TECHNICAL REPORT

SINTEF Energy Research

Address: NO-7465 Trondheim, NORWAY
Reception: Sem Sælands vei 11
Telephone: +47 73 59 72 00
Telefax: +47 73 59 72 50

www.energy.sintef.no

Enterprise No.:
NO 939 350 675 MVA

SUBJECT/TASK (title)

ELCBAS/SEA for Windows Administrators Guide

CONTRIBUTOR(S)

Tormod Lund, Nils Eggen, Birger Stene

CLIENTS(S)

Joint Project: ABB Kraft AS, Siemens AS, Sintef Energy Research AS, Statnett SF

TR NO.	DATE	CLIENT'S REF.	PROJECT NO.
TR A5835	2003-12-22		12X246

ELECTRONIC FILE CODE	RESPONSIBLE (NAME, SIGN.)	CLASSIFICATION
030620bs13462	Ove Grande	Unrestricted

ISBN N0.	REPORT TYPE	RESEARCH DIRECTOR (NAME, SIGN)	COPIES	PAGES
82-594-2508-4		Petter Støa	10	22

DIVISION	LOCATION	LOCAL FAX
Energy Systems	Sem Sælandsv. 11	+47 73 59 72 50

RESULT (summary)

This report describes the installation, initial configuration and maintenance of the Elcbas/SEA software for the Windows platform.

Please see SINTEF's homepage at: http://www.sintef.no/ELCOM-90. From here you can download the latest version of all relevant documents as pdf-files for free.

KEYWORDS

SELECTED BY AUTHOR(S)	Data Communication	Communication Protocols
	Control Centres	ELCOM-90

Document Identity: TR A5835 Revision: 04

Technical Reference:

Approved by:

Attested by:

Date	Revision	Synopsis
2002-09-17	01	New Document (pre-release)
2003-12-22	02	First Official Release
2004-03-02	03	Updated After Comments
2011-01-10	04	Updated for Elcbas V6.5

Table of Contents

Table of Figures

Preface

Purpose

The purpose of this document is to describe the installation, initial configuration and maintenance of the Elcbas/SEA software for the Windows platform.

Intended Audience

This is a technical document. Some familiarity with Elcom concepts is assumed.

Associated Documents

1. TR A4124 -- ELCOM-90 Application Service Element, Users Manual
2. TR A5834 – Elcbas/SEA Configuration Guide
3. TR A5833 – Elcbas/SEA Programmers Guide
4. TR A3825 – Elcom User Element Conventions

Consult [4] for a more extensive list of Elcom documentation.

Acronyms

SEA	Simplified Elcom API
API	Application Programming Interface
SEAIN	The initiator in the Elcbas/SEA system
SEARS	The responder in the Elcbas/SEA system

Trademarks

None

1. Introduction

1.1 Overview of the Elcbas Software

An Elcbas installation consists of the following programs:

- e90.exe – The Elcom-90 protocol provider
- superv.exe – The Elcom-90 supervisor
- ElcCfg.exe – The Elcbas/SEA Configuration program
- seain.exe – The Elcbas/SEA initiator user element
- sears.exe – The Elcbas/SEA responder user element
- elcman.exe – The Elcom Manager umbrella service
- savelogs.exe – The Log Snapshot program
- ad_tls.exe – The Elcom Adaptation program for TLS (encrypted commonuication)

As well as a number of optional test programs.

2. Installation

2.1 System Requirements

The Elcbas/SEA runs Windows XP and newer, client and server opertiong systems.

In general, the most recent service pack is recommended.

Hardware requirements will depend on the size of the Elcom configuration, but in most cases a pc meeting the minimum requirement for the selected OS will run Elcbas as well.

2.2 Installation Procedure

The Elcbas/SEA Software uses Windows Installer version 3. Start by running the ElcbasSetup.msi file.

Note: for first-time installation on a PC, you should normally ensure that the Visual Studio redistributable is present in the same directory as the kit (vcredist_x86.exe).

Elcbas is currently only supplied as a 32-bit windows application, but will run on 64-bit OS variants.

