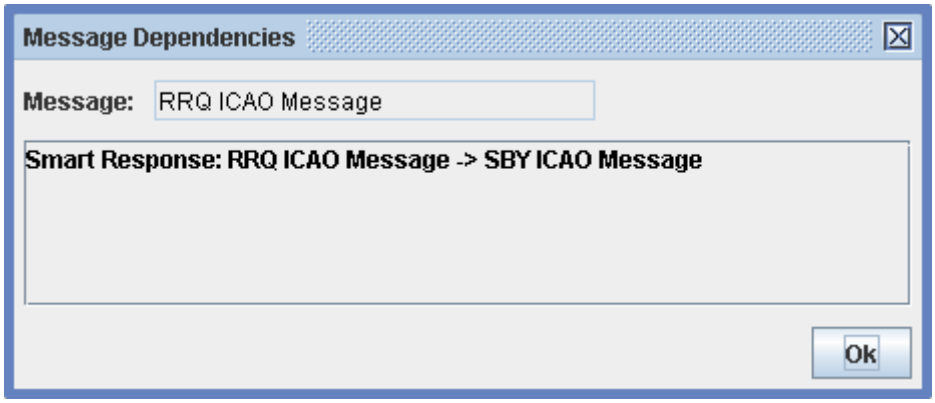


Figure 25: Message Dependencies window

8.11 Smart responses

A smart response is an association between two message patterns, one is defined as being the incoming message while the other as the outgoing message. When a smart response enabled CP receives a message from the SuT, it checks the incoming message pattern against all enabled smart responses for matches. Whenever a match is found, ETIC responds to the SuT with the outgoing message. Depending on the smart response definitions there may be more than one match leading to several smart responses.

Within a session, smart responses are defined and managed with the Smart Response window as shown in Figure 26 below. This window is opened by using the menu option *View/Smart Responses* or the toolbar button  . Each specific smart response can be enabled or disabled by using the corresponding checkbox. Disabled smart responses are displayed in grey. Enabled pre-defined smart responses are displayed in blue and enabled user-defined smart responses are displayed in black. The message pattern content is displayed in the IN Content and OUT Content fields.

For each smart response, a response delay can be specified to ensure ETIC sends the response after a given number of seconds. As more than one smart response definitions may match an incoming message, the use of the stop checkbox and the  arrows for smart response ordering can be combined to meet the desired smart response behaviour.

To summarise, the following six pre-requisites are necessary for an ETIC CP to send message ‘B’ as a smart response to incoming message ‘A’ from a System under Test:

- a CP is defined in ETIC to enable two way communications with the SuT
- the CP is ‘Smart Response Enabled’ in the ‘Modify Communication Partners’ window
- message pattern ‘A’ is defined in ETIC
- message pattern ‘B’ is defined in ETIC
- a Smart Response is defined and enabled in ETIC where ‘A’ is the ‘In’ message and ‘B’ is the ‘Out’ message in the ‘Smart Response’ window
- the SuT sends message ‘A’

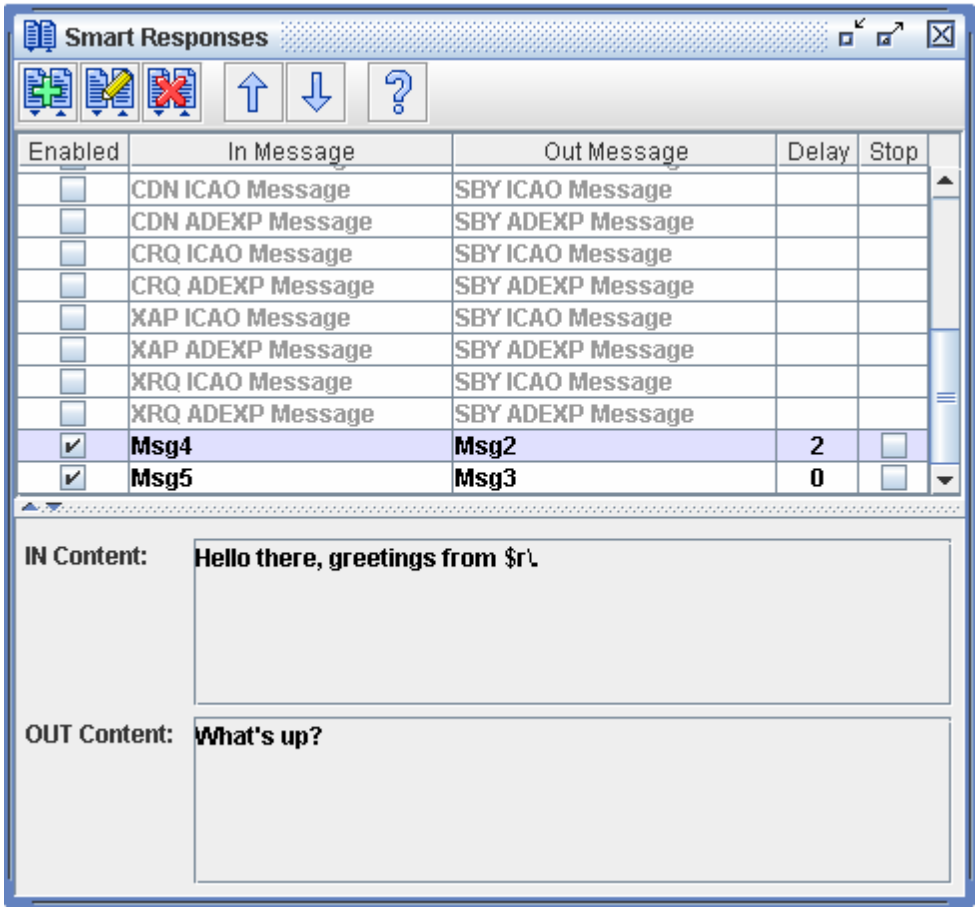


Figure 26: Smart Responses window

8.11.1 Adding a smart response

A smart response is created by using the toolbar button  . This opens the window shown in Figure 27 below. By default, newly created smart responses are enabled.

A smart response is defined by associating an incoming message with an outgoing message. The smart response can be further characterised by its status (enabled/disabled), a time to delay the transmission of the smart response, and whether ETIC should stop or continue searching for additional smart response matches.

Smart responses cannot be added to a locked or conformance session.

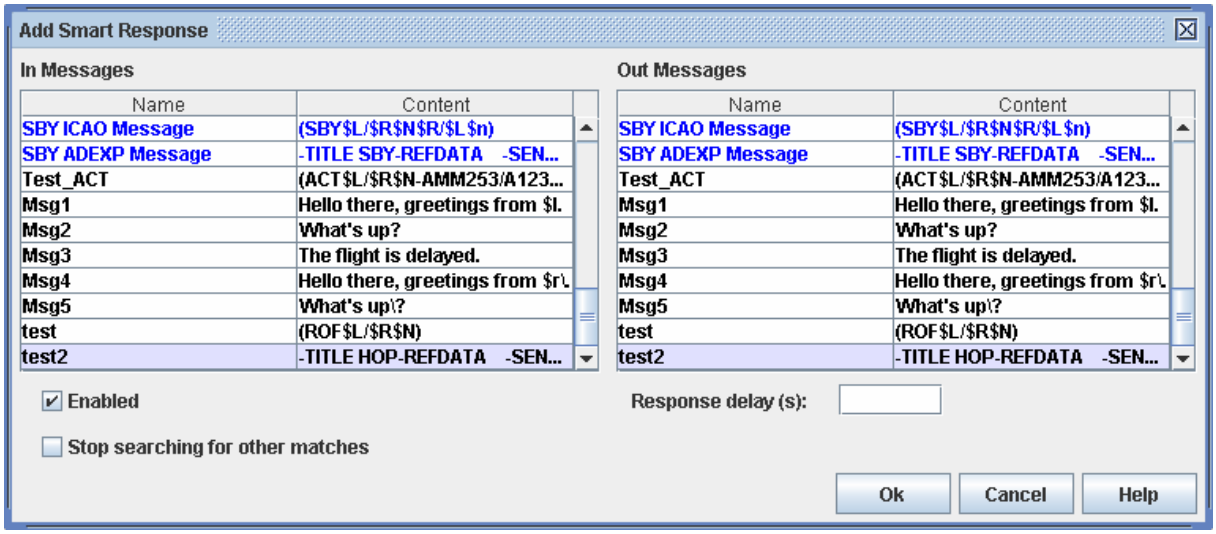


Figure 27: Add Smart Response window

8.11.2 Modifying a smart response

A smart response is modified by using the toolbar button  , by double clicking on the smart response in the Smart Response window or by selecting the smart response and pressing the keyboard space bar. This opens a window similar to Figure 27 as it has a different title.

When working with a locked or conformance session, smart responses cannot be edited.

8.11.3 Deleting a smart response

A smart response is deleted from the session by using the toolbar button  or by selecting the smart response and pressing the delete key. A confirmation window will appear to prevent accidental deletion.

When working with a locked or conformance session, smart responses cannot be removed.

8.12 Sequences

Within a session, sequences are defined and managed with the Sequence window shown in Figure 28 below. This window can be opened by using the menu option *View/Sequences* or the toolbar button  .

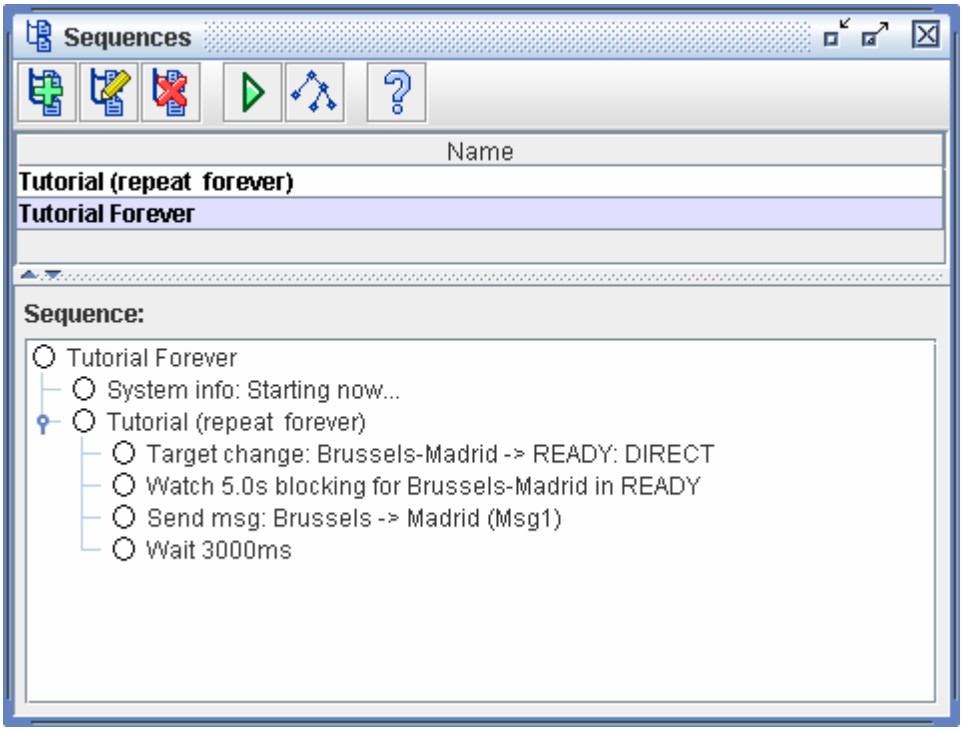


Figure 28: Sequence window

8.12.1 Adding a sequence

A sequence is created by using the toolbar button  . This opens the window shown in Figure 29 below. The default number of iterations for a sequence is 1. To change the number of iterations, click on the 'Sequence' radio button under 'Modify Item' and enter the required number into the '#Iterations' field. Section 8.12.2, Sequence items, provides a detailed description of the sequence items.

To add a sequence item create its details in the Add Item field. Click on the arrow to insert it in the sequence item list. New items are always added below the currently selected item.

Sequences can not be added to a locked or conformance session.

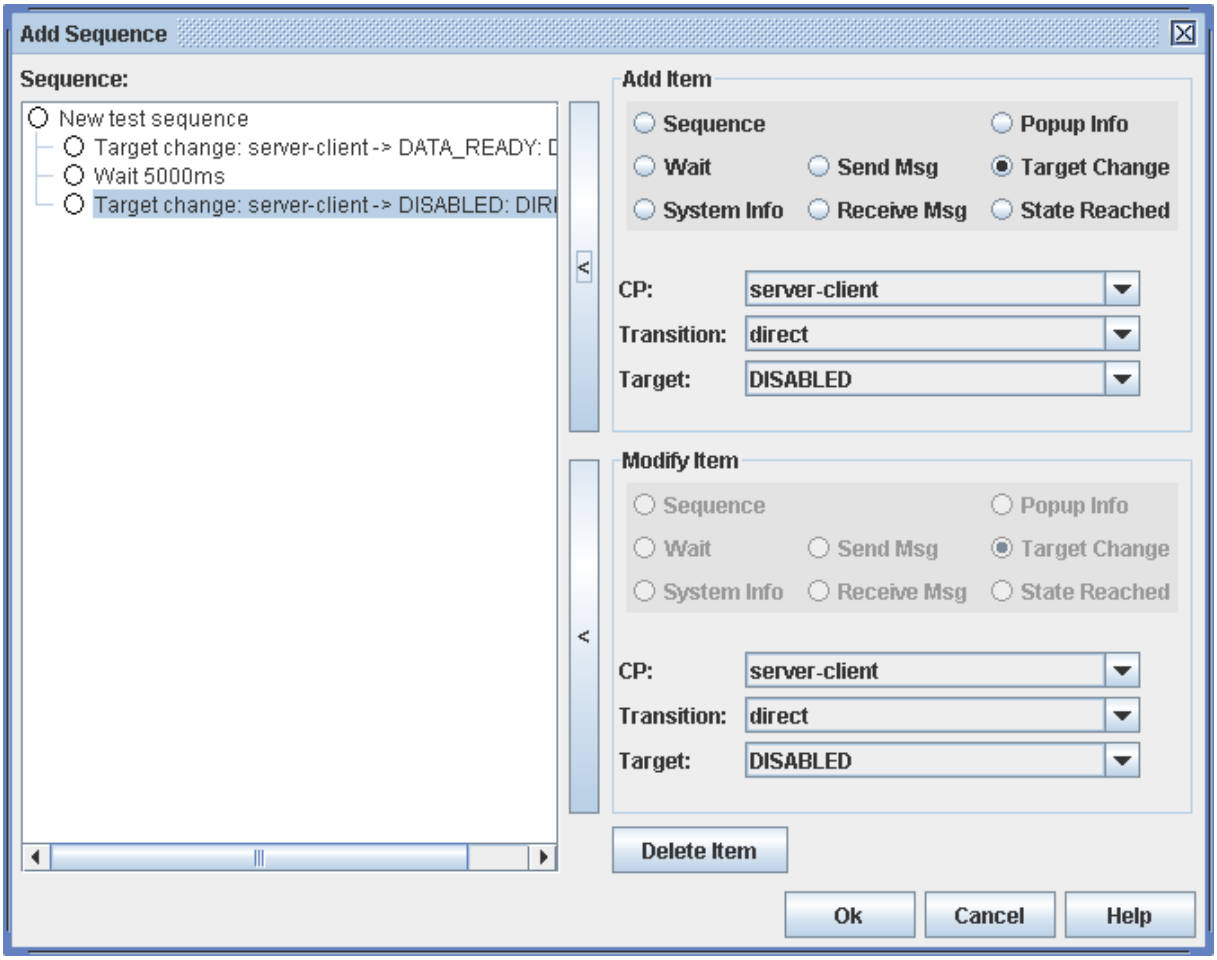


Figure 29: Add Sequence window

8.12.2 Sequence items

A sequence is an ordered list of items that can be a:

- **Sequence**: This creates a sub-sequence by nesting to a previously defined sequence.
- **Wait action**: insert a time wait period within the sequence.
- **System Info action**: a system information string will be displayed in the monitor window and logged. Note that 'System Info' is intended to be used for logging and recording progress whilst a Sequence is being run.
- **Send Message action**: send a given message from a CP.
- **Popup Info action**: Display a popup message window to the user. This action can be set as blocking (the sequence waits until the user closes the popup) or non-blocking (the sequence will continue executing the next sequence item and the popup window will remain until the user closes it). Note that 'Pop Up' info is intended to be used for giving actions or instructions to the person running a Sequence.
- **Target Change action**: change the target state of a CP (the sequence will continue executing the next sequence item event though the CP has not yet reached the target state).
- **Receive Message event watch**: check that an incoming message matches a pre-defined message pattern.
- **State Reached event watch**: check that a given CP reaches (or is in) a given state.

