

EC1000 API and Tools Software Installation

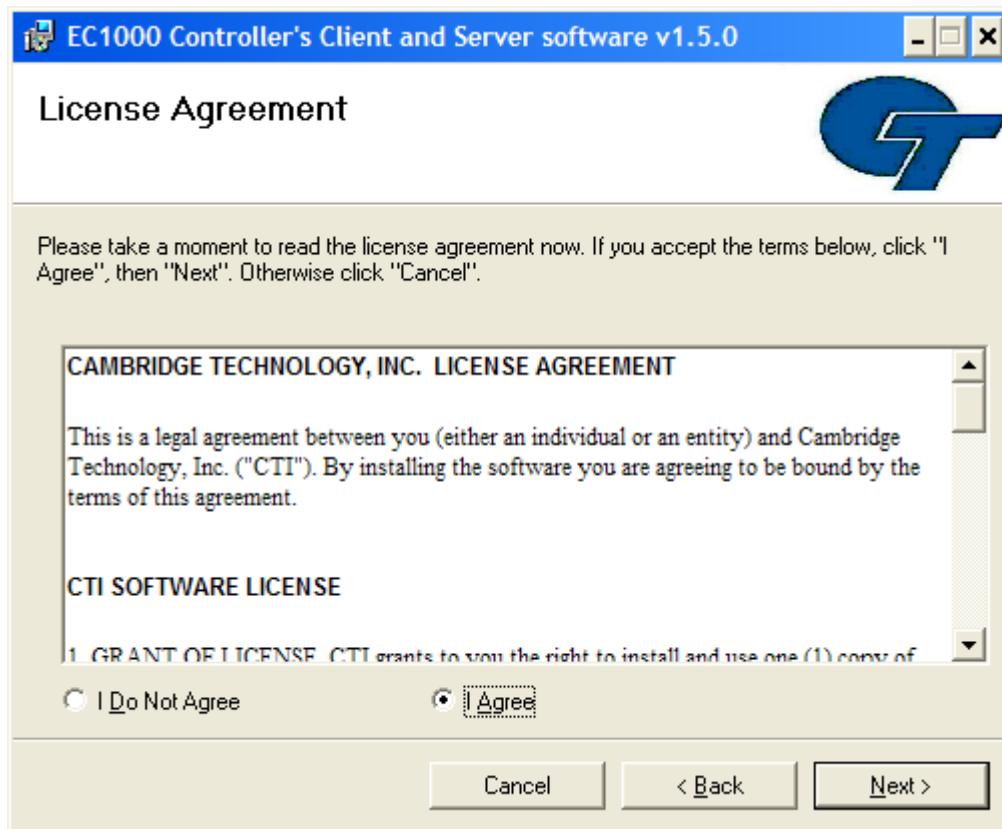
1 Getting started

The EC1000 software Application Programming Interface (API) and usage sample programs are installed on a user's computer from a CD shipped with the EC1000 module. The API requires the Microsoft .NET version 2.0 Framework to be installed on the target machine. The installation does not Framework if it does not exist.