Once the installation is started, you should get this window:

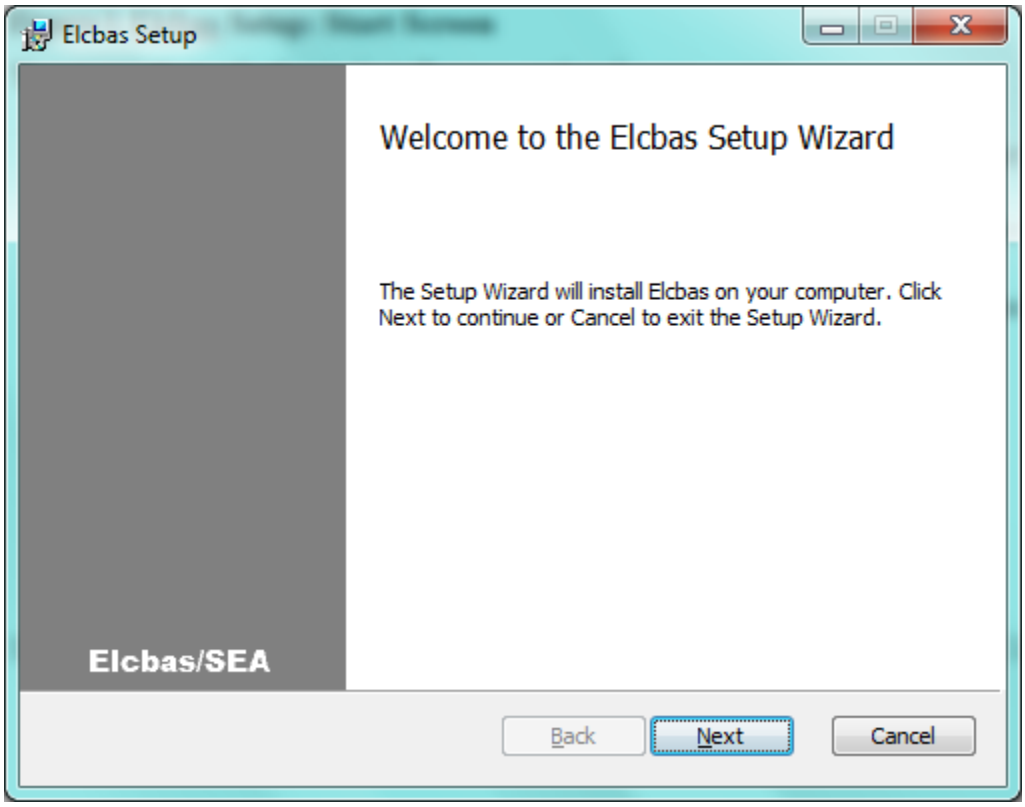

Figure 1 Elcbas Setup: Start Screen

The next screen after this may show a license agreement, you should review this and select accept to continue.