For event watch items:

- A maximum time period in seconds (delay) has to be specified in which the event is to occur. The timer range is [0 - 2147483.647] seconds, empty means infinity.
- Event watches can be blocking or non-blocking. In the first case, the sequence waits until the event occurs or the time period expires. In the second case, the sequence will continue executing the next sequence item while the event watch continues working in the background.
- Event watches can be negated meaning that the event watch becomes successful if the event does not occur within the time period. The Receive Message event watch will be successful within the time period if all incoming messages do not match the message pattern. The State Reached event watch is successful within the time period if the CP state never reaches the defined state. In both cases, the event watches become successful when the timer period expires.

Every sequence is assigned with an iteration parameter that defines the number of times it is to be repeated or looped. For ease of use, the number of iterations is appended to the sequence name when different to 1. To change the number of iterations, click on the 'Sequence' radio button under 'Modify Item' and enter the required number into the '#Iterations' field. Valid iteration values are:

- 0 which can be used to prevent a sequence from running
- a value in the range [1- 2147483647]
- '*' means that the sequence should loop forever.

8.12.3 Modifying a sequence

A sequence can be modified by using the toolbar button  , by double clicking on the sequence in the Sequence window or by pressing the keyboard space bar when it is selected in the Sequence window. This opens the window shown in Figure 30 below. Section 8.12.2, Sequence items, provides a detailed description of the sequence items.

To edit a sequence item, select it within the sequence list and its details will appear in the 'Modify Item' field. Click on the '<' arrow button to apply any changes. Within the sequence list items cannot be moved or duplicated.

Note that to change the number of iterations, click on the 'Sequence' radio button under 'Modify Item' and enter the required number into the '#Iterations' field.

Sequences cannot be edited when working with a locked or conformance session.

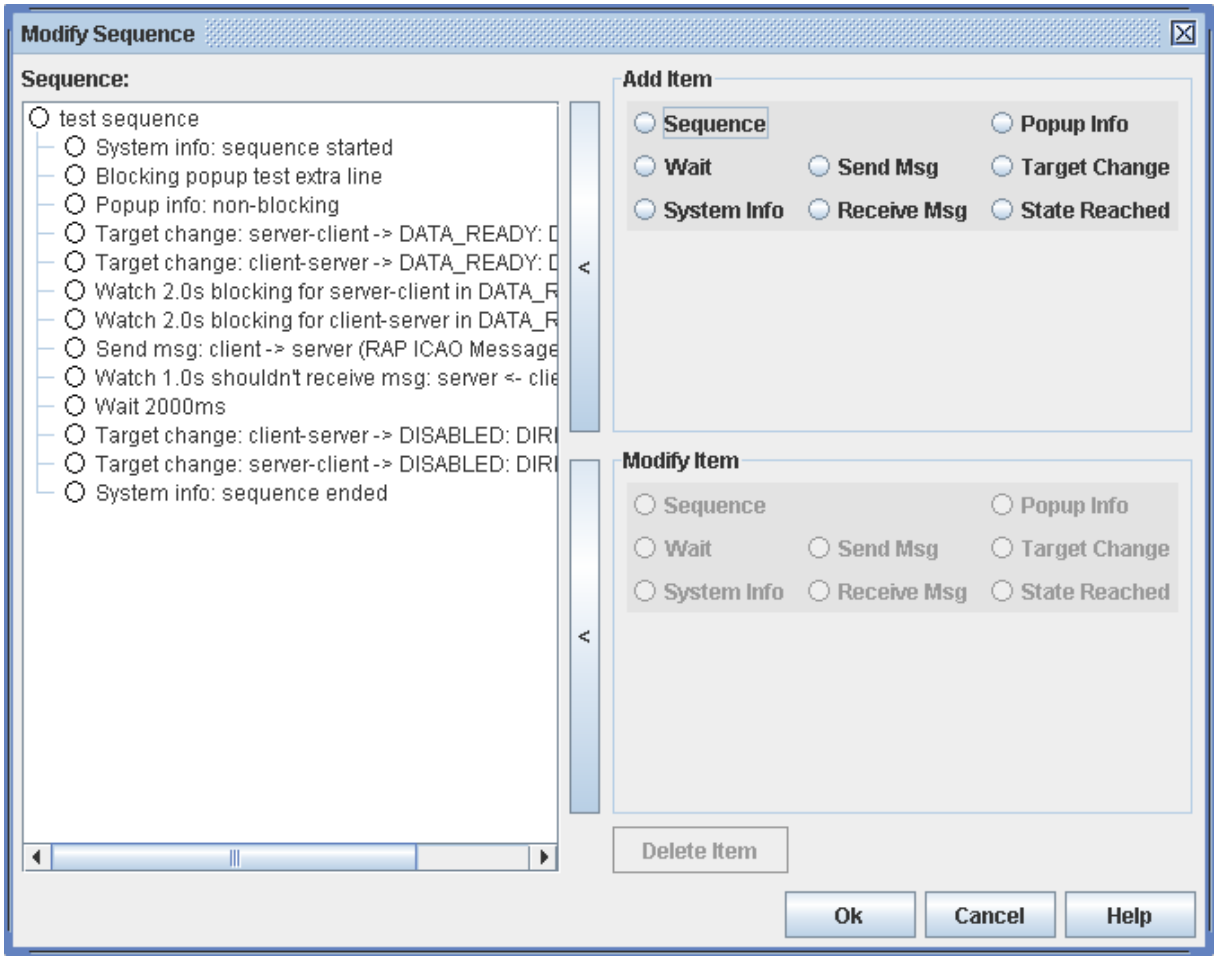


Figure 30: Modify Sequence window

8.12.4 Deleting a sequence

A sequence can be deleted from the session by using the toolbar button  or by selecting the sequence and pressing the keyboard delete key. A confirmation window will appear to prevent accidental deletion.

Sequences can not be removed from a locked or conformance session. Furthermore, a sequence may not be deleted if other ETIC entities such as a sequence depend on it.

8.12.5 Running a sequence

A Run Sequence window as shown in Figure 31 below will appear once the selected sequence is opened by using the toolbar button  . The ‘Stop on failure’ checkbox defines whether the sequence should continue or not executing sequence items even if one item fails. Note that if ‘Stop on failure’ is ticked for a Sequence with ‘Number of Iterations’ greater than 1, the sequence will stop as soon as any failure is encountered. It will not keep trying until the specified number of iterations has been attempted.

Sequences can be run concurrently but the user is to ensure that there are no inter-dependencies.

When running a sequence, the current state of a sequence item is displayed using the following colour code convention:

- White: The sequence action or event watch has not yet been executed.
- Orange: The sequence action or event watch is being executed.
- Green: The sequence action or event watch is successful.
- Red: The sequence action or event watch is unsuccessful.

A sequence (or sub-sequence) will be reported as failed if any action or event watch item is unsuccessful. For example, Figure 31 below, the second sub-sequence item fails to send a message. When the last sub-sequence item is executed the sub-sequence will be reported as failed. The overall test Sequence will inherit the failure and also be reported as failed.

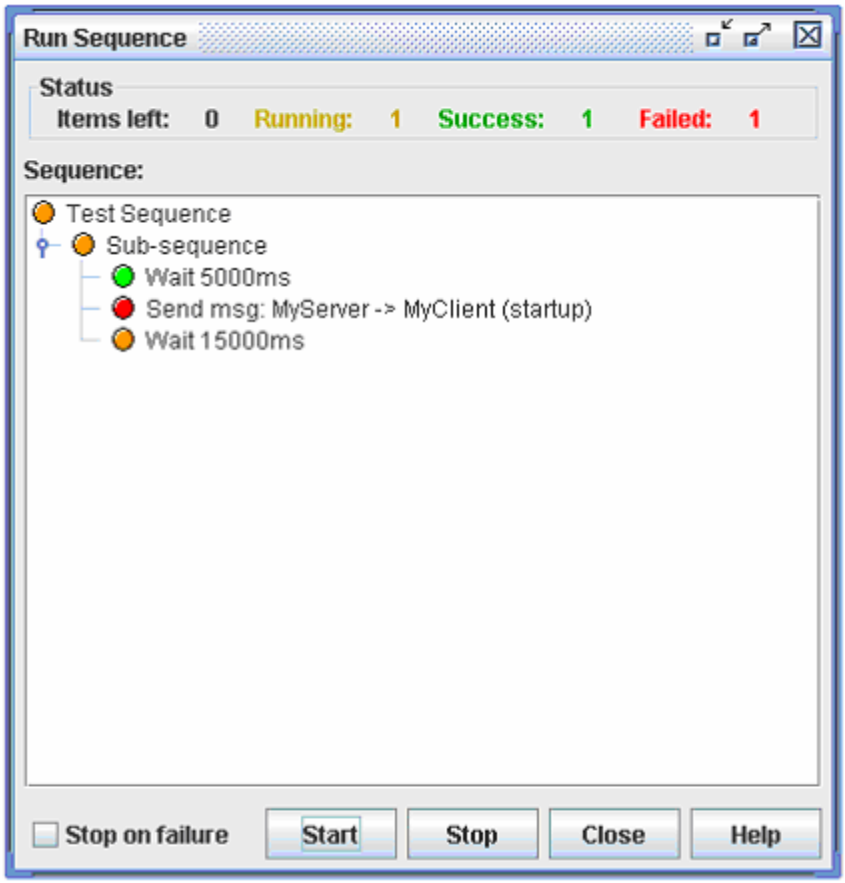


Figure 31: Run Sequence window

Figure 31 also shows that sub-sequence items can be displayed simply by clicking on the switch that links the sub-sequence to its parent sequence.

8.12.6 Check dependencies of a sequence

To determine which sequences are using the selected sequence, the toolbar button  is to be used. As shown in Figure 32 below, this displays a window containing the names of all the sequences that are using it. A sequence cannot be deleted if used by another sequence.

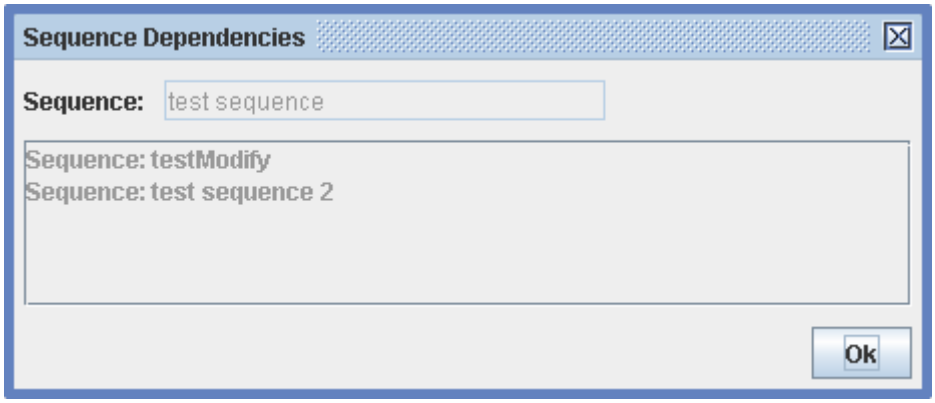


Figure 32: Sequence Dependencies window

8.13 Relays

Multiple relays can be configured between multiple CPs to allow ETIC to act as a relay between external FDE ICD 1 and/or FMTP implementations. The flow of each relay can be configured as bi-directional or uni-directional and the type of messages to be relayed can be customised by the user. Relay configuration and state is shown in the Relays windows Figure 33, which can be opened by using the menu option *View/Relays* or the toolbar button  .

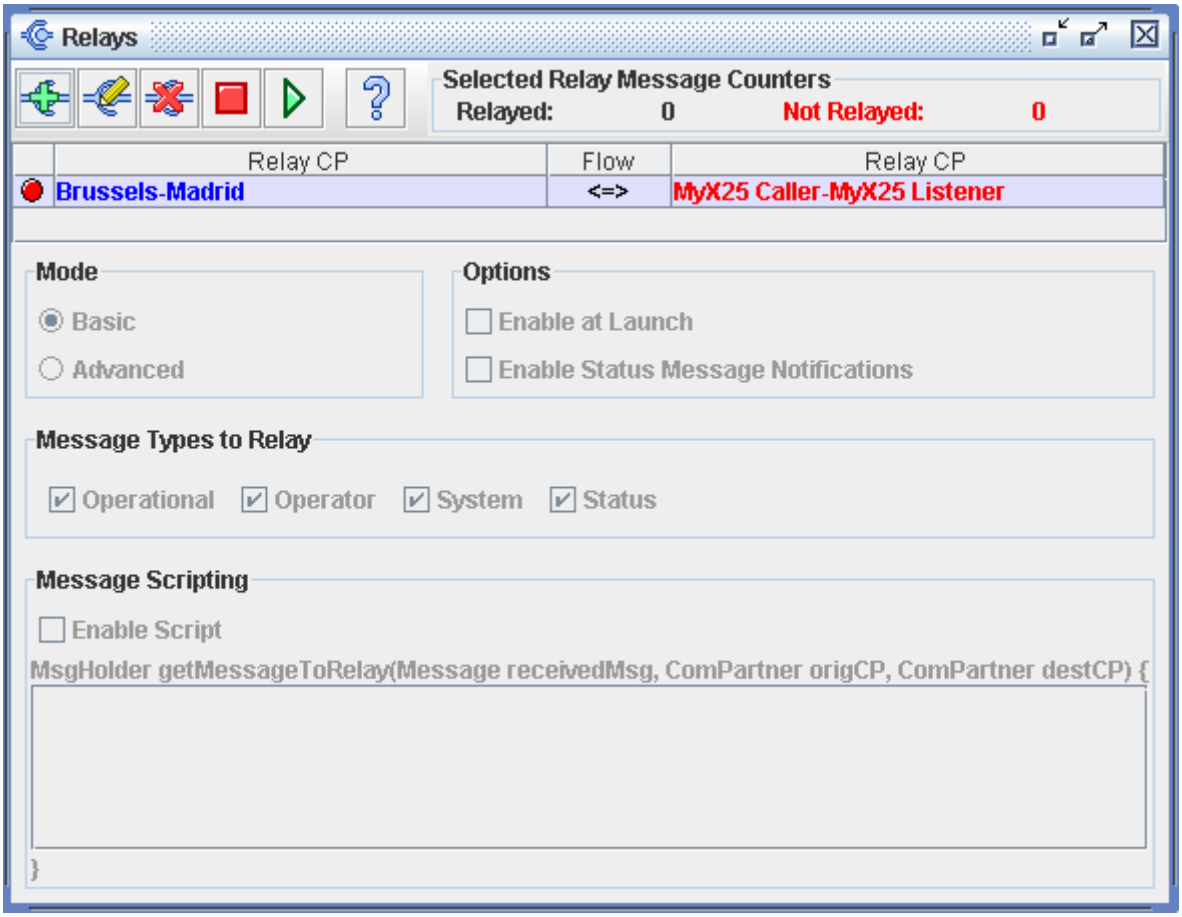


Figure 33: Relays window

The ETIC system setting ‘data section max size’ defines the maximum size of user data. This setting must equal or exceed the capabilities of the external systems.

