

Table of Contents

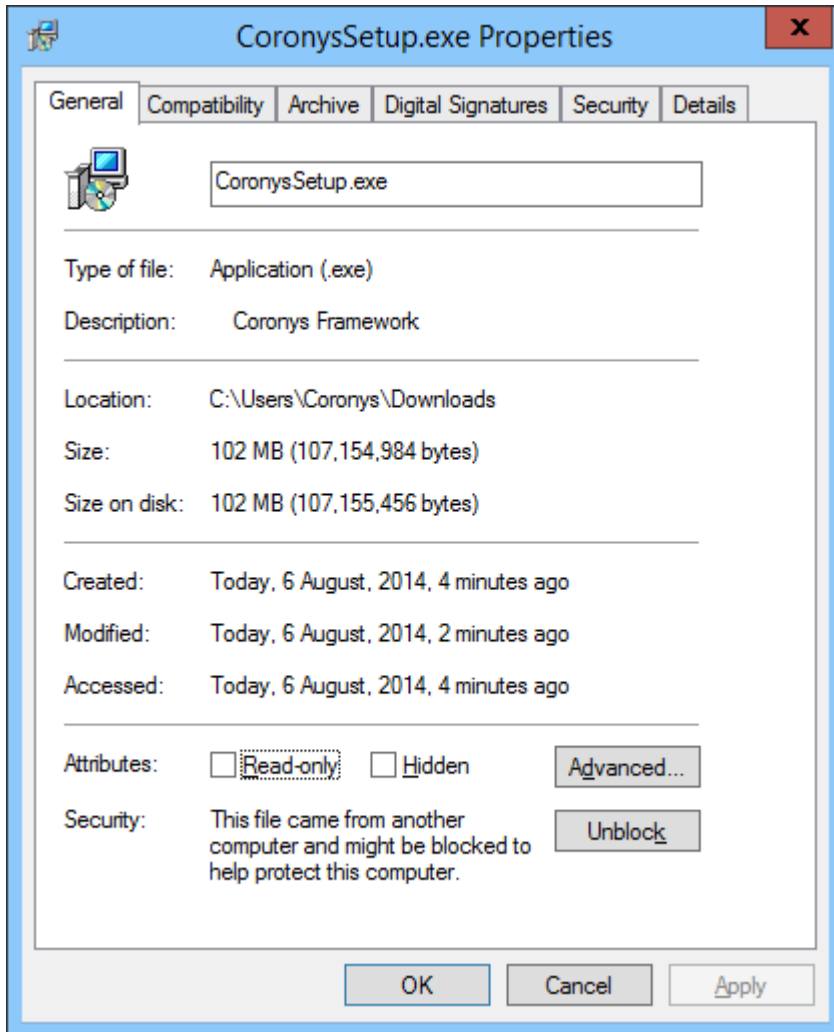
ETS DEMO SETUP	2
Quick review	2
Prepare Demo	5
License Server Installation	5
Recommended start up documents	6
Overview	7
Create a demo project.....	8
Create a work space.....	8
Insert test steps.....	9
Create Test Case	11
Execute Test Case	12
Save Test Case.....	12
Internal Terminal	12
The ESL (Easy scripting language)	13
Debugging.....	13
The Main framework	14
ETS Demo Limitations	15

ETS DEMO SETUP

ETS demo is very easy to install.

Run CoronysSetup.exe application to setup the Coronys framework.

In case of windows 8 you might get a message that it protects the file before running.



Just unblock and run it.

Quick review

After setup finishes two folders are created.

C:\ETS folder with the ETS package;