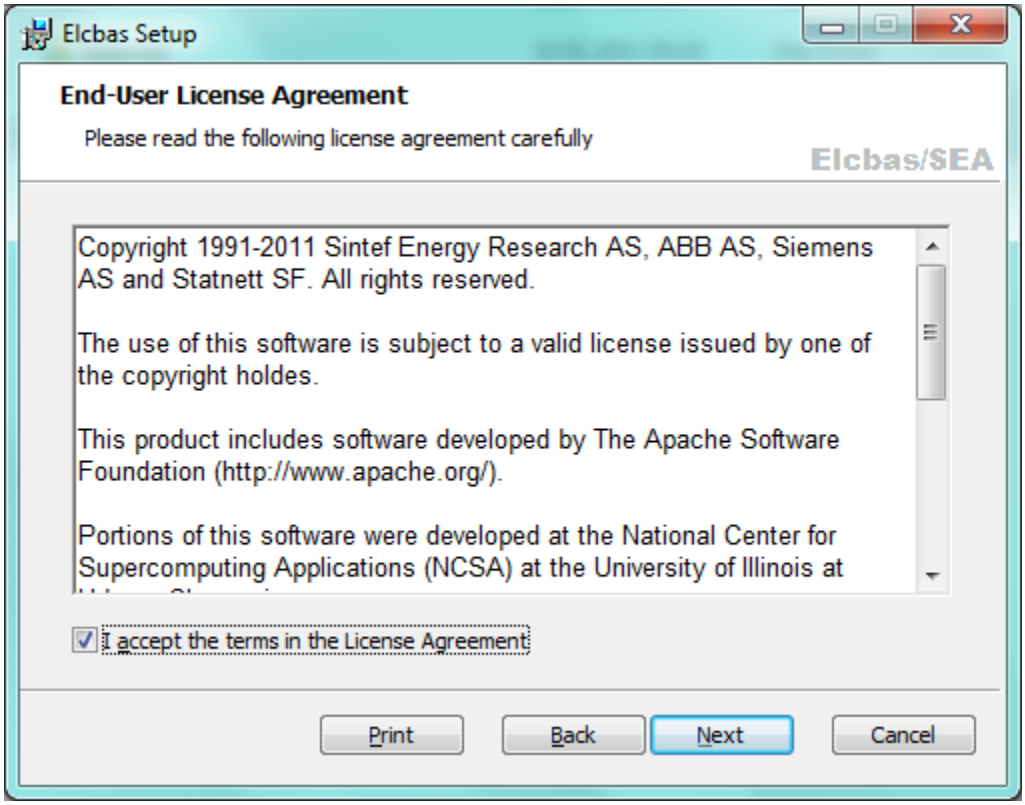

Figure 2 Elcbas Setup: License agreement

The setup type screen allows you to choose between the following setup types:

- Typical – installs the common options for communication without support for development or encryption.
- Custom – allows you to select at a detailed level, as well as to specify the directories for the program files and the log and configuration files.
- Complete – installs everything.

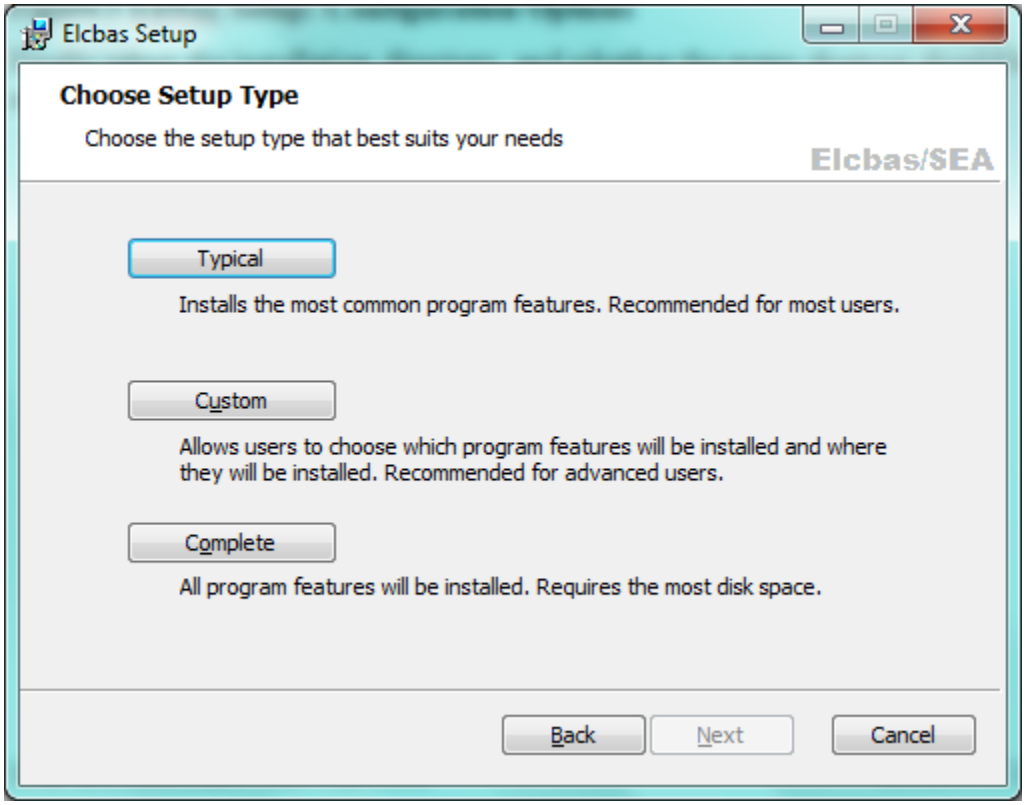

Figure 3 Elcbas Setup: Setup Type

Selecting the Custom setup type brings up the following dialog.