8.13.1 Basic relays

ETIC CPs that belong to a basic relay are not end-points of an FDE or FMTP association; the end-points of the association are the external systems which exchange STARTUPs and HEARTBEATs according to their respective state machine via ETIC. As a consequence, CP targets are restricted to DISABLED or READY.

For a basic relay to operate, both CPs of the relay must at least reach the READY state with the external systems. To ease relay start-up, it is recommended that the target at launch of such CPs is set to READY and that the relay is enabled at launch.

8.13.2 Advanced relays

Advanced relays differ from basic relays in that the ETIC CPs can be end-points of the FDE or FMTP association. For example, one segment of the advanced relay can be an FDE ICD association with an external FDE system reaching the DATA_READY state and the second segment can be FMTP connection with an external FMTP system reaching the READY state.

Users are advised to carefully configure multiple relay combinations as it can lead to very complicated interfering message flows between external systems that are difficult to diagnose. To ease relay startup, it is recommended that advanced relays and CP targets are enabled at launch.

8.13.3 Add a relay

A relay is created by using the toolbar button  . This opens the window shown in Figure 34 below in which the user can select the two CPs forming the relay. CPs that are already part of another relay which might create message interference are displayed in grey.

Relays can be characterised by setting the following items:

- **Flow:** <=> for bi-directional, => and <= for uni-directional
- **Mode:** basic or advanced relay
- **Options:** enable relay at ETIC launch, notify non-relay issues to external systems with status messages
- **Message Types to Relay:** user defined filter of messages types to be relayed
- **Message script:** user defined groovy scripts to process incoming messages

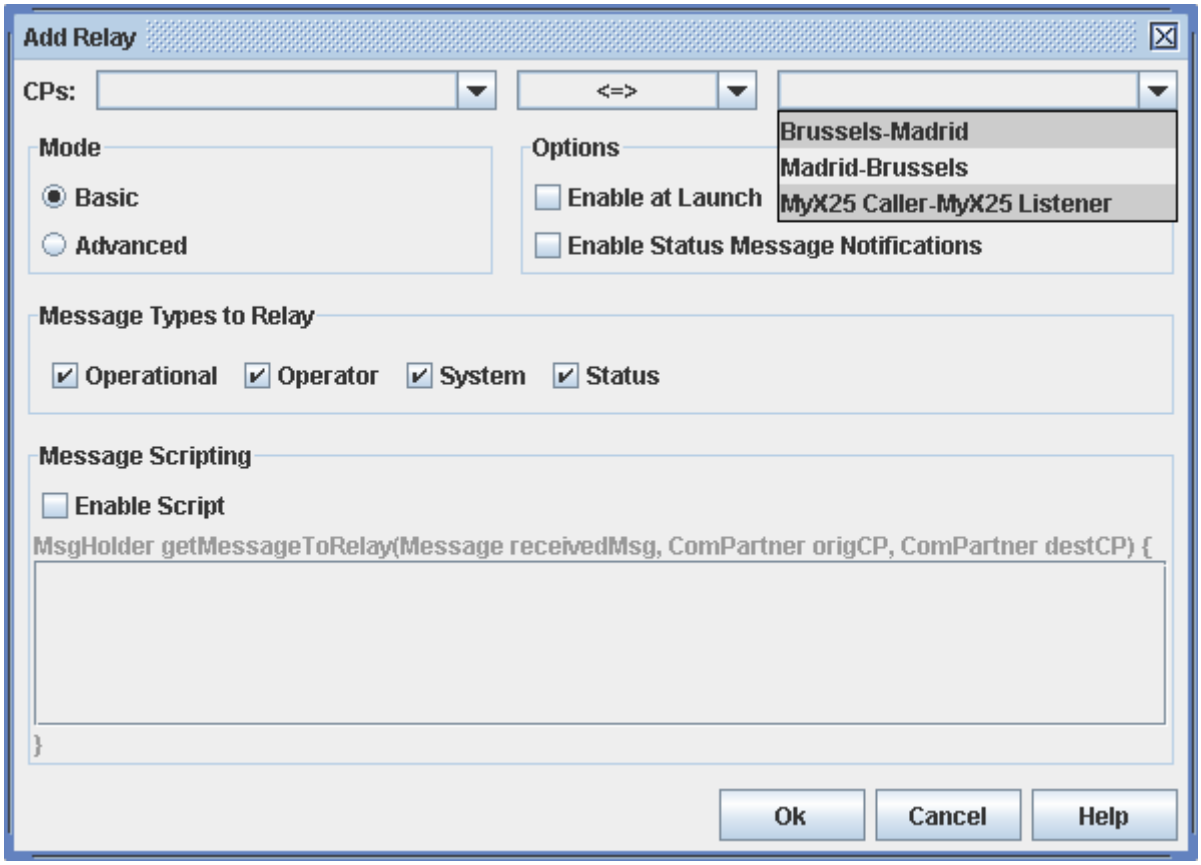


Figure 34: Add Relay window

8.13.4 Relay message scripting

Each relay can be configured with a message script which is code to be applied to the incoming message. Message scripting provides a versatile means to detect the presence or absence of specific string patterns in the incoming message, can be conditioned to the type of message and source/destination communication partners and can be followed by a decision to relay or drop the incoming message.

The code is based on groovy, examples and further references are documented in section 11.4. It is to be noted that ETIC does not validate the syntax of the script.

8.13.5 Modifying a relay

Once a relay is disabled using the toolbar button  , it can be modified by using the  button, which opens a new window similar to the Add Relay window.

8.13.6 Deleting a relay

Once a relay is deleted using the toolbar button  , it can be deleted by using the  button.

8.13.7 Enabling a relay

The relay shown in Figure 33 can be enabled by using the toolbar button  . Relays can be enabled concurrently but the user is to ensure that there are no inter-dependencies.

The current state of a relay is displayed using the following colour code convention:

- Red: The relay is disabled.
- Orange: One segment of the relay is not ready or has failed, no messages can be relayed.
- Green: Messages can be relayed.

In case ETIC cannot relay an incoming message, it is displayed in red within the Monitor window (see section 8.14, Monitoring events) followed by a system information message. For bi-directional relays, ETIC can also send a status message back to the originator as a notification.

8.14 Monitoring events

Monitoring runtime events is an essential function of ETIC. For this reason the events are cached to the memory buffer as well as written to the log system. To monitor the captured or the local events, open a Monitor Window as shown in Figure 35 below by using the menu option *View/New Monitor* or the toolbar button  .

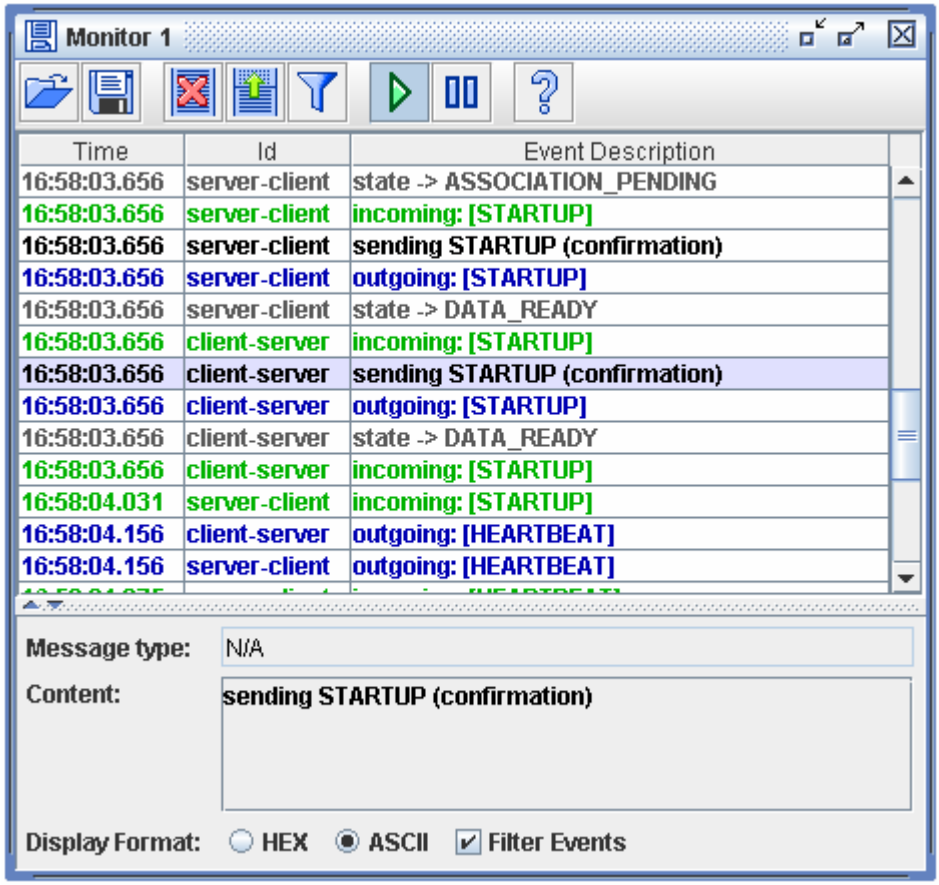


Figure 35: Monitor window

While ETIC is running, all captured events are immediately displayed in order to provide a live view. It is important to understand that memory constraints can condition the content of the live view as described in section 8.17, Memory management.

As a general rule, incoming messages are displayed in green and outgoing messages are displayed in blue. Messages that are received in the wrong state, corrupt messages or messages that could not be relayed are displayed in red.

8.14.1 Change the title of a monitor window

When multiple Monitor windows are open, it can be useful to rename them. This can be done by right clicking on the Monitor window title bar, which opens a window to change the title name.

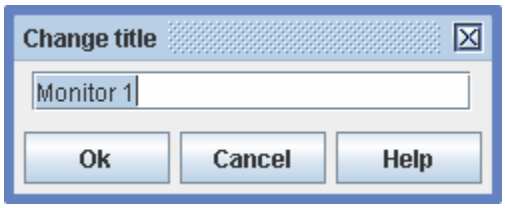


Figure 36: Change title window

8.14.2 Save a monitor window

The contents of a monitor window can be saved to an XML log file at any time by using the Monitor window toolbar button  , which causes the “Save” window to pop up as shown in Figure 37. Note that when filtering is enabled (‘Filter events’ checkbox), this will only save the events that are visible in the Monitor window.

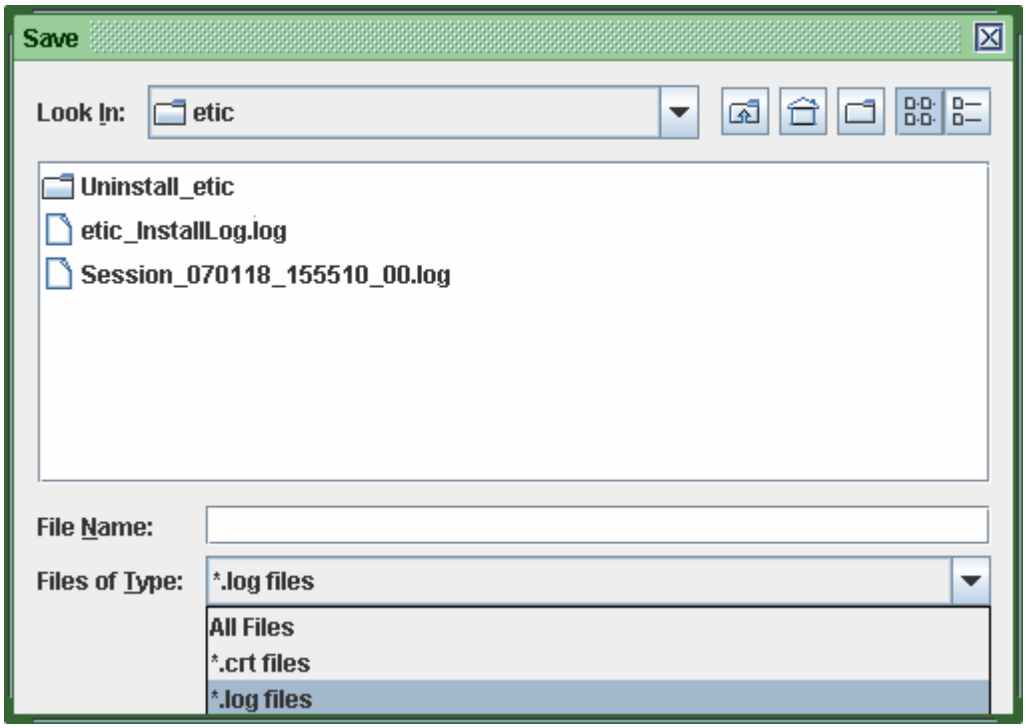


Figure 37: Save log file window

If the “Files of Type” drop down menu is set to “*.crt files” or “*.log files”, ETIC will automatically append the .crt or the .log extension to the file name chosen when saving the file, and files displayed in the “Save” window will also be filtered on the .crt or .log extension.

If the “Files of Type” drop down menu is set to “All Files”, ETIC will not append any extension nor filter the files displayed.

8.14.3 Open a previously saved monitor window

A previously saved Monitor window log file can be loaded by using the Monitor window toolbar button  .

For ease of use, session log files (see section 8.4, Event logging system) and monitor log files have the same XML format, and are handled in the same way.

8.14.4 Filtering monitor window display

By default, when a Monitor window is opened filtering is enabled (‘Filter events’ checkbox). The event display start time is set to the current time. As a result, only new events will be displayed. The user can disable the filtering by selecting the ‘Filter events’ checkbox which also disables the following filter related buttons.

Specific filter settings can be assigned to each monitor window. These settings can be edited by using the toolbar button  . This opens the Filter Settings window as shown in Figure 38 below. The filter settings can be dynamically enabled or disabled at any time by using the checkbox ‘Filter Events’ on the Monitor Window. Refer to Table 11 for a detailed description of each setting.