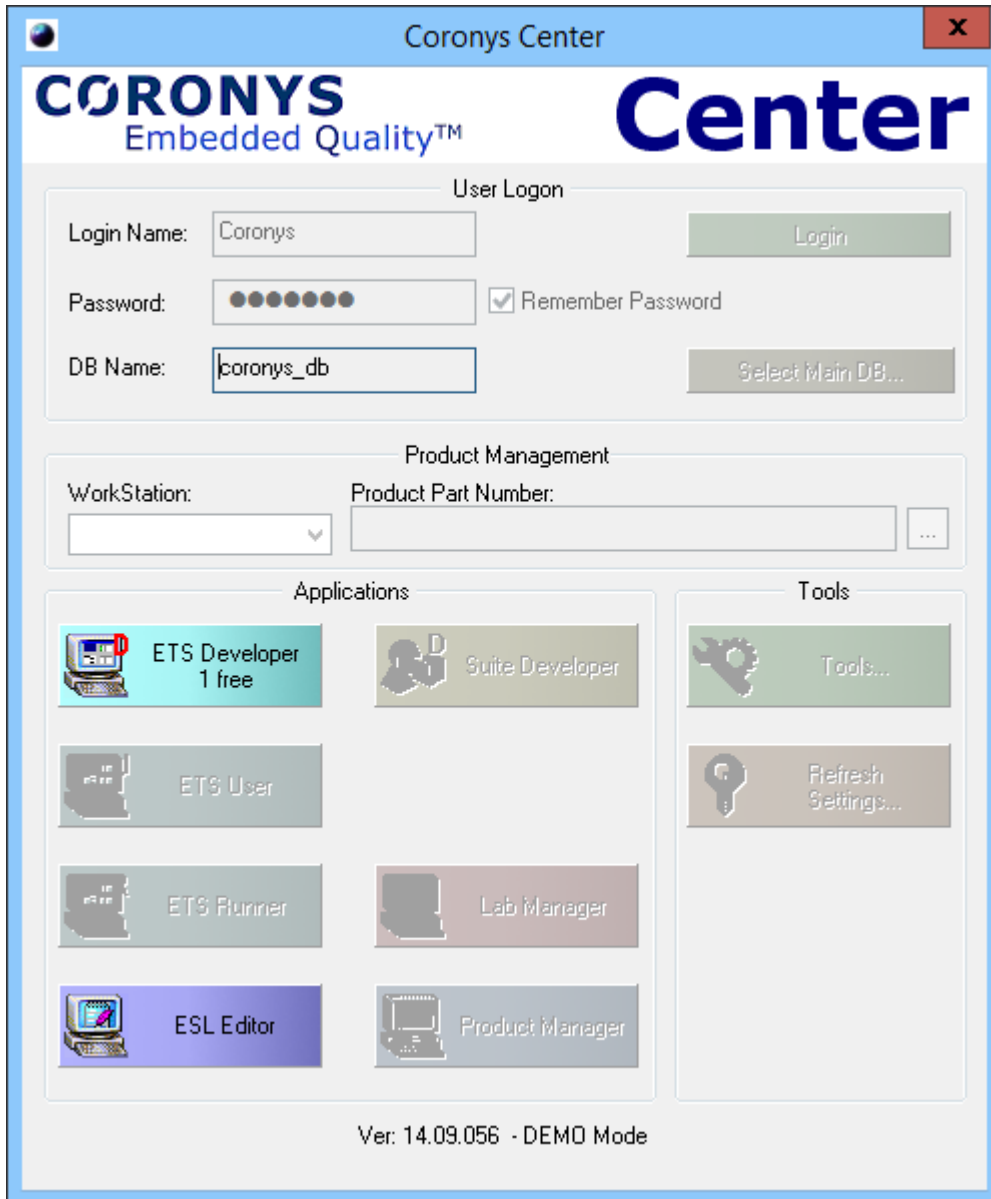
C:\ETS_Workspaces folder that holds your projects.



On a desktop you will find the Coronys starter application or you can start from:

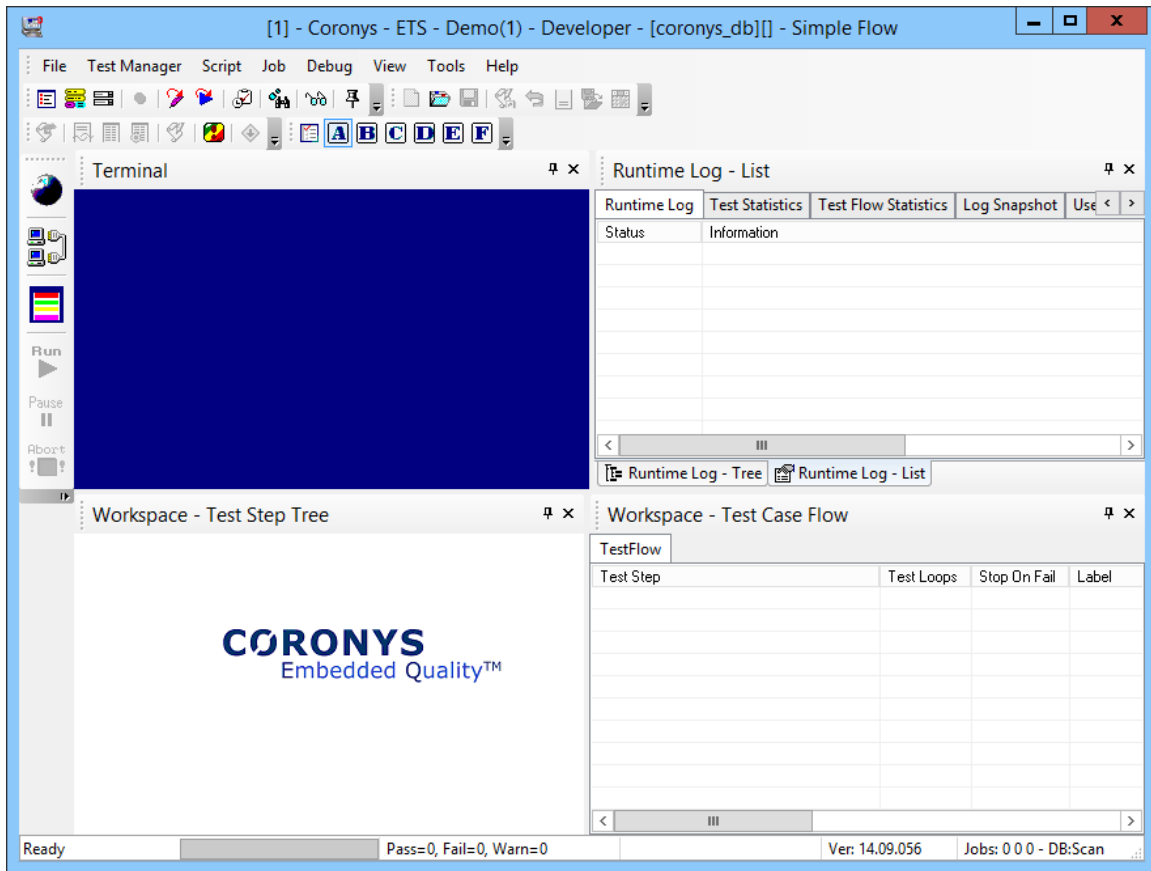
Start->Coronys menu.

Double click to open the starter application:



NOTE: the demo mode requires a waiting of about 20 seconds to enter to demo.

After about 10 seconds the ETS developer icon is enabled. Click it to start the ETS framework.



As you can see by default the ETS opens four windows.

1. On upper left an online terminal to use with serial port or telnet or telnet SSH.
2. On lower left a test tree window to build your test steps (see tutorial later).
3. On lower right a test case builder.
4. On upper right a test execution online log.
5. For easy start we recommend to read the manuals from c:\ets\manuals.
6. For easy training start the c:\ets\manuals\ETS-Programmers Guide.pdf

Prepare Demo

In a folder c:\ ETS_Workspaces you will find a Samples folder.

This folder holds a samples workspace.

Open the workspace from ETS menu: File->Open Workspace...

Select **WS_Tree.tdf** file. The workspace is loaded.

License Server Installation

To install the license server contact Coronys support at support@coronys.com.

Recommended start up documents

For more information start reading the following PDF files:

CoronysCenter.pdf	The ETS starter application.
ETS-Users Guide.pdf	Describes the ETS parts.
ETS-Programmers Guide.pdf	Builds a demo project and shows how to make and use test cases.
ESL-Editor.pdf	Describes the ESL editor parts.
ESL-Syntax.pdf	Describes the internal Easy Scripting Language (ESL). The ESL is the interface to the ETS runtime engine.
More...	There are more PDF files related to ETS modules. All documents are located at c:\ets\manuals folder.

Overview

ETS is a test engine framework. The ETS is designed as an operating system, it means, once the ETS is executing, you can do online, manual and background tests at the same time.

Explanation:

You can open your serial port using the internal built-in terminal window. In this stage you can operate your device manually. You can write a simple script that can program the ETS to listen to a data pattern; each pattern creates an event that can for example run a script.