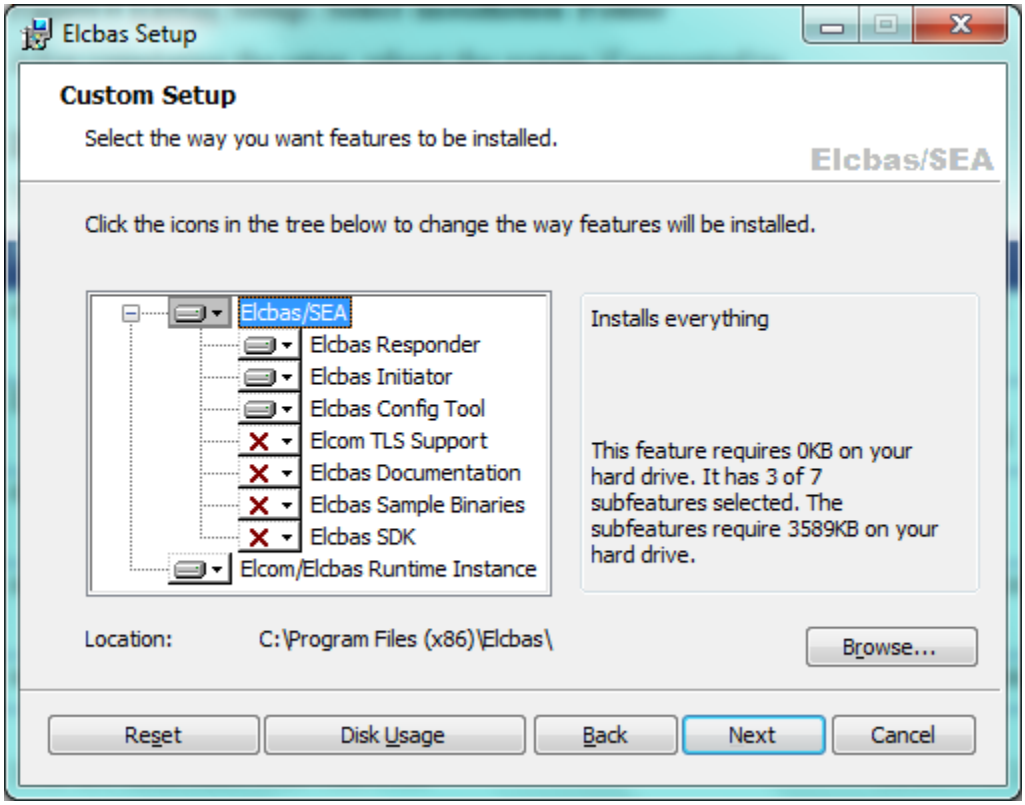

Figure 4 Elcbas Setup: Custom Setup

The features are as follows:

- Elcbas Responder – used to supply data to partners using Elcom.
- Elcbas Initiator – used to receive data from partners using Elcom.
- Elcbas Config Tool – used to modify the Elcbas configuration files.
- Elcom TLS Support – used to add support for encrypted Elcom communication.
- Elcbas Documentation – the Elcbas documentation set in PDF format.
- Elcbas Sample Binaries – simple test clients to test Elcbas (the source for these are included with the SDK).
- Elcbas SDK – support for developing Elcbas clients. Currently C/C++ client development is supported.
- Elcom/Elcbas Runtime Instance – used to control the run-time directory for Elcbas. See note below.

After this, a confirmation screen is shown, as well as a completion screen when the installation is done.

Reboot the computer if prompted to.

2.3 Files Installed

The following files are installed on the system (for a complete install):

Name	Directory	Type	Usage
ad_tls.exe	bin	Program	Elcom-90 TLS adaptation process
e90.exe	bin	Program	Elcom-90 protocol provider
ElcCfg.exe	bin	Program	Elcbas/SEA Configuration Tool
elcman.exe	bin	Program	Elcom Manager Umbrella service
seain.exe	bin	Program	Elcbas/SEA Initiator user element
sears.exe	bin	Program	Elcbas/SEA Responder user element
superv.exe	bin	Program	Elcom-90 supervisor
t_in_cmd.exe	bin	Program	Test program for sending commands and setpoints
t_in_data.exe	bin	Program	Test program for receiving data
t_rs_cmd.exe	bin	Program	Test program for receiving commands and setpoints
t_rs_data.exe	bin	Program	Test program for sending data
t_rs_uns.exe	bin	Program	Test program for sending data changes
savelogs.exe	bin	Program	Program to create a snapshot of log and configuration files.
openssl.exe	bin	Program	Openssl utility program.
curses.dll	bin	Library	Emulation of unix curses (for superv.exe)
elc_alib.dll	bin	Library	Elcom-90 alib
libxml2.dll	bin	Library	XML Library used for Elcbas
log4cpp.dll	bin	Library	Logging library used