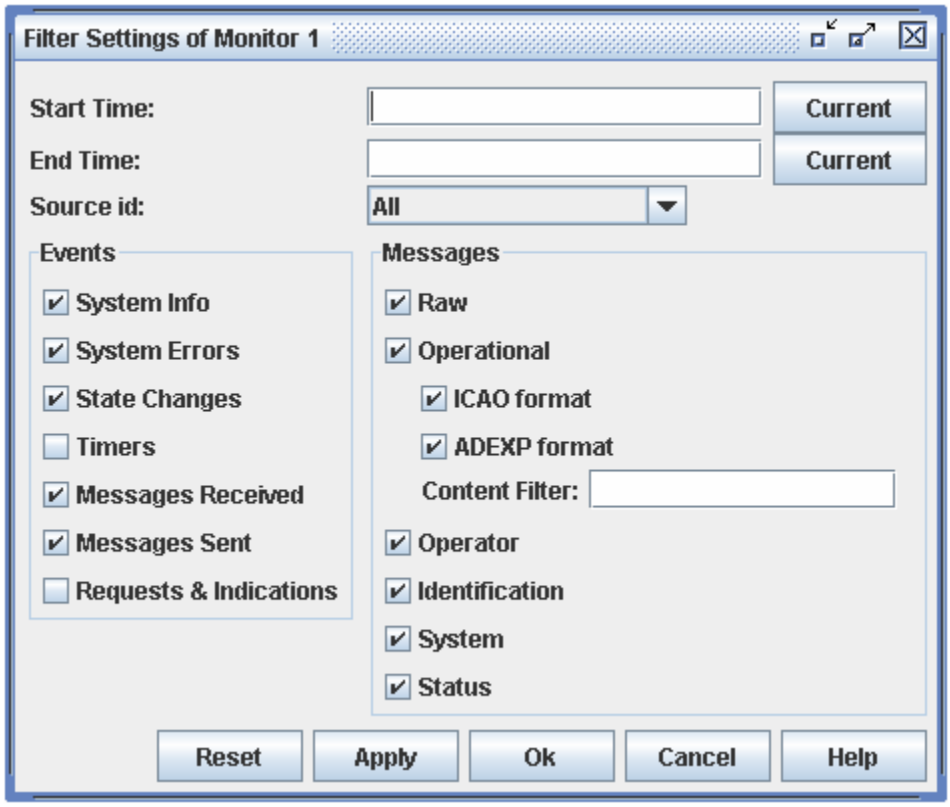


Figure 38: Filter Settings window

Setting	Description
Start Time	This is the event display start time for the monitor window. It will only display events that occurred after this time. The date/time format is “DD-MM-YYYY HH:MM:SS.mmmm” but it is sufficient to enter the HH:MM and ETIC will make use of the current date. When this setting is empty it is equivalent minus infinity to display all past events retained in memory.
End Time	This is the event display start end time for the monitor window. It will only display events that occurred before this time. The date/time format is “DD-MM-YYYY HH:MM:SS.mmmm” but it is sufficient to enter the HH:MM and ETIC will make use of the current date. When this setting is empty it is equivalent infinity which will display all future events.
Source id	This dropdown setting allows to filter on CPs that are present within the Monitor window.
System Info	This checkbox setting enables or disables the display of System Information events.
System Errors	This checkbox setting enables or disables the display of System Error events.
State Changes	This checkbox setting enables or disables the display of State Change events.
Timers	This checkbox setting enables or disables the display of Timer events.
Messages Received	This checkbox setting enables or disables the display of Message Received events.
Messages Sent	This checkbox setting enables or disables the display of Message Sent events.
Requests & Indications	This checkbox setting enables or disables the display of FMTP or FDE ICD Part 1 service primitive invocation events.
Raw	This checkbox setting enables or disables the display of sent Raw messages.
Operational	This checkbox setting enables or disables the display of sent and received operational messages.
ICAO format	This checkbox setting enables or disables the display of sent and received operational messages complying with the ICAO format.
ADEXP format	This checkbox setting enables or disables the display of sent and received operational messages complying with the ADEXP format.
Content Filter	The Content Filter field setting for operational messages enables or disables the display of operational messages that match the regular expression. When this fields is empty all matches match and will be displayed.
Operator	This checkbox setting enables or disables the display of sent and received operator messages.
Identification	This checkbox setting enables or disables the display of sent and received identification messages.
System	This checkbox setting enables or disables the display of sent and received system messages.
Status	This checkbox setting enables or disables the display of sent and received status messages.

Table 11: Filter Settings

8.14.5 Show all events

The user can display past events by using the toolbar button  . This will set event display start time to the oldest date/time of the operating system, so that all events retained in memory can be displayed.

8.14.6 Clearing the monitor window

The monitor window can be 'cleared' at any time by using the toolbar button  . This does not actually delete the events from the memory buffer, but simply changes the event display start time of the Monitor window to the current time. As a result, the past events are not displayed. This is very different to clearing the events from memory described in section 8.17.1.

8.14.7 Start or stop the auto-scroll feature

By default, Monitor windows scroll in order to display the most recent events. Scrolling can be disabled by clicking on the toolbar 'pause' button  . The scrolling feature can be enabled at any time by using the toolbar button  .

8.15 Opening the help system

A context sensitive help system is available at any time by using the menu option *Help/Help*, the Help buttons at the lower right corner of a window or the question mark button  in the window toolbar. This opens the ETIC Help dialog as shown below in Figure 32. The help system can be considered a separate application and requires time to open.

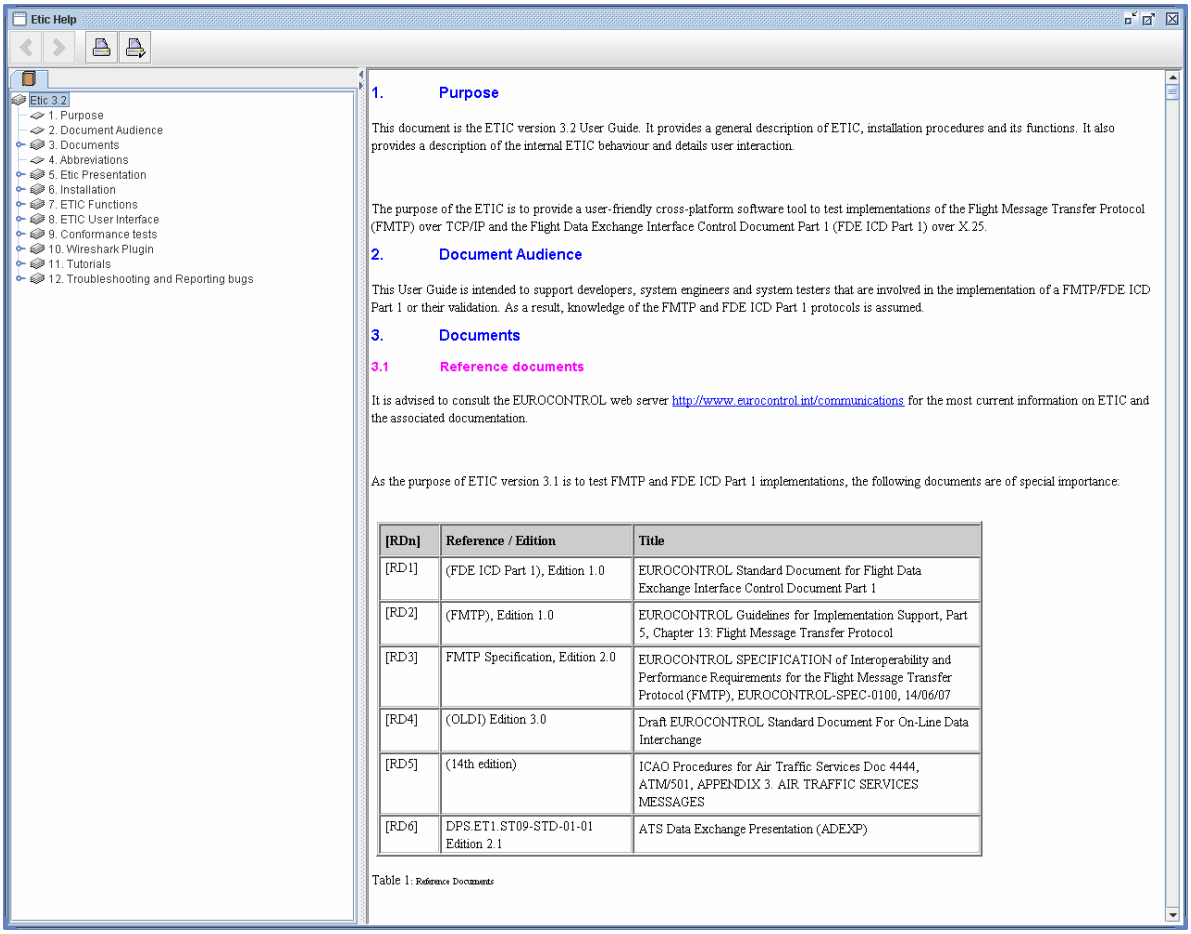


Figure 39: ETIC help

8.16 Opening the About window

The ETIC About window is opened by using the menu option *Help/About*. It indicates the exact release date and version of the ETIC application.

Figure 40: About window

8.17 Memory management

8.17.1 Internal management

During a session, all events are stored in memory until the memory buffer is cleared or overwritten with new events. A live Monitor window provides a view of all events that exist in the memory buffer. Once this memory buffer runs out of space, ETIC will automatically remove the oldest events according to the parameters defined in the System Settings (see section 8.2, Changing or reviewing the system Settings) affecting the content of Monitor window. Note that the actions taken by ETIC when this memory buffer runs out of space will override any 'Pause' (see section 8.14.7) that has been applied to the Monitor display scrolling. Note also that the user can apply filtering to the monitor to increase the time until the buffer becomes full.

In parallel, the Java run-time is constantly creating and deleting objects leading to Java garbage collection cycles to recover unused space. ETIC Users can force ETIC to release memory as described below.

8.17.2 Clearing events from memory

It is possible to delete all events stored in memory at any time by using the menu option *Session/Clear events* or the toolbar button  . Consequently, all opened Monitor windows will be emptied as there are no events to display.

Clearing events from memory does not impact the logging system whereby the events will be stored to a session and system log.

8.17.3 Clearing the system memory

To force a Java garbage collect and possibly free up some system memory, click on the  icon next to the memory consumption information bar. This information bar becomes visible once the associated checkbox is enabled in the System Settings (see section 8.2, Changing or reviewing the system Settings).

9 Conformance Tests

9.1 Introduction

One of the main benefits of this version of ETIC is pre-defined conformance session to validate an implementation. For this purpose, ETIC enters the special conformance mode in which most of the ETIC functions are inhibited. In conformance mode, the user can only

- Configure the IP addresses of the pre-defined communication partners,
- Execute the predefined conformance sequences, and
- Open Monitor windows.

In practice, to validate conformance with the FMTP protocol, the following steps need to be followed:

1. The SuT has to be configured for the conformance test.
2. ETIC needs to be in conformance mode (see section 8.8, Conformance sessions).
3. The conformance session is configured by setting the IP addresses of the CP and the SuT (this configuration can be saved by using the save button).
4. The client or server conformance test sequence is run.
5. Any pop-up should be carefully read and the described actions strictly followed.

The test is either successful or unsuccessful. When successful, a certificate can be produced and stored locally. This certificate is a binary file that can be emailed to any counterpart such as etic@eurocontrol.int. Certificates can be viewed with Monitor window (*System/View Log*).

If the test is unsuccessful or fails, the SuT did not respond as expected by the test sequence. The user should then determine which sequence item has failed. Looking to the previous sequence items as well as the events in an event window, the cause of the failure should be easily deduced.

If a conformance session fails, then either:

1. Corrective action on the SuT needs to be taken,
2. The sequence is not adapted to the system, or
3. The sequence is wrong.

It is possible that the sequences do not properly inter-work with the SuT. For instance, an implementation could be such that it tries to reach the DATA_READY state as soon as it enters the READY state. In other words it automates the MT-Associate service upon an MT-Connect indication. If the SuT cannot perform a MT_CONNECT without performing an MT-Associate it is highly likely that the conformance test will fail. In such a case it is advised to contact the ETIC support team at etic@eurocontrol.int.

Conformance session sequences have been carefully built and tested. Nevertheless, if a problem is suspected, a problem description should be sent to the ETIC support team at etic@eurocontrol.int.

9.2 The System_under_Test_is_TCP_CLIENT conformance test

This conformance test verifies whether the SuT complies with the FMTP TCP/IP client specification. It is assumed that the SuT can remain in the READY state without trying to automatically reach DATA_READY. The configuration of the SuT should be as follows (Local and Remote ids are editable):

- Local id: SUT_CLIENT
- Remote id (i.e. ETIC id): ETIC_SERVER
- Ti timer set to 10s
- Tr timer set to 8s
- Ts timer set to 6s

Furthermore, **it is assumed that the SuT is capable** of sending an operational message (even with meaningless content), is capable of receiving an operational message with dummy content without disconnecting and is capable of transmitting at least 3 STARTUP messages following a Tr timeout in ASSOCIATION_PENDING.

Before starting the sequence, the IP addresses of the ETIC CP named ETIC_CLIENT have to be configured. For practical reasons the conformance session allows for both IPv4 and IPv6 addresses.

Once the sequence starts, the ETIC user will be invited through popup windows to invoke the following FMTP services on the SuT:

- MT_CONNECT,
- MT_ASSOCIATE,
- MT_STOP,
- MT_DISCONNECT,
- Sending an operational message

Subsequence items are named according the row of the Tables A.4 and A.6 of Reference [RD2]. For instance, _A.4.9 checks that the actions defined at the ninth row of table A.4 (i.e. State=ID_PENDING, event=Ti_timeout) are correctly performed by the SuT.

Seq_Ref	Annex A table number	Annex A state	Annex A event
_A.4.1,5,8	4	IDLE	L_setup
		CONNECTION_PENDING	R_setup
		ID_PENDING	R_disconnect
_A.4.9	4	ID_PENDING	Ti_timeout
_A.4.10 (send accept)	4	ID_PENDING	R_Accept
_A.4.11a (malformed id)	4	ID_PENDING	R_id (first case)
_A.4.11a (wrong local id)	4	ID_PENDING	R_id (first case)
_A.4.11b	4	ID_PENDING	R_id (second case)
_A.6.2 client	6	READY	R_Disconnect
_A.6.3,8 client	6	READY	L_startup
		ASSOCIATION_PENDING	Tr_timeout
_A.6.6 client	6	ASSOCIATION_PENDING	R_disconnect
_A.6.9 client	6	ASSOCIATION_PENDING	R_startup

_A.6.16 client	6	DATA_READY	R_data
_A.6.12 client	6	DATA_READY	R_disconnect
_A.6.19 client	6	DATA_READY	R_shutdown
_A.6.15 client	6	DATA_READY	Ts_timeout
_A.6.14 client	6	DATA_READY	L_data
_A.6.1 client	6	READY	L_disconnect
_A.6.7 client	6	ASSOCIATION_PENDING	L_shutdown
_A.6.5 client	6	ASSOCIATION_PENDING	L_disconnect
_A.6.13 client	6	DATA_READY	L_shutdown
_A.6.11 client	6	DATA_READY	L_disconnect
_A.6.17 client	6	DATA_READY	R_heartbeat
_A.6.18 client	6	DATA_READY	Tr_timeout

Table 12: Conformance items for SuT being TCP client vs. Reference [RD2]

9.3 The System_under_Test_is_TCP_SERVER conformance test

This conformance test verifies whether the SuT complies with the FMTP TCP/IP server specification. It is assumed that the SuT can remain in the READY state without trying to automatically reach DATA_READY. The configuration of the SuT should be as follows (Local and Remote ids are editable):

- Local id: SUT_SERVER
- Remote id (i.e. ETIC id): ETIC_CLIENT
- Ti timer set to 10s
- Tr timer set to 8s
- Ts timer set to 6s

Furthermore, **it is assumed that the SuT is capable** of sending an operational message (even with meaningless content), is capable of receiving an operational message with dummy content without disconnecting and is capable of transmitting at least 3 STARTUP messages following a Tr timeout in ASSOCIATION_PENDING.

Before starting the sequence, the IP addresses of the ETIC CP named ETIC_SERVER have to be configured. For practical reasons the conformance session allows for both IPv4 and IPv6 addresses.

Once the sequence starts, the ETIC user (i.e. the tester) is requested to act on the SuT. In particular, the following FMTP requests are required:

- MT_CONNECT,
- MT_ASSOCIATE,
- MT_STOP,
- MT_DISCONNECT,
- Sending an operational message

As for the client case, subsequence items are named according the row of the Tables A.5 and A.6 of Reference [RD2].