By design the ETS is designed to create a batch of automation tests (Test Cases) and execute them automatically. The automatic tests can be executed at the same time with manual and waiting events. **This working mode gives you the power to catch unexpected error messages from your tested device any time** (while the ETS is running scripts or in an idle mode).

The following sections describe the following:

Create Demo Project	Creates the first work space to work on it, creating test steps and test case flow, executing and saving of created test case.
Working with the terminal	Working with the internal terminal manually and using a script.
The ESL (Easy scripting language)	This is your interface (API) to the ETS engine.
Debugging	You can debug your script using breakpoint and stepping.
The Test Case	The test case is a script that it is generated by the ETS generator.
ETS Demo Limitations	See the limitation list

Table 1 - References

ETS-Users Guide.pdf	Describes the ETS menus, windows and dialog boxes.
ETS-Programmers Guide.pdf	Directs step by step a simple automation project.
ESL-Editor.pdf	Describes the ESL Editor menus, windows and dialog boxes.
ESL-Syntax.pdf	Describes the ESL syntax and use.

Create a demo project

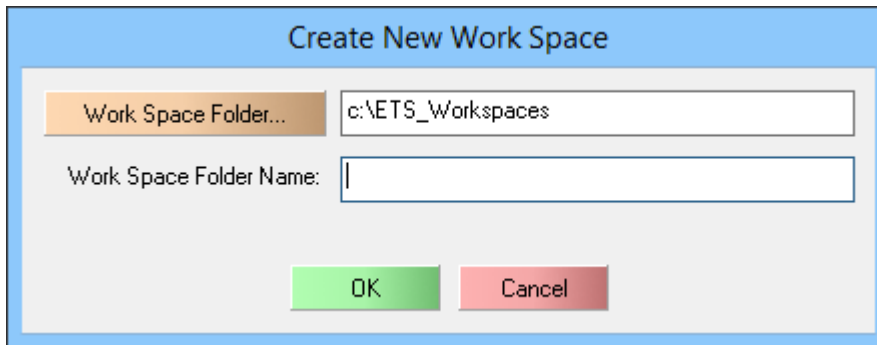
This demo project will build mainly to show the ETS options. A real project can be defined and created using Coronys consultants.

Create a work space

The first thing we have to do is to create a work space (like a solution or project).

The work space holds all necessary files and scripts. The ETS work space files are text based and are friendly to use with any version control application. With the full version you can control the version control from ETS menus.

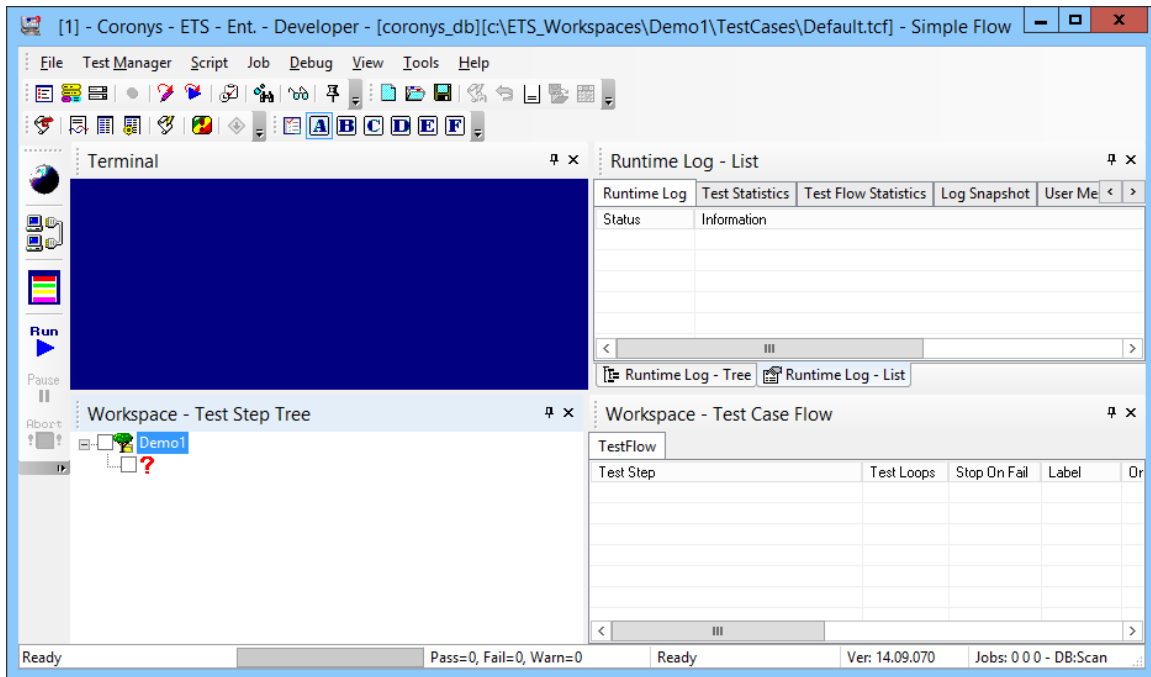
To create the work space, select “File->Create New Work space...” from the main menu.



The dialog box by default points to c:\ETS_Workspaces folder (you can change it).

Fill the “Work Space Folder Name:” with a name (for example demo1). A folder demo1 and some more files in it are created.

In the ETS bottom left window a new tree is created.

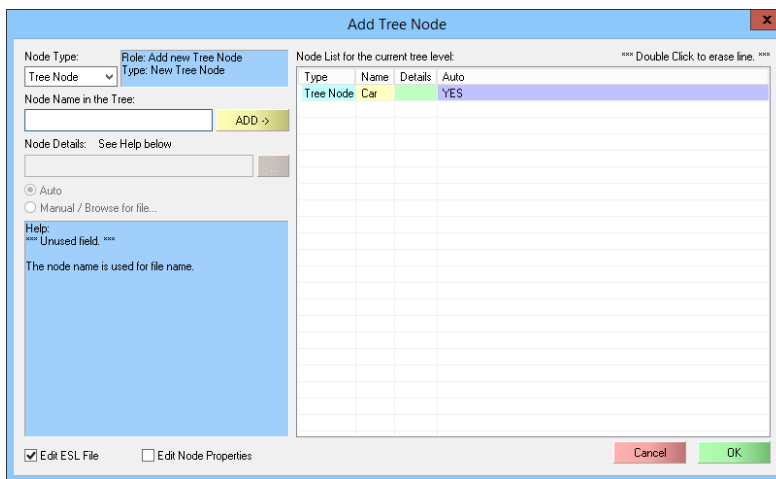


Insert test steps

The created tree holds our test steps. Each test step is an executable script. About scripting – read the ESL syntax pdf.

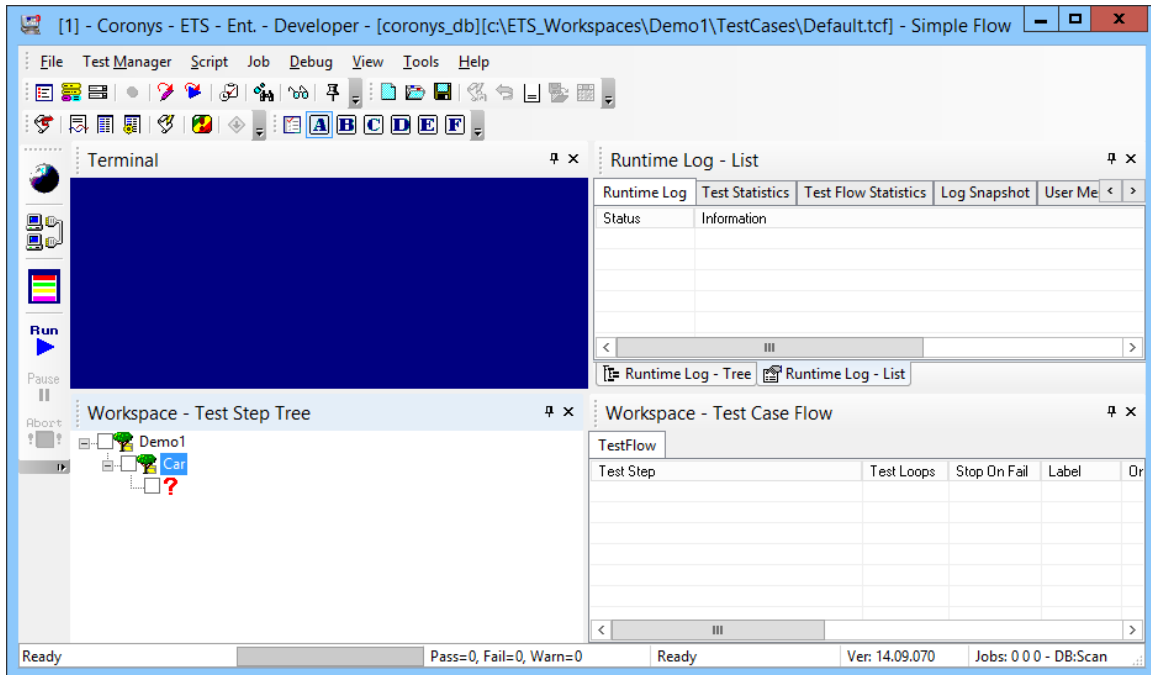
To add new tree node, open the main tree node, a red question mark is shown. Right click on it and select “Edit Node->Add New...” option.

On **Node Type** select “Tree Node” (the last one), On **Node Name** write the node name (e.g. Car) and click on “ADD->” button. In this way you can add more siblings; click OK.



The tree will show the car node. The tree nodes are like folders. If you open it again you will see a new red question mark.

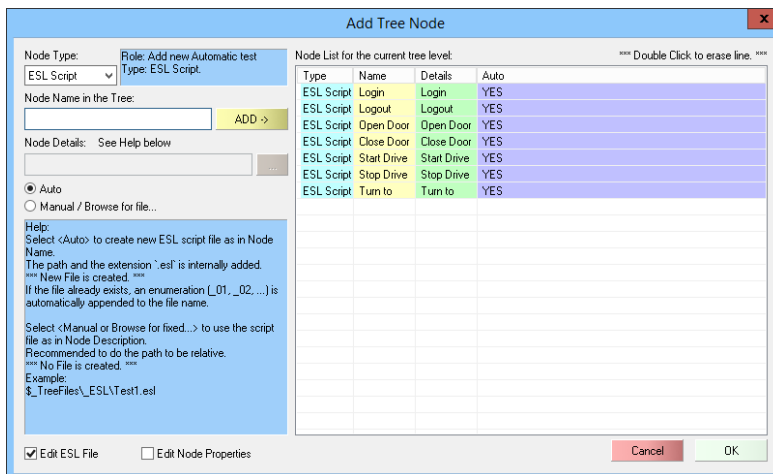
Here we will add some scripts.

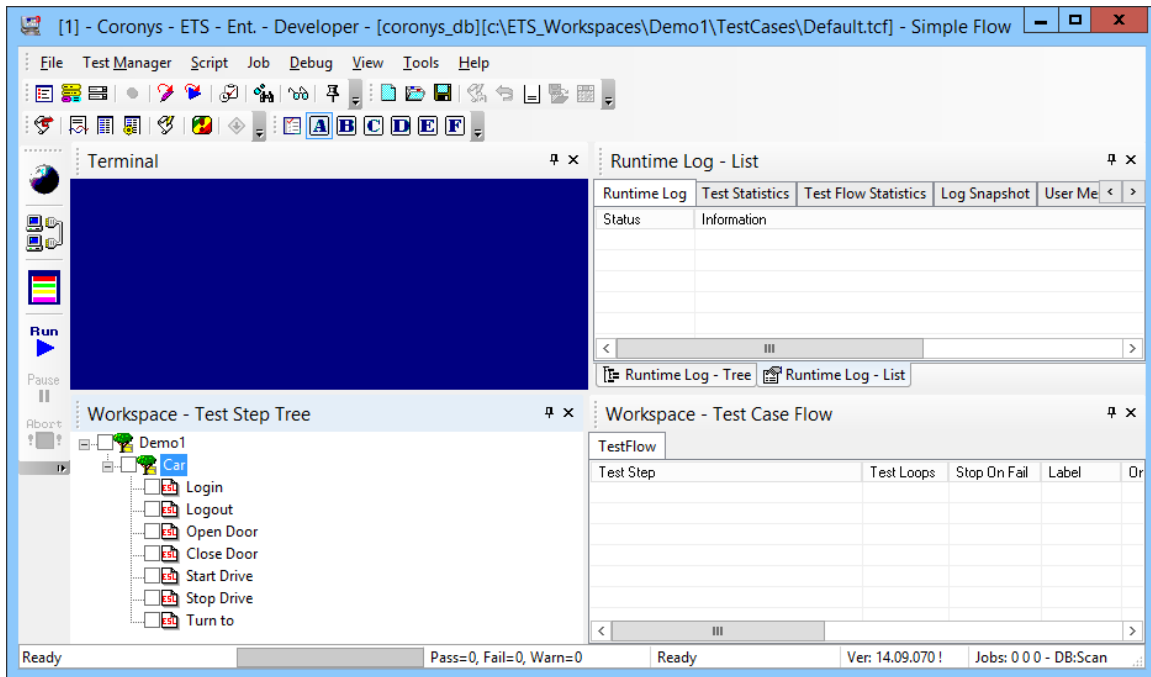


To add new tree node, open the main tree node, a red question mark is shown. Right click on it and select “Edit Node->Add New...” option.

On **Node Type** select “ESL Script” (the first one), On **Node Name** add the following node names:

Login, Logout, Open Door, Close Door, Start Drive, Stop Drive, ... and click OK.





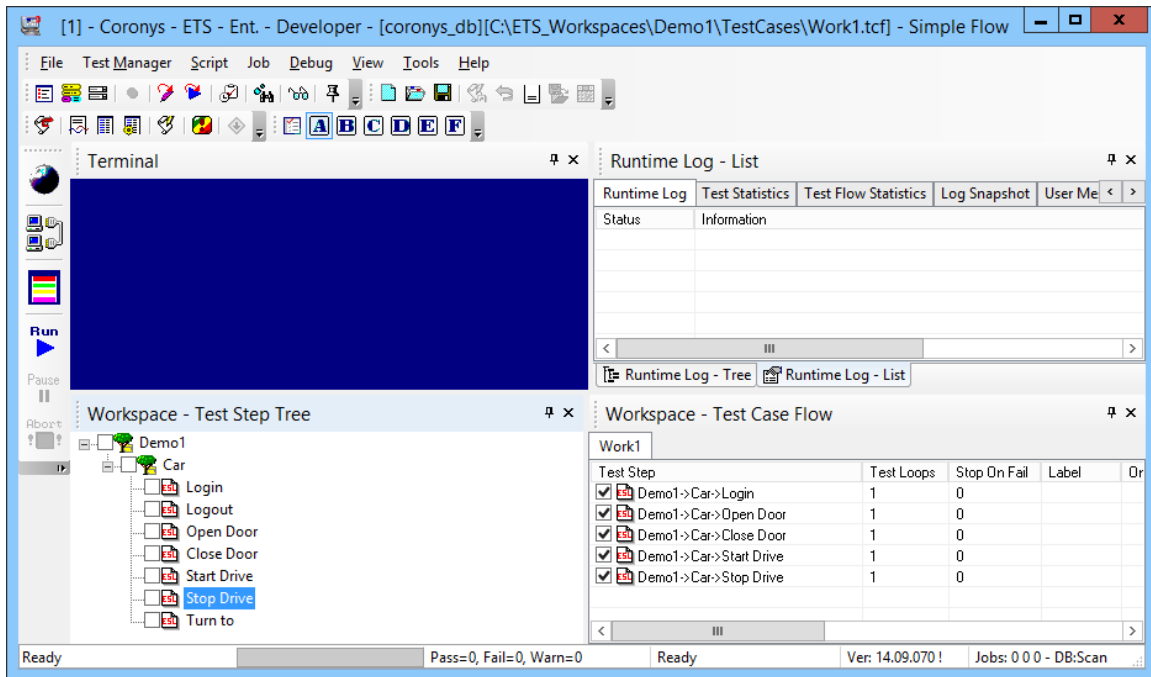
As you can see a new test steps are added, the ESL Editor is open and all out new scripts are ready to edit. In this stage we can minimize the editor.

Create Test Case

The next thing to do is to create a test case. The test case is a collection of test steps.

To create a new test case select "File->New Test case" from the main menu. Browse to our work space and go into "C:\ETS_Workspaces\Demo1\TestCases" folder. Write a file name (e.g. Work1) and save.

On the bottom right (Test Case Flow) window a new TAB with Work1 is created.



Now drag and drop from the tree the test steps to here. (in a demo version you can add up to 16 steps).

Execute Test Case

To execute the test case simply click on a Run Button (On the left bar). The execution log on the runtime log window is shown (the upper right window).

Save Test Case

To save the test case select “File->Save Test Case...”, a popup notification shows the saves file name.

Internal Terminal

The ETS internal terminal is ready to work programmed and manually. In the full version you are able to open unlimited terminals.

The terminal is used to open RS-232 connection, Telnet connection, Any TCP connection, UDP monitoring and SSH connection.

The connection can be established by right click into the terminal window, from the menu select “Settings->Interface Settings...”.

Once the connection is established, you can send from Keyboard or from script, the received data is displayed over the terminal.

To send a data from a script see the *Send* command.

Example: Send “dir^M”

Sends the “dir” followed New Line (Enter). Some connections needs LF+CR than use ^M^J.

You can also set an event that waits for a string. See *OnRxStringDo* command.

The ESL (Easy scripting language)

The ETS power is the scripting engine. Using the scripting you can made a great automation tasks.

The scripts are text based with a very easy syntax.

For full ESL syntax explanation, see *ESL_Syntax.pdf* file. Use the ESL editor to get the best

The ESL has blank lines, comment lines and command lines.

The comment lines begins with a semicolon “;” or with a double slash “//”.

A command line must begin with an ESL command or an assignment expression.

Example:

```
; This is an ESL comment line. It can follow ESL command
// this is also a comment line. It CAN'T follow ESL command
Declare a, b ; declare two local variables
a = 100 ; set 100 to a
b = a * 20 ; another assignment expression
Print b $ “^M^J” ; output the result to internal terminal with a LF+CR
End
```

Debugging

The ETS has internal ESL debugger.

To open the debugger window select “Debug->ESL Debugger...” from the ETS main menu.

There are several ways to start the ESL debugger.

1. From the ESL editor press F5, the ESL debugger pops up and enters step by step mode.
2. Using the Debug command or set a breakpoint.
3. In order to stop in a breakpoint or in a debug command, the ESL debugger must be at debug mode.

4. To set the ETS in debug mode, open the ESL debugger and check the “Debug Mode” on the bottom left.
5. After entering into debug mode, just step the commands.

For more debugger operations read [ETS_UsersGuide.pdf](#) and [ESL_Syntax.pdf](#).

The Main framework

The ETS has internal generator that creates a main framework (test case) script that does a lot of functionality for you.

To create it - just do the following:

1. Create scripts in a test step tree.
2. Select the scripts into the test case flow window.
3. Click the Run button; it will create a main script that will call all your scripts with a statistics that you can see it on a Runtime Log window (Upper right).
4. To make more complex flow, see the [ETS_ProgrammersGuide.pdf](#). There you can see looping, getting parameters from files...

ETS Demo Limitations

The ETS limitations usually are the windows limitations.

Number of instances in the same PC	1
Number of maximum variables	180
Number of maximum test flows	3
Number of maximum test flow lines	16
Inter-Process communication	Disabled
Structures	Disabled
Enums	Disabled
User defined menus	Disabled
GPIB commands	Disabled
Job execution	Disabled
Hash Tables	Disabled
Link Lists	Disabled
VBScript and JavaScript execution	Disabled
File change and track events	Disabled
Time and Timer events	Disabled
Link from runtime log to terminal logs	Disabled
Multiple Serials ports	Disabled, only the online is enabled
Multiple Telnets and SSH	Disabled, only the online is enabled

Telnet Server	Disabled
SNMP	Disabled
UDP server	Disabled
XML commands	Disabled
GUI management commands	Disabled
Most of system commands	Disabled
Data Base management commands	Disabled

The demo version has:

- Serial port or Telnet or telnet SSH – only 1 connection.
- Communication commands.
- String Manipulation commands.
- File manipulation commands.
- Script flow commands – If, For, While, Do, Foreach, CallTo, Gosub, Goto,...
- DLL commands
- Terminal event commands.
- Lookup Table commands.

NOTE: The ETS designed to work with windows 7 and above. In case of Windows XP the application is compatible but starting 2015 the support will be limited.

The setup is done from extracted folder.