			for Elcbas
libapr-1.dll	bin	Library	Apache portable runtime
libeay32.dll	bin	Library	Openssl library
ssleay32.dll	bin	Library	Openssl library
log4cxx.dll	bin	Library	Log4cxx logging library
seapi.dll	bin	Library	Elcbas/SEA seapi library
NTEventLogAppender.dll	bin	Library	log4cpp support
zi32z64.dll	bin	Library	Zip file support
ELCBAS-TRACE.CFG	templates	Config	Elcbase trace configuration
elcman.ini	templates	Config	Configuration for elcman.exe
seain.ecf	templates	Config	Elcbas Configuration for Initiator
sears.ecf	templates	Config	Elcbas Configuration for Responder
elc-conf	templates	Config	Elcom-90 provider configuration
ad_tls.log_config	templates	Config	ad_tls log4cxx configuration
seain.log_config	templates	Config	Elcbas-90 Initiator log4cpp configuration
sears.log_config	templates	Config	Elcbas-90 Initiator log4cpp configuration
savelogs.ini	templates	Config	Configuration for the savelogs program
ELCBAS-IM-SERVER.STD	templates	Config	Elcbas Initiator internal configuration
ELCBAS-INITIATOR.STD	templates	Config	Elcbas Initiator internal configuration
ELCBAS-RM-SERVER.STD	templates	Config	Elcbas Responder internal configuration
ELCBAS-	templates	Config	Elcbas Responder

RESPONDER.STD			internal configuration
elcom.txt	templates	Config	Elcom-90 static text file
error.txt	templates	Config	Elcom-90 static text file
general.txt	templates	Config	Elcom-90 static text file
elc-xcp	templates	Config	Sample xcp configuration file (for TLS)
openssl.cnf	templates	Config	Sample openssl configuration file (for openssl.exe)
seapub.h	include	Header	Elcbas/SEA SEAPI header file
seapi.lib	lib	Link Library	Elcbas/SEA link library (for seapi.dll)

Additionally, sample source code is installed under the subdirectory samples (if selected), and PDF documentation is installed in the doc subdirectory.

Configuration files are installed to the templates directory, and copied to the designated runtime directory, which may be specified in the custom screen during setup, and which deafults to %ALLUSERSPROFILE%\Elcbas\run.

If an older version of Elcbas is upgraded, and the configuration was in the bin directory, this setting is maintained.

In general, an upgrade will not change these files, but the newest version will always be available in the templates directory.

Some or all of the files in templates may be replaced by supplying replacements in a templates subdirectory in the directory where the kit is kept.

2.4 Upgrade Procedure

Use the same kit for upgrading the software. To install an older version, uninstall first (from the control panel, Add/remove software).

At upgrade, essential configuration files will not be replaced in the bin directory, unless the force update flag is set.

Reboot the computer if prompted to do so.

After the upgrade, the Elcom Manager service must be restarted manually.

3. Configuration

3.1 Elcom Configuration

The elcom configuration file, elc-conf and must be inspected and modified according to your needs. Please refer to [1].

3.2 Service Configuration

Normally, the Elcbas system is run as a Windows service, under the control of the elcman program. The elcman program uses a configuration file, elcman.ini, to determine which programs to run and other options.

As installed, elcman will run a complete Elcbas system (initiator, responder and elcom provider) under the service name Elcom Manager.

The elcman.ini file is in windows .ini-file format, and has the following options:

Section	Setting	Default	Remarks
global	processes	0	The number of programs being run under elcman.
	interval	5	Interval between check of child status. In seconds.
	restarts	3	Maximum number of restart attempts for a failed process.
	service_name	ElcomManager	The short-form service name for the service. Only used at install/uninstall of the service.
	display_name	Elcom Manager	The long-form or display name of the service. Only used at install/uninstall of the service.
	auto_start	0	Set to one if the service should be configured to autostart. Only used at install/uninstall of service.
	env_count	0	A count of environment variables to set/clear prior to starting any children. Each environment variable is specified in an [env_n] section, with n starting at 1, up to and including the value of this setting.