Seq_Ref	Annex A table number	Annex A state	Annex A event
_A.5.1,5	5	IDLE	R_setup
		SYSTEM_ID_PENDING	Ti_timeout
_A.5.4	5	SYSTEM_ID_PENDING	R_Disconnect
_A.5.6	5	SYSTEM_ID_PENDING	R_accept
_A.5.7a	5	SYSTEM_ID_PENDING	R_id (malformed id)
_A.5.7b,10	5	SYSTEM_ID_PENDING	R_id
		ID_PENDING	R_disconnect
_A.5.11	5	ID_PENDING	Ti_timeout
_A.5.12 (startup)	5	ID_PENDING	R_startup
_A.5.13	5	ID_PENDING	R_accept
_A.6.2 server	6	READY	R_Disconnect
_A.6.3,8 server	6	READY	L_startup
		ASSOCIATION_PENDING	Tr_timeout
_A.6.6 server	6	ASSOCIATION_PENDING	R_disconnect
_A.6.9 server	6	ASSOCIATION_PENDING	R_startup
_A.6.16 server	6	DATA_READY	R_data
_A.6.12 server	6	DATA_READY	R_disconnect
_A.6.19 client	6	DATA_READY	R_shutdown
_A.6.15 server	6	DATA_READY	Ts_timeout
_A.6.14 server	6	DATA_READY	L_data
_A.6.1 server	6	READY	L_disconnect
_A.6.7 server	6	ASSOCIATION_PENDING	L_shutdown
_A.6.5 server	6	ASSOCIATION_PENDING	L_disconnect
_A.6.13 server	6	DATA_READY	L_shutdown
_A.6.11 server	6	DATA_READY	L_disconnect
_A.6.17 server	6	DATA_READY	R_heartbeat
_A.6.18 server	6	DATA_READY	Tr_timeout

Table 13: Conformance items for SuT being TCP server vs. Reference [RD2]

10 Wireshark Plugin

10.1 Overview

Wireshark® (http://www.wireshark.org) is used by network professionals around the world for troubleshooting, packet analysis, software and protocol development, and education. The development supersedes the popular Ethereal® which is no longer maintained. Wireshark is freely available as open source, and is released under the GNU General Public License. It runs on all popular computing platforms, including Unix, Linux, Mac OS X and Windows. Its open source license allows the addition of enhancements such as plugins for user created protocols.

The FMTP plugin is made available to ETIC users as a complementary tool for packet analysis as shown in Figure 41.

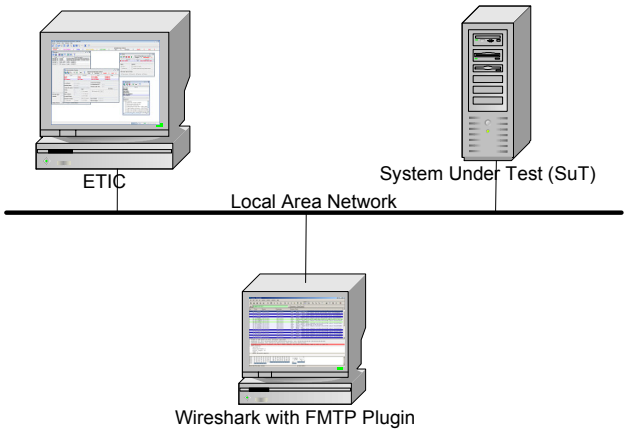


Figure 41: FMTP packet analysis

Pre-compiled versions of the plugin per operating system are included in the ETIC 3.2.2 CD-ROM or available for download from the ETIC/FMTP OneSkyTeam. By adding the FMTP plugin to their Wireshark or Ethereal software, users can decode and colour IP packets that contain FMTP messages as shown in Figure 42.

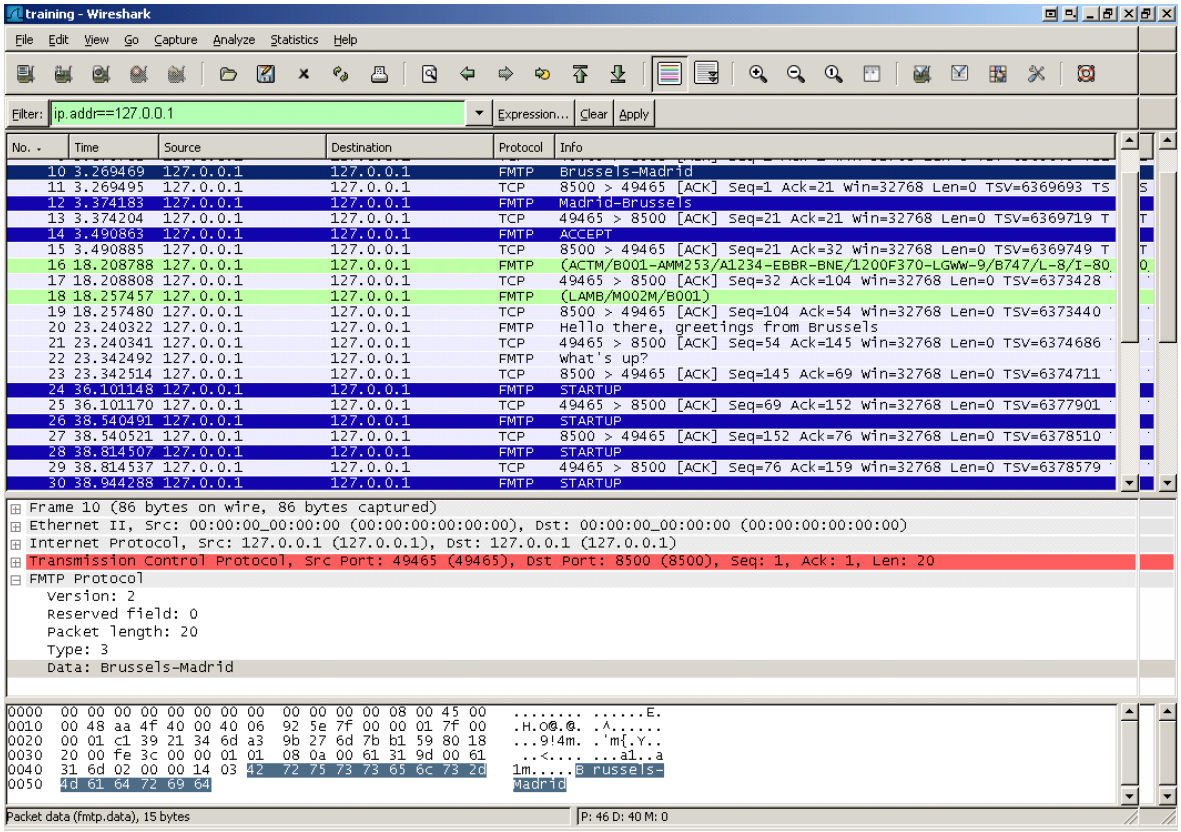


Figure 42: Wireshark FMTP plugin

The colour coding scheme allows to distinguish between connection establishment related messages, data messages and disconnection messages (SHUTDOWN/REJECT).

10.2 Installation

10.2.1 Plugin

Plugin installation simply involves copying the appropriate pre-compiled binary to the Wireshark (or Ethereal) program directory. Typically the plugin folder is located as follows:

- **Windows** c:\program files\wireshark\plugins\1.0.0 (if version 1.0.0 of Wireshark is being used)
- **Linux and Mac OS X** /usr/lib/wireshark/plugins// or /usr/local/lib/wireshark/plugins//
- **Solaris 9** /opt/sfw/lib/wireshark/plugins//

For new installations, pre-complied versions of the Wireshark software including the FMTP plugin are available for download from the FMTP/ETIC OneSky Team. Refer to Wireshark documentation for unlisted operating systems.

10.2.2 Colouring Rules

While the FMTP plugin allows Wireshark decode the FMTP packets, rules can also be defined to provide better visibility of FMTP packets by assigning different colours per FMTP packet content (see Figure 42). Ready-made rules are provided on the ETIC 3.2.0 CD-ROM or available for download from the ETIC/FMTP OneSkyTeam with the text file named `fmtpcolours`.

To install the rules, launch Wireshark (or Ethereal). Select VIEW and COLORING RULES. Import the `fmtpcolour` file which contains three colour rules named FMTP Issue, FMTP Id and FMTP Data which are defined in plain text:

```
@FMTP Issue@(fmtp.typ == 4 && fmtp.data == "00") || (fmtp.typ
== 3 && fmtp.data ==
"REJECT")@[64540,949,949][65038,64082,64082]
@FMTP Id@ fmtp.typ == 3 || (fmtp.typ == 4 && fmtp.data ==
"01")@[4694,1275,43193][65038,65038,65038]
@FMTP     Data@fmtp.typ     ==     1@[47697,65533,41011][0,0,0]
```

Wireshark is sensitive to the listed order of the rules from top to bottom. Users are advised to select all the FMTP colour rule rows then use the ‘UP’ button (located on right side of window) to move the FMTP colour rules above the standard TCP colour rule. ETIC users can create their own colour rules via the Wireshark application.

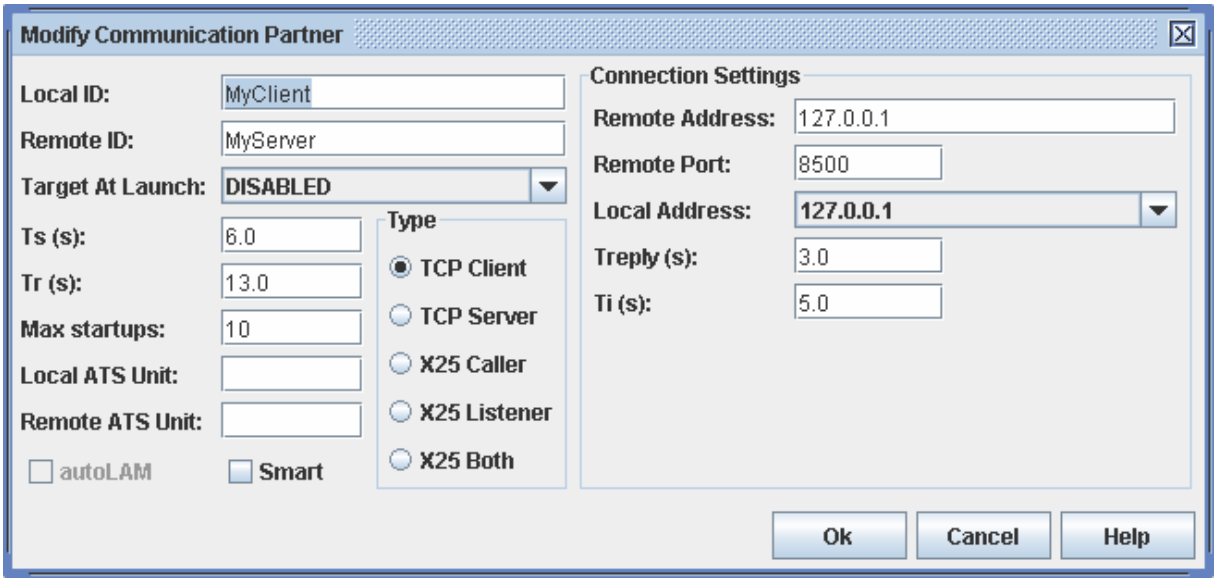
11 Tutorials

11.1 FMTP connection

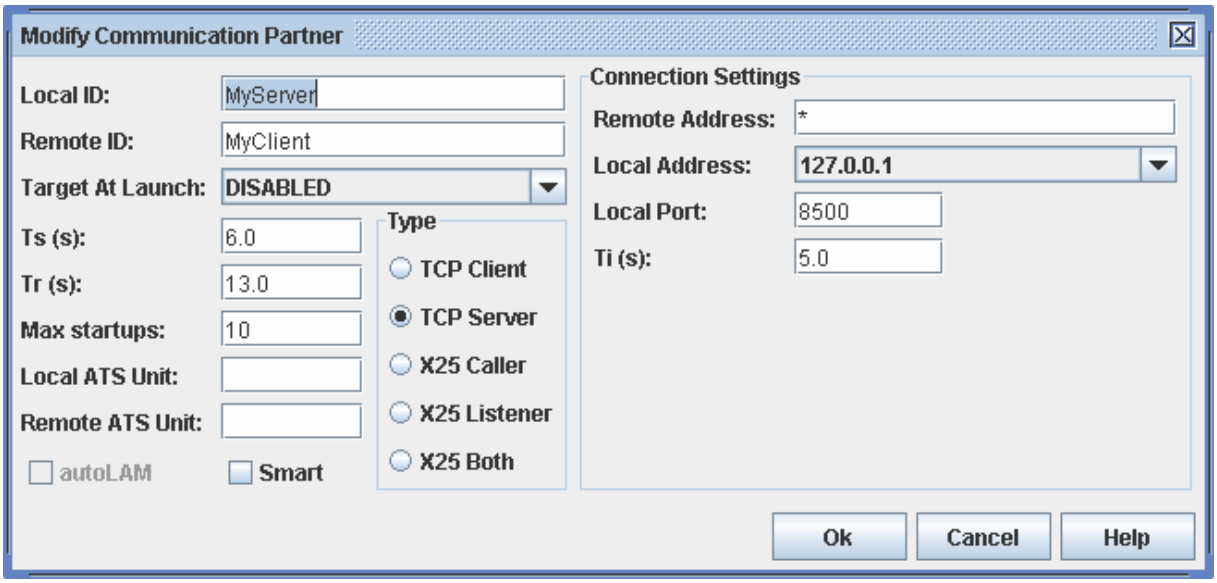
This is a step by step approach to launch a communication session with ETIC using the localhost interface or local IP address.