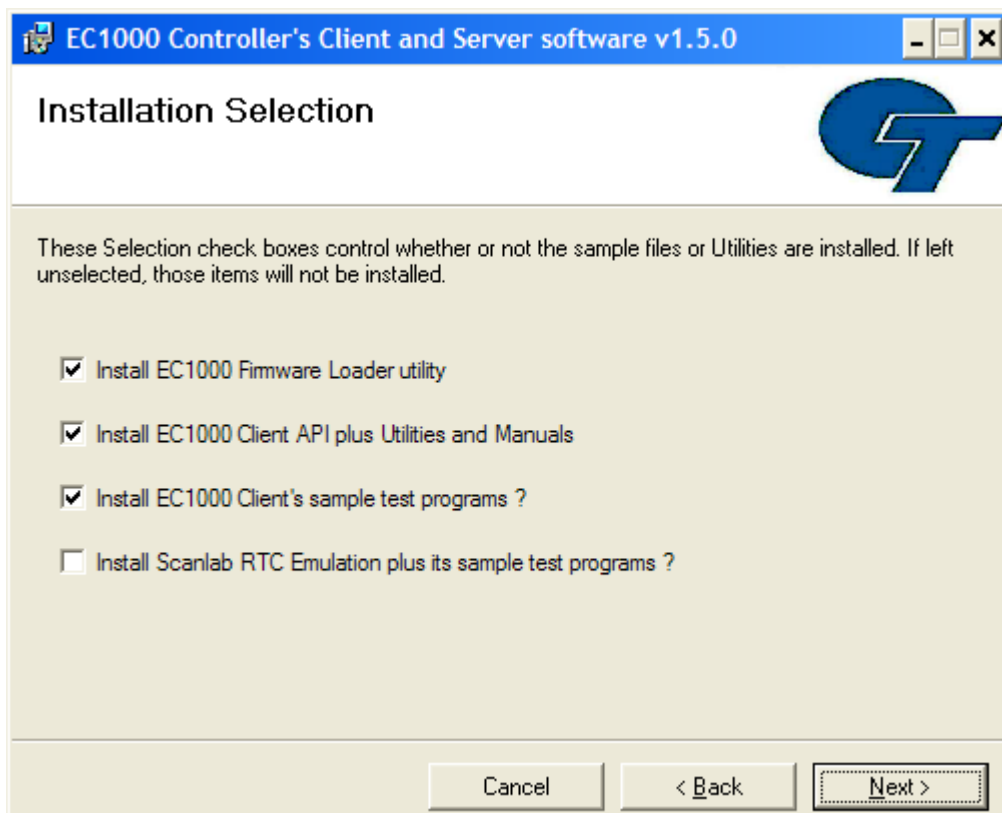
Place the CD into your target computer and run the Setup.exe program in the top level directory of the CD. Follow the steps as directed by the installation script. The following series of screens will appear:



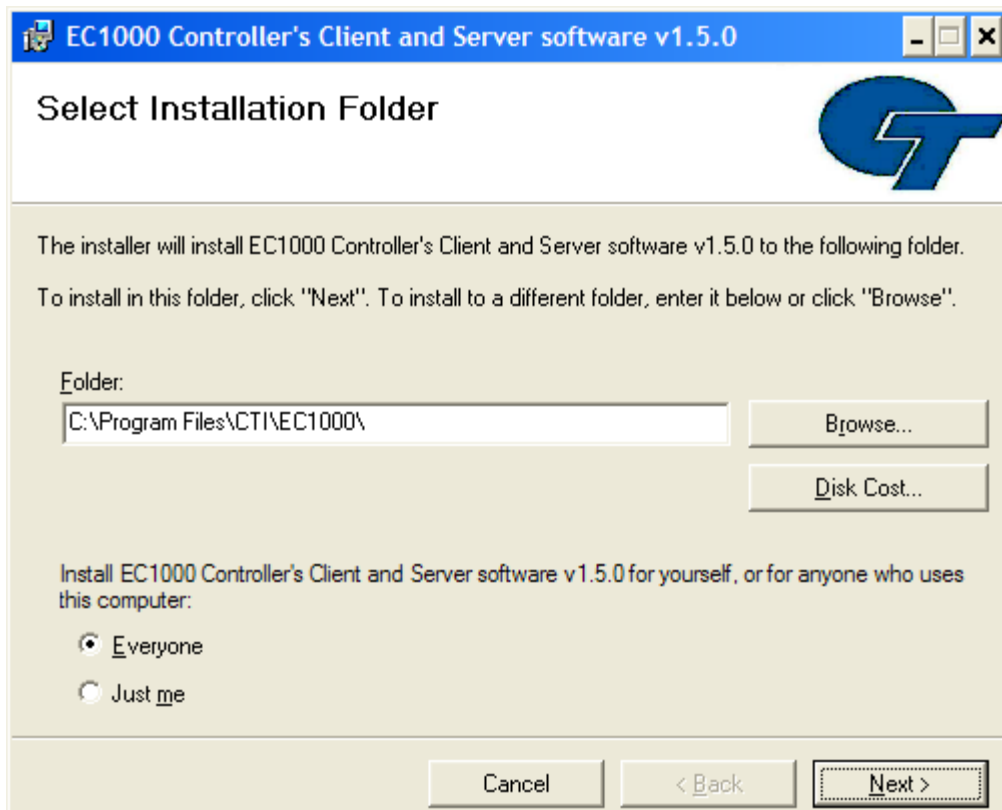
1. Press "Next>" to continue.