env_*n*	name	“”	The name of an environment variable to set/clear
	value	“”	The value of the environment variable. If blank or missing, the environment variable will be cleared.
proc*n*	name	“”	Name for the command (used in logging)
	command	“”	The command string to run
	logfile	“”	The name of the log file to be used for standard output/standard error.
	group	0	The ‘program group’ for this process. If one program in a group stops, all the programs in the group will be restarted.
	stopmethod	0	Method for controlled stop of this process. Current support is: 0 (default) – use TerminateProcess 1 – use a Win32 event object, which is set by elcman to request a stop.
	stopevent	“”	The name of the stop event object when stopmethod is 1.
	gracetime	5	The delay imposed after a failure before a restart is attempted (in seconds).

The Elcom Manager service may be further modified in the control panel applet, e.g. changing the autostart option, and possibly the account the service is run from (defaults to LocalSystem).

Note that such changes will currently be lost when upgrading or reinstalling.

3.3 Log Configuration

The log functions of Elcbas is currently being migrated to use the log4cpp library, which is based on the commonly used log4j library for Java.

Currently the log files ELCBAS-IN-T.LOG, ELCBAS-RS-T.LOG, ELCBAS-IN-EVT.LOG. ELCBAS-RS-EVT.LOG, ELCBAS-IN-API.LOG and ELCBAS-RS-API.LOG are created through this library.

The log4cpp library is configured by the files seain.log_config and sears.log_config, respectively. Please refer to comments in the files, consult log4cpp documentation or contact your vendor if you need to change these files.

3.4 Savelogs Configuration

The savelogs program is configured with an ini-file, savelogs. ini. The directory for the zipped archive, as well as the pattern for the file name may be set, as well as a list of files to include. Add new files by adding file_n keys, but note that the file numbers must be an unbroken sequence. Any files listed but not present are quietly ignored when creating the archive.

4. Operation and Supervision

4.1 Starting and Stopping the Software

When running Elcbas as a service, this service can be started and stopped as any other windows service, e.g.:

- By using the services control panel applet (under Administrative Tools in Windows 2000 and newer).
- By using the services MMC snap-in in Windows 2000 and newer (available e.g. in the Computer Management Console from Administrative Tools, or Manage from the context menu on My Computer).
- By using the NET command from a command line window:

  net start ElcomManager (to start the service)

  net stop ElcomManager (to stop the service)

  net start (to list active services)

  The command is not case sensistive, and the long service name may be used if quoted.

The Elcom Manager service may also be paused/continued, but this is only intended for testing (a pause implies suspending the main thread of all child processes).

The processes making up an Elcbas System may also be run individually (seain. sears and e90 are console mode programs). Note that e90 must be started prior to any of seain/sears.

4.2 Using the Log Files

The following log files exist in a running Elcbas system:

Name	Source	Usage
ELCBAS-IN-API.LOG	Initiator	Detailed log for Elcom alib api-calls. May be turned off from the Elcbas configuration. Rotates with a backup copy using the suffix .1
ELCBAS-IN-T.LOG	Initiator	Debug/Trace information from the initiator. Rotates with a backup copy using the suffix .1
ELCBAS-IN-EVT.LOG	Initiator	Initiator event log. Covers communication events internally and externally. May be redirected to the

		Windows event log.
ELCBAS-IN-1.ERR ELCBAS-IN-2.ERR	Initiator	Initiator Error log.
ELCBAS-RS-API.LOG	Responder	Detailed log for Elcom alib api-calls. May be turned off from the Elcbas configuration. Rotates with a backup copy using the suffix .1
ELCBAS-RS-T.LOG	Responder	Debug/Trace information from the responder. Rotates with a backup copy using the suffix .1
ELCBAS-RS-EVT.LOG	Responder	Responder event log. Covers communication events internally and externally. May be redirected to the Windows event log.
ELCBAS-RS-1.ERR ELCBAS-RS-2.ERR	Responder	Responder Error log.
e90.log	Provider	Standard output/error from the e90.exe when run from elcman.exe.
seain.log	Initiator	Standard output/error from the seain.exe when run from elcman.exe.
sears.log	Responder	Standard output/error from the sears.exe when run from elcman.exe.
ad_tls.log	TLS adaptation	Log from the TLS adaptation (if used). Rotates with a backup copy using the suffix .1

A snapshot (as a zip-file) of the Elcbas log and configuration files may be taken with the tool savelogs. If run interactively, files may be removed from the resulting archive (but not added). Savelogs may also be run quietly with the /q command line option. The savelogs archives are by default created in the savelogs subdirectory of the bin directory.