1. Start a new ETIC session and select the CP edition window 

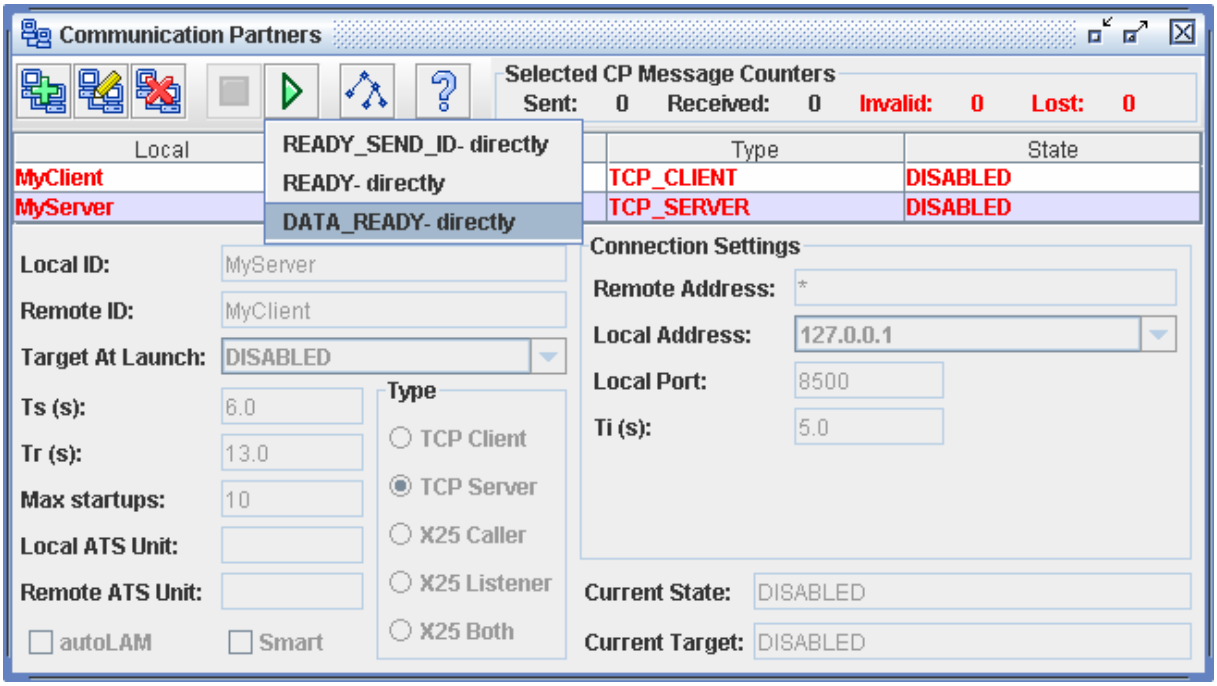
2. Select the  icon to create an entry as follows and click OK



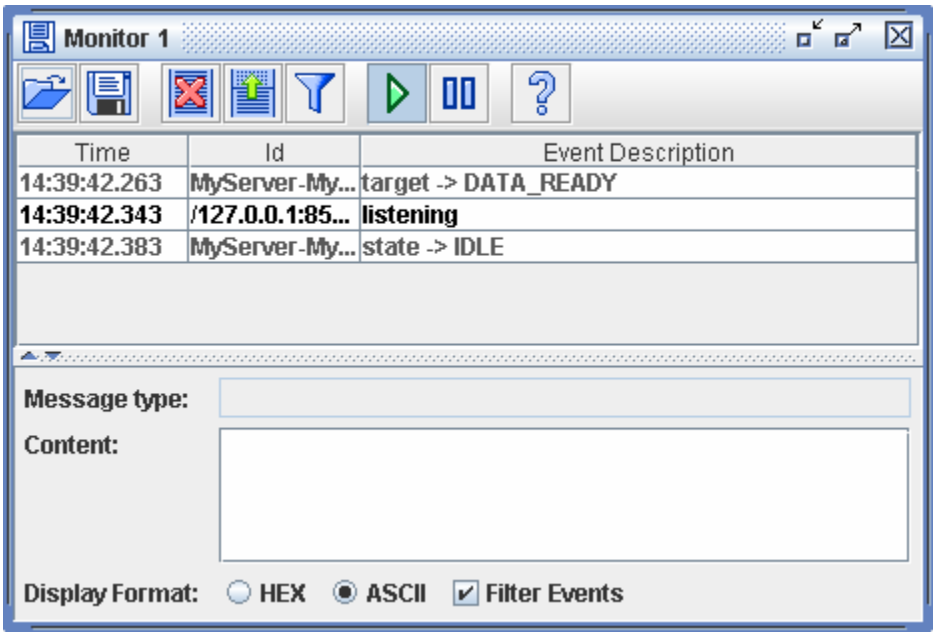
3. Select the  icon to create an entry as follows and click OK



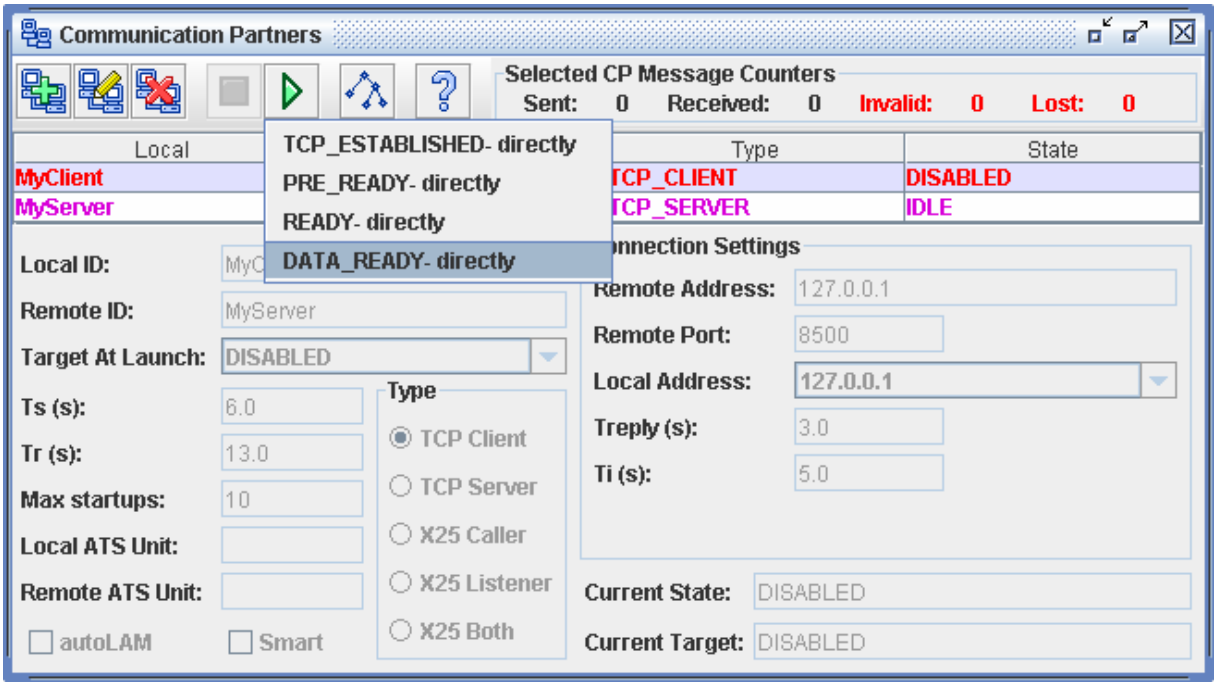
4. In the CP window select the entry that starts with MyServer and set the target state to be DATA_READY to start the server process



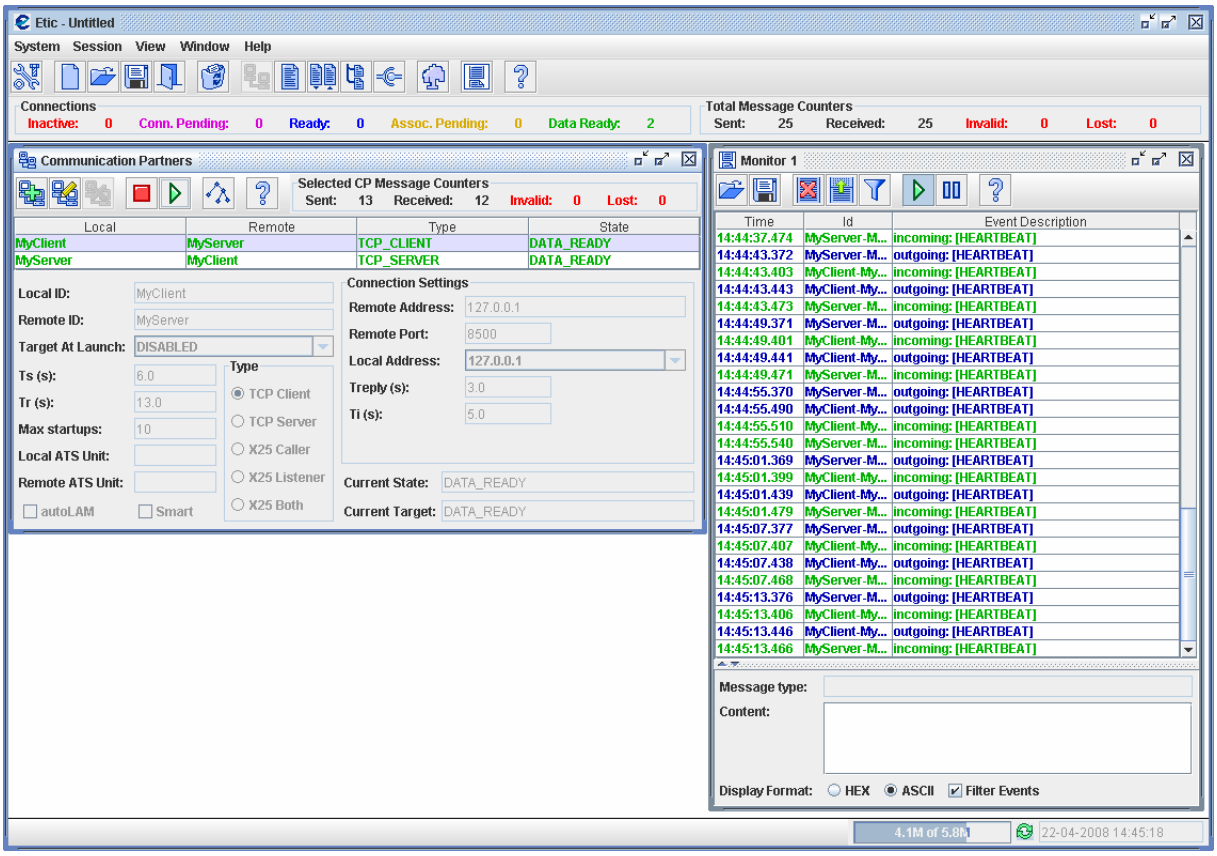
5. Open a Monitor window, you will need to use the  button to see the previous server transitions



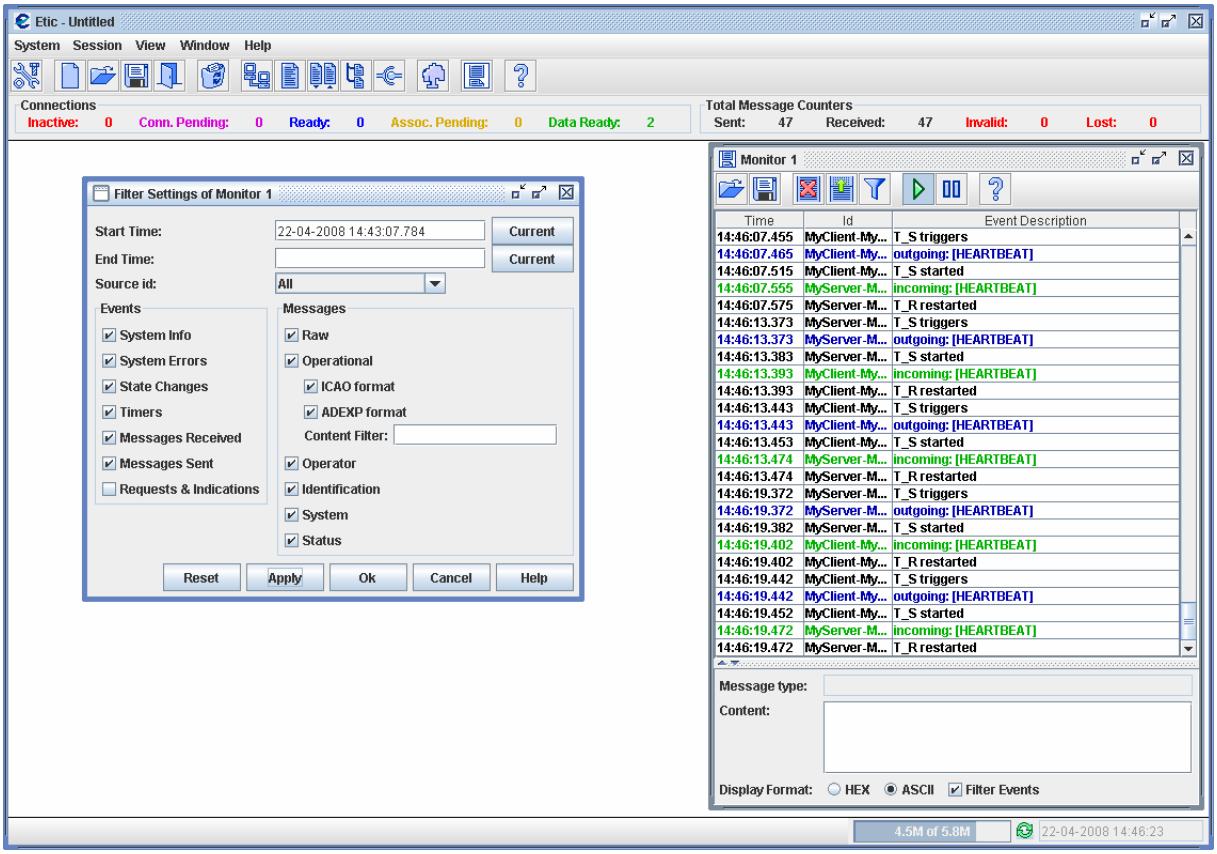
6. In the CP window select the entry that starts with MyClient and set the target state to be DATA_READY to start the FMTP association



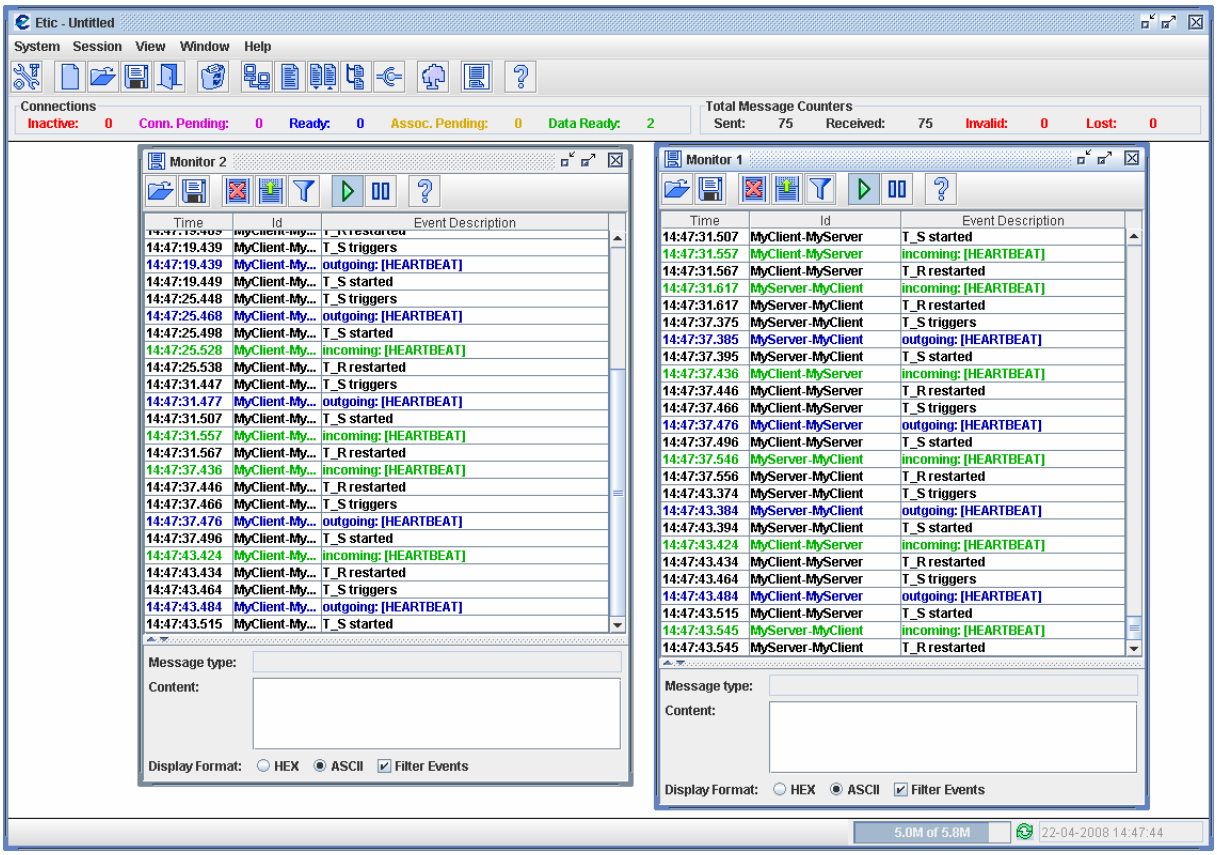
7. The states of both entries in the CP window should move to DATA_READY and the Monitor window should display the exchange of HEARTBEAT messages every 6 seconds



8. Open the filter settings of the Monitor window with the button  and select the Timers check-box. Click apply to note the changes in the monitor window



9. Open a new monitor window and set the filter settings to track the timer events for MyClient only

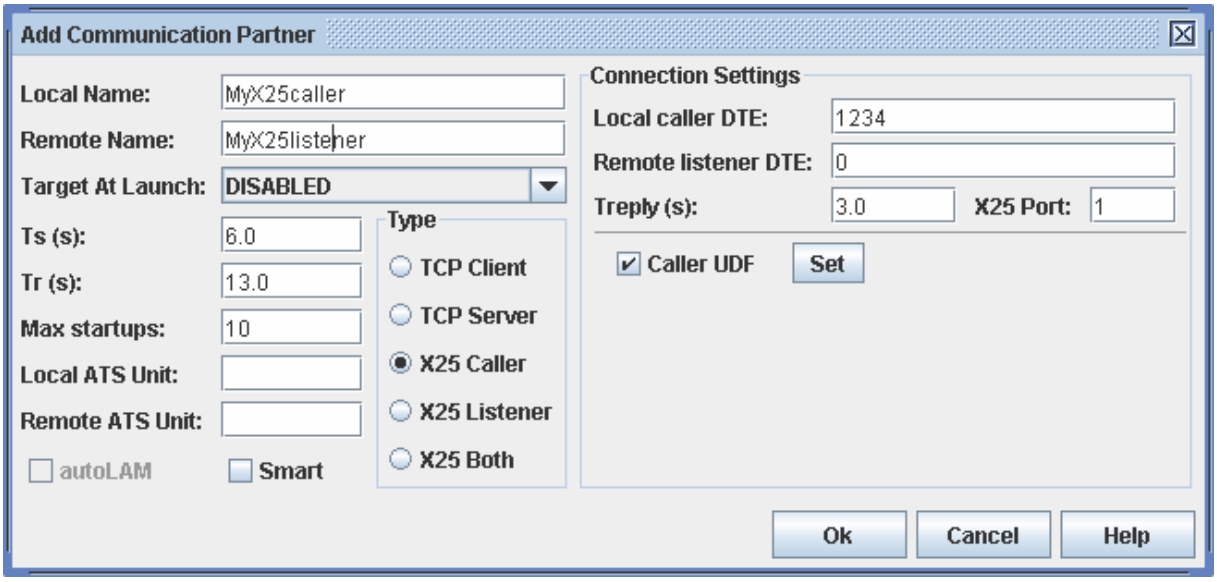


11.2 FDE ICD Part 1 connection

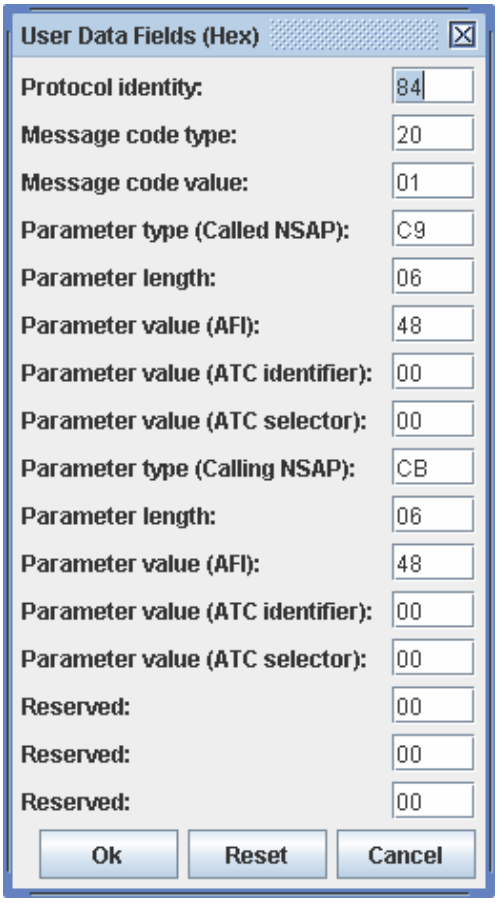
This is a step by step approach to launch a communication session between ETIC and a remote FDE ICD Part 1 Communication Partner.

1. Start a new ETIC session and select the CP edition window 

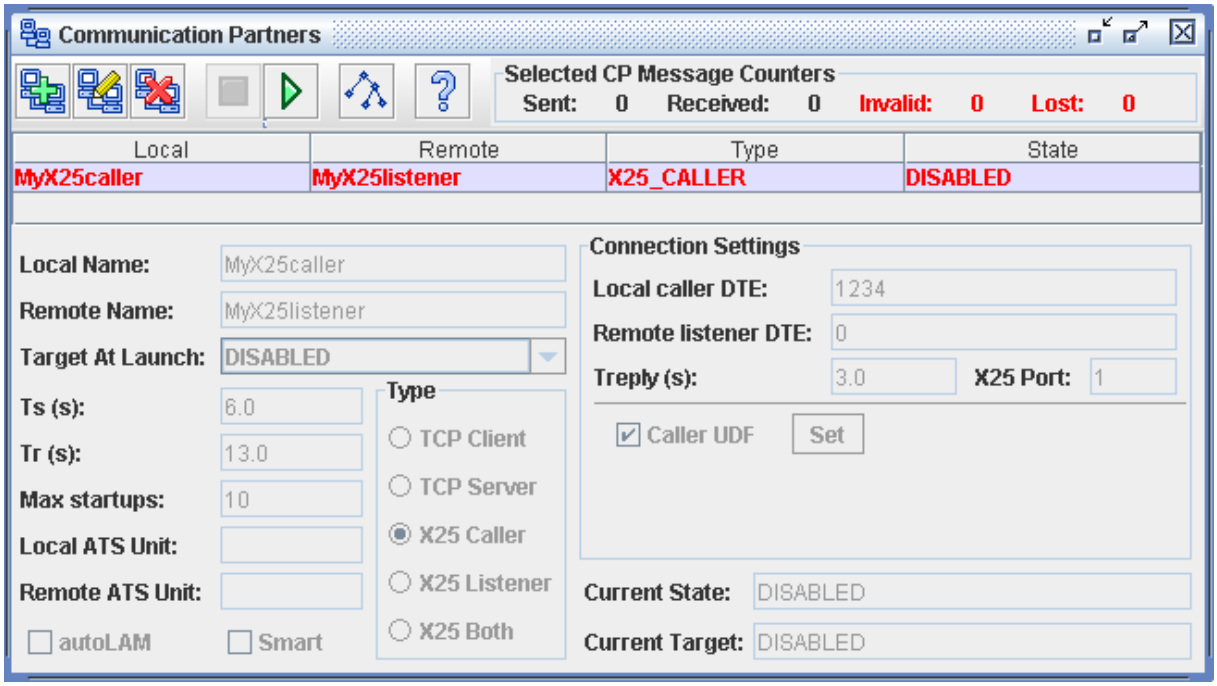
2. Select the  icon to create an entry as below. You must enter the appropriate local and remote DTE addresses.



3. Select the Set button to edit the user data field of the X.25 connection request packet and adapt them to your needs. Then click OK and close the previous CP editor window by clicking OK



4. In the CP windows select the entry that starts with MyX25caller and set the target state to be DATA_READY to start the X.25 connection establishment



11.3 Basic relays

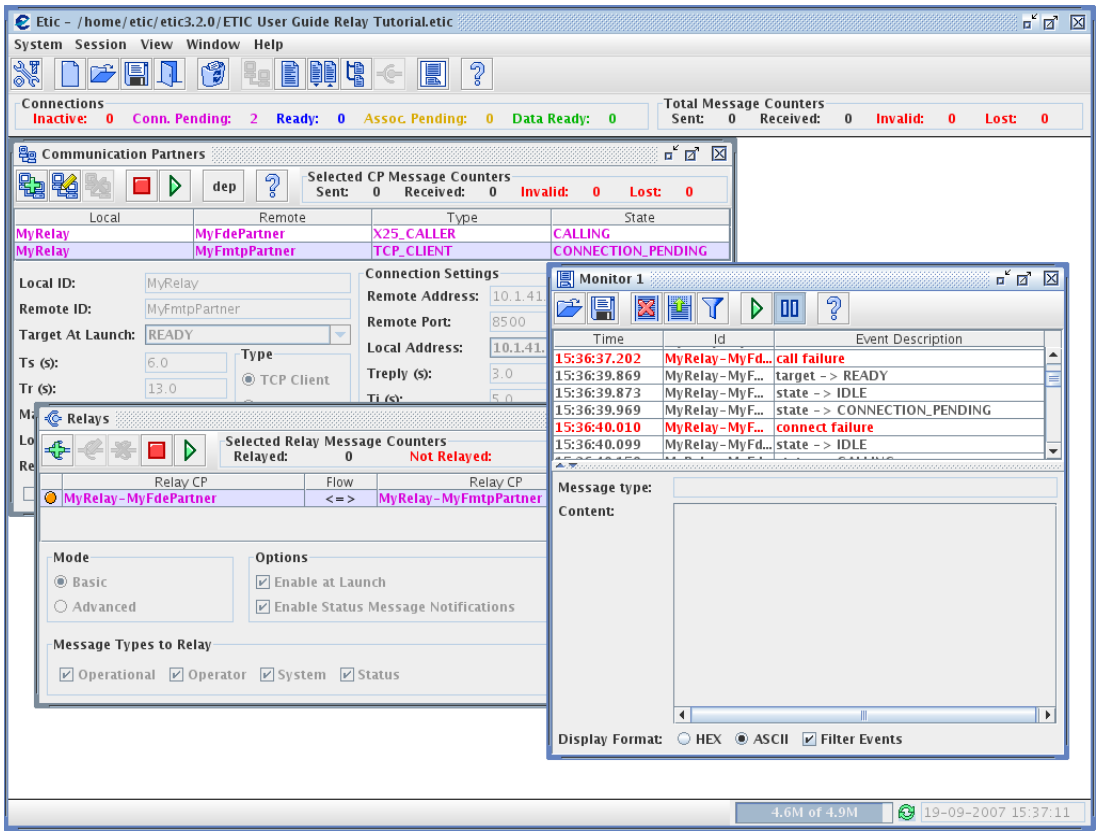
This is a step by step approach to establish an association between a remote a remote FDE ICD Part 1 Communication Partner named MyFdePartner and a remote FMTP Communication Partner named MyFmtpPartner via ETIC by defining a basic relay. The screenshots are based on ETIC release version 3.2.0 (ETIC 3.2.2 users may note the absence of the session tree icon and the modified dependency icon).

1. Start a new ETIC session and select the CP edition window 

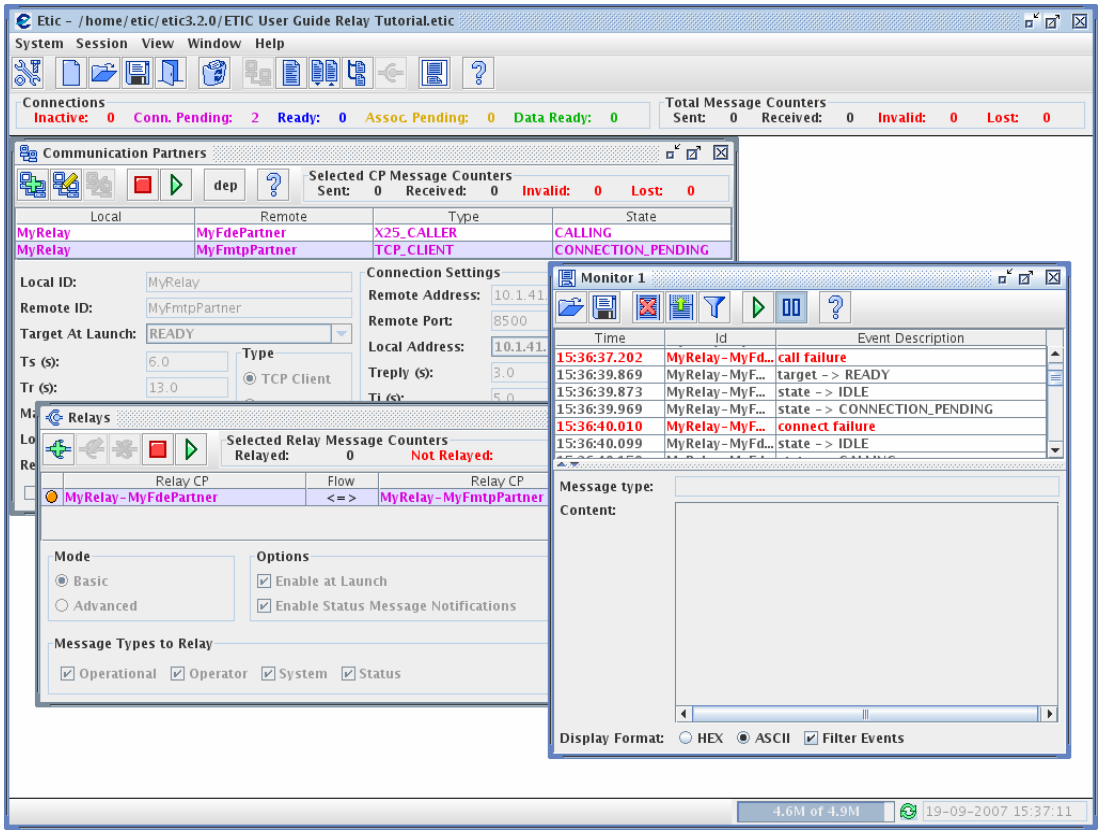
2. Select the  icon to create two CP definitions as below. You must enter the appropriate local and remote DTE and IP addresses.

3. Select the Relays window 

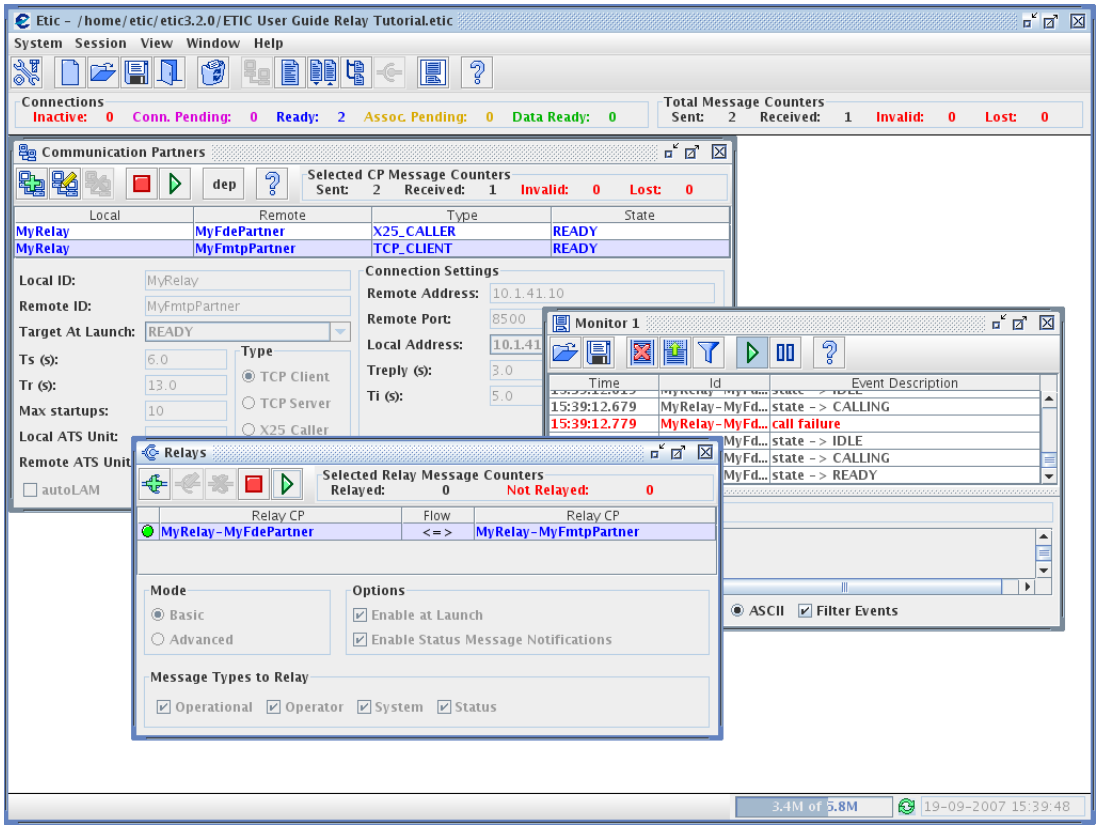
4. Select the  icon to create a new bi-directional basic relay as below. Customise the messages types that need to be relayed.



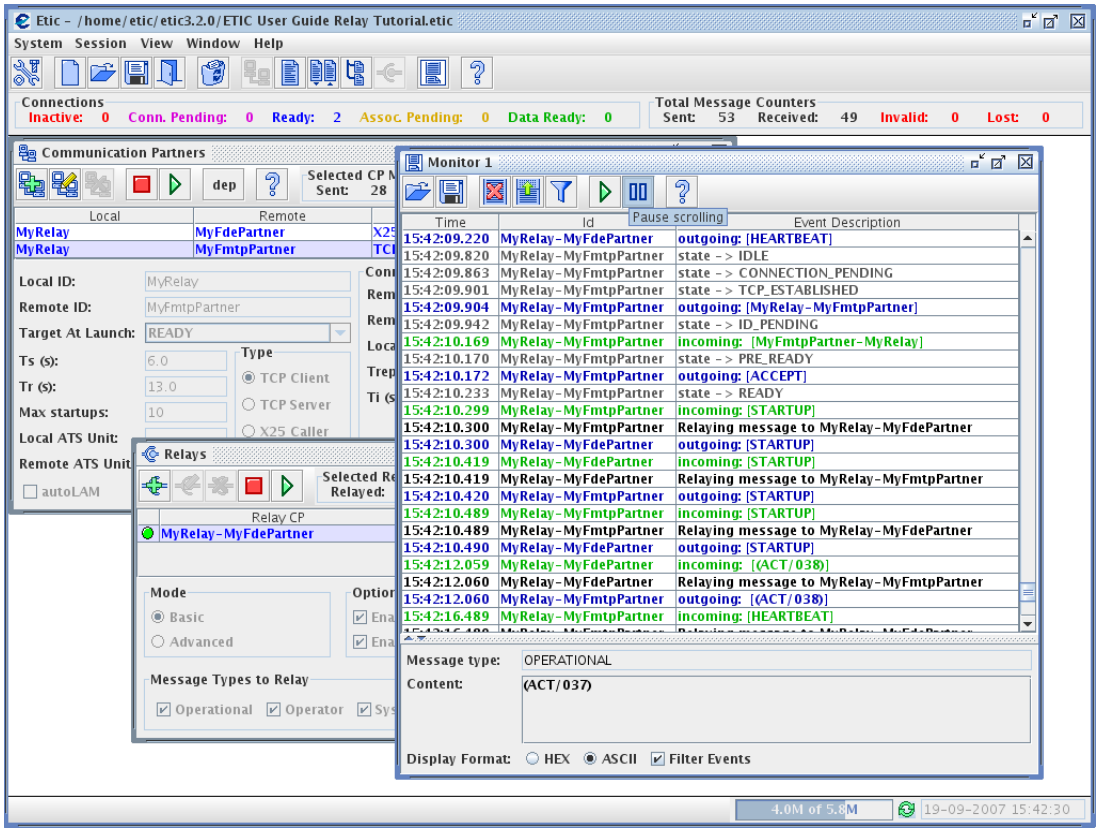
5. Start the relay using the  icon within the Relays window. The CPs will try to reach their READY state and the relay will indicate an orange status as no data can be relayed



6. When the external FMTP and FDE systems reach the READY state with ETIC, the relay will indicate a green status as data messages can be exchanged. It is recommended to configure the CPs that form a relay with a target at launch set to READY. Equally, the relay itself can be enabled at launch. These simple configuration settings greatly ease the use of the ETIC relay functionality.



7. When the external FMTP and FDE systems attempt to reach the DATA_READY state between each other they will exchange STARTUP messages. Once they are in DATA_READY, HEARTBEAT and data messages can be exchanged monitored. All these messages are relayed by the basic relay and can be monitored in real-time.



11.4 Relay message scripting

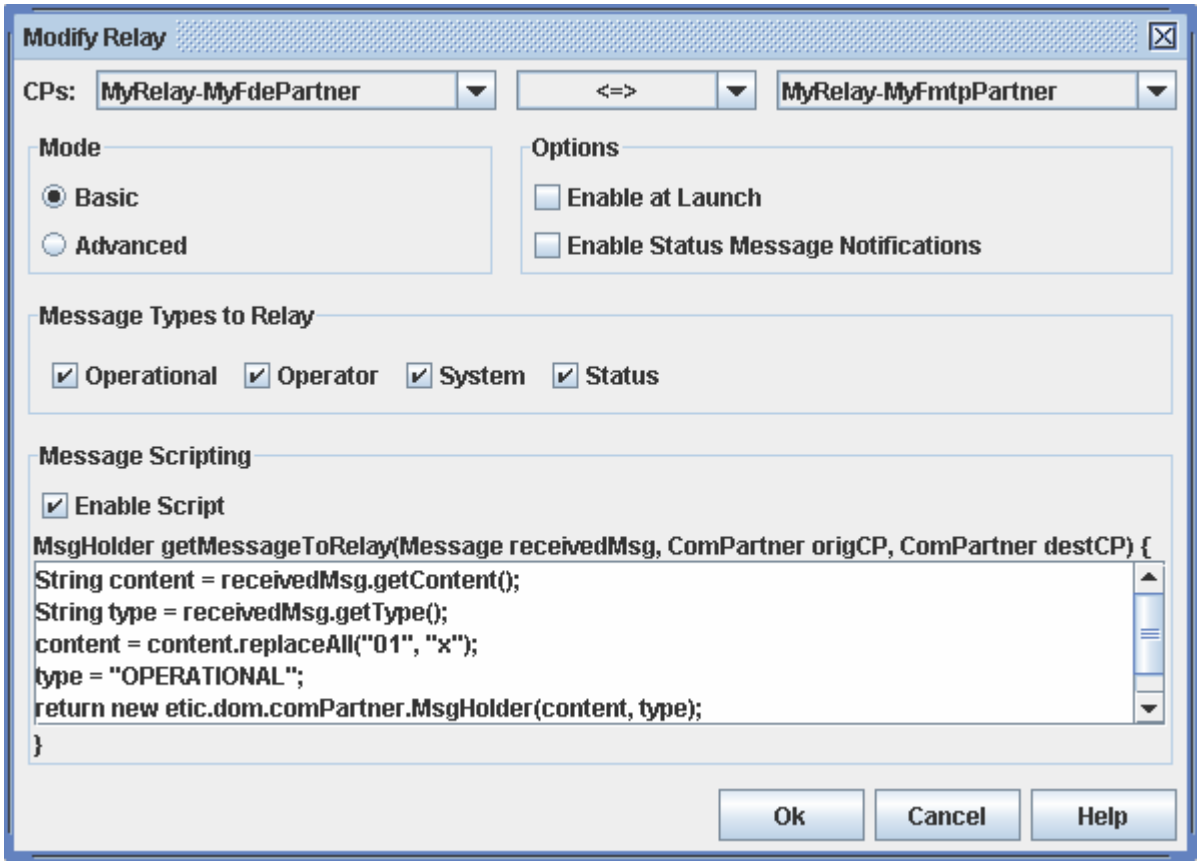
Within relay configurations, users can make use of the message script text box enter the code to be processed upon receipt of an incoming message. This code has to be written with groovy syntax which can be understood as a simplified version of Java. For more information about groovy and code rules, the following reference websites should be consulted http://groovy.codehaus.org/ and http://java.sun.com/j2se/1.5.0/docs/api/.

Each incoming message is stored into an internal structure (type, content) where type is the TYP value of the message and the content is the user data of the FDE ICD or FMTP message. Valid type values are OPERATIONAL, OPERATOR, IDENTIFICATION, SYSTEM and STATUS. The message script will interact with this structure and will return a new message to be relayed. If content or type of the new message is null, the received message will not be relayed. Furthermore, scripts can also integrate scripts based on the concerned communication partner

Below are three cases to describe how groovy can be used to modify incoming messages, make relay decisions and access communication partner details.

11.4.1 Modifying message content

In this case, the user has defined a script that replaces all occurrences of string “01” by string “x” and ensures that the messages type is set to OPERATIONAL before it is relayed.




```
String content = receivedMsg.getContent();                    (1)
String type = receivedMsg.getType();                          (2)
content = content.replaceAll("01", "x");                      (3)
```

```
type = "OPERATIONAL";                                          (4)
return new etic.dom.comPartner.MsgHolder(content, type);       (5)
```

Line (1): the content of the incoming message is stored in the String variable named content
Line (2): the type of the incoming is stored in the String variable named type
Line (3): replace all occurrences of "01" by "x" in the String named content
Line (4): forces the type of the received message to OPERATIONAL
Line (5): returns a new message with updated content and type

11.4.2 Modifying message content and relay decision

In this case, for any incoming ABI or ACT in ICAO format, the user needs to substitute certain characters by others, remove certain characters and only relay the message depending on the value of a given field. For example, in the below message:

```
(ABIM/RO001-AMM253/A7012-LMML-BNE/1221F350-EGBB-9/B757/M-
15/N0480F390 UB4 ROE UB4 BPK UB3 HON-80/N-81/W/EQ Y/NO)
```

They need to:

- replace M/RO by M/R (and also incoming messages with M/RF);
- remove -80/N; and
- to detect the second flight level, if the flight level is below 245, relay the message if it is above do not relay the message.

This complex situation can be handled by ETIC with the following message script:

```
String content = receivedMsg.getContent();                          (1)
String type = receivedMsg.getType();                                (2)
if (content.startsWith("(ACT") || content.startsWith("(ABI")) {     (3)
  content = content.replaceAll("M/R0", "M/R");                      (4)
  content = content.replaceAll("M/RF", "M/R");                      (5)
  content = content.replaceAll("-80/N", "");                        (6)
  String[] split = content.split("N\\d\\d\\d\\dF\\d\\d\\d");        (7)
  String flightLevel = content.substring(split[0].length() + 6,    (8)
    split[0].length() + 9);
  Integer tempValue = Integer.parseInt(flightLevel);                (9)
  int flightLevelValue = tempValue.intValue();                      (10)
  if (flightLevelValue > 245) {                                     (11)
    content = null;                                                 (12)
  }                                                                 (13)
  return new etic.dom.comPartner.MsgHolder(content, type);          (14)
}                                                                   (15)
else {                                                              (16)
  return new etic.dom.comPartner.MsgHolder(content, type);          (17)
}
```

Lines (1) and (2) are the same as in the previous example;
Line (3) tests whether the incoming message starts with (ACT or (ABI;

- If the test fails, line (17) indicates that the incoming message is relayed as-is.
- If the test succeeds, the procedure continues at line (4);

Lines (4) to (6) indicate the String manipulations;
Line (7) a regular expression extracting part of the message containing one of the two flight levels;
Line (8) a further extraction of the flight level still in string format;
Lines (9) and (10) transformation of the flight level substring into an integer;

Line (11) tests whether flight level value is greater than 245;

- If the test is successful, the message should not be relayed so the line (12) sets the content to null.
- If the test fails, the transformed message is returned and relayed line (14).

11.4.3 Accessing communication partner details

With message scripting, it is also possible to access details of the communication partners that form the relay.

```
String localId = origCp.getLocalId();                    (1)
String remoteId = origCp.getRemoteId();                  (2)
String localATS = origCp.getLocalATS();                  (3)
String remoteATS = origCp.getRemoteATS();                (4)
String state = origCp.getState();                        (5)

String localId = destCp.getLocalId();                    (6)
```

The above lines indicate how to access the information of the communication partner that is the source of the incoming message. Line (1) and (2) provide access the local/remote IDs of the CP, line (3) and (4) to the ATS unit values and line (5) to the state of the CP.

Line (6) illustrates that same data can be accessed from the intended recipient of the relayed message.

It is not possible to modify communication partner values but the above details will allow users to further enhance their message scripts.

12 Troubleshooting and Reporting Bugs

12.1 Known issues

1. Main frame resizing (obvious on Linux and Solaris, less severe on Windows)
2. Online help: slow to appear the first time as it needs to be loaded into memory
3. JAVA 1.5.0/1.6.0 NIO class fails for IPv6 on Windows XP/2000/2003/Vista (bug not applicable to Mac OSX, Solaris or Linux). This bug has been registered by Sun and can be monitored on the Java Bug Database at http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6230761. You can freely sign up with Sun and vote for this bug to be corrected. Windows IPv6 support will be limited to Vista with the release of JRE 7.0.
4. If ETIC X.25 support was selected during Windows installation but there is no Eiconcard at ETIC launch, a Pop-up will display with the message “missing eicon.dll” (click OK to continue using ETIC).
5. The Windows installer does not support Windows Vista.
6. Support for Java 1.6/1.7 will be validated for a future release of ETIC.

12.2 Troubleshooting and customisation with ETIC

Customisation is possible by editing the etic.lax file which is a textual file of runtime options.

- *lax.stdout.redirect=, lax.stderr.redirect=*
  - Empty (default when installed)

    - ***console***
    - Filename

- lax.nl.java.option.additional:
    - Set the Java memory heap: -Xmx512m (set max heap to ½ Gb)
    - Enable assertions: –ea

12.3 Trouble report

ETIC may have software bugs/issues. If ETIC does not appear to function in an expected way, this can be reported to the ETIC support team by email to etic@eurocontrol.int. In order to ascertain the exact nature of the problem, a report with all the following details should be included:

- A detailed and comprehensive description of the problem. In particular, the session file definition should be attached to the report as well as the system log file.
- The frequency at which the problem occurs: always, often, rarely or not reproducible.
- If the problem is not reproducible, what series of actions triggered the problem.
- The configuration of the ETIC platform, i.e. the ETIC version, the underlying OS and the JVM version. This information data is available from the *System/System Info*.
- For X.25 related problems:
    - Ensure availability of X.25 from *System/System Information*
    - The EICON card model and driver version
- If ETIC ends abnormally, a file named `etic_abnormal_termination_ID.txt` (where ID is an identification number) can be found in the installation directory of ETIC (or in the running directory if you are not using the ETIC standard launcher). This should also be attached to the report.

========== END OF DOCUMENT ==========