4.3 Using Operating System Tools To Monitor the Software

Some useful commands are:

- The control panel services applet, to verify that the Elcom Manager service is running.
- Alternatively, the ‘net start’ command without a service name lists running services.

- The 'netstat' command lists active TCP/IP connections, and is useful to verify if Elcom is connected. Look for the specified Elcom port (e.g. 5997):
  - In the local address column for connections where the local system is responder.
  - In The remote address column for connections where the local system is initiator.
- The task manager can be used to verify if the processes of a running Elcom system are active:
  - Look for e90.exe (protocol), elcman.exe (the service/watchdog program), seain.exe (Initiator) and sears.exe (Responder).

4.4 Using Other Tools

Some useful third-party tools are available:

- From sysinternals (http://www.sysinternals.com):
  - Process explorer – a better task manager
  - Tcpview – a dynamic TCP/IP connection viewer
  - Dbgview – for dynamic log viewing (if configured in the .log_config files).
- A useful, freeware network sniffer, ethereal, is available from http://www.ethereal.com

4.5 Using the Test Programs

The following are brief usage descriptions of the test programs included in the kit (if selected when installing):

4.5.1 t_in_data

This program is a simple program to receive data from the initiator and list to a console window. Must be configured with the api-name t_in_dat.

4.5.2 t_rs_data

This program is a simple test data provider for use with the responder. It is run in a console window, and may be triggered with a second instance running in a different window, e.g. to change data values.

Usage:

'server' mode: t_rs_data [name] [-1] [-v n]

where:

name – is an alternative api-name (t_rs_dat is the default), for testing existing configurations.

-1 – indicates SEAPI responder Version 1 mode (use to simulate older clients).

-v n – indicates the verbosity level (current range 0 to 2)

When running like this, t_rs_data will supply initial data consisting of numbers corresponding to group index for float/integer groups, and alternative on/off for status groups.

The 'client' usage is as follows: t_rs_data [-q|-uf|-us|-ui] [object_name] [value] [quality].

Examples:

t_rs_data -q – list all data currently requested

t_rs_data -q OBJ – list data for object OBJ

t_rs_data -uf OBJ 2.5 – updates object OBJ to have value 2.5

t_rs_data -uf OBJ 1.0 144 – updates object OBJ to have value 1.0 and quality 144 (held).

t_rs_data -us OBJ2 2 – updates object OBJ2 to have value 2, ON f this is a status

4.5.3 t_rs_uns

This program is used to send unsolicited data in SEAPI Version 1 mode (for the responder). Used in conjunction with t_rs_data with the –1 switch.

Example:

t_rs_uns OBJ 1 2.5

OBJ is the object name, 1 is the elcom data type (floating in this case) and 2.5 is the value.

4.5.4 t_in_cmd

This program is used to send a command or a setpoint through the initiator and wait for the response.

Usage: t_in_cmd [name] [-c|-f|-i] [-I|-C|-E|-H] –v n OBJ1 val1 [OBJn valn…]

where:

name – is an alternative API-name (t_in_cmd is the default)

-c – this is a command

-f – this is a floating point setpoint

-i – this is an integer setpoint

-I – send as immediate mode command (default)

-C – send as check before execute

-E – send as execute

-H – send as inhibit

-v n gives the verbosity level (0 or 1)

OBJ1 val1 is the object name and value (note: for commands use 1 for on and 0 for off)

For setpoints, several name value pairs may be given, provided this matches the configuration (multiple setpoints may be given in one elcom telegram if they are at consecutive indices in one elcom group).

Examples:

t_in_cmd –c COMMAND 1

t_in_cmd –f SETP 2.5

4.5.5 t_rs_cmd

This program is used to respond to commands received through the responder. It has the following usage: t_rs_cmd [name] [-t|-b|-n|-i|-a|-] [-1][-v [n]]

Where:

name is an alternative API name

-t – let the commands time out (give no response)

-b – respond that commands are blocked

-n – respond with no connection

-i – respond with illegal command value

-a – respond with no authority

-o – respond with illegal object

-1 – open the API in version 1 mode (use to simulate older clients).

-v n gives the verbosity level (0 or 1)

The default is to respond with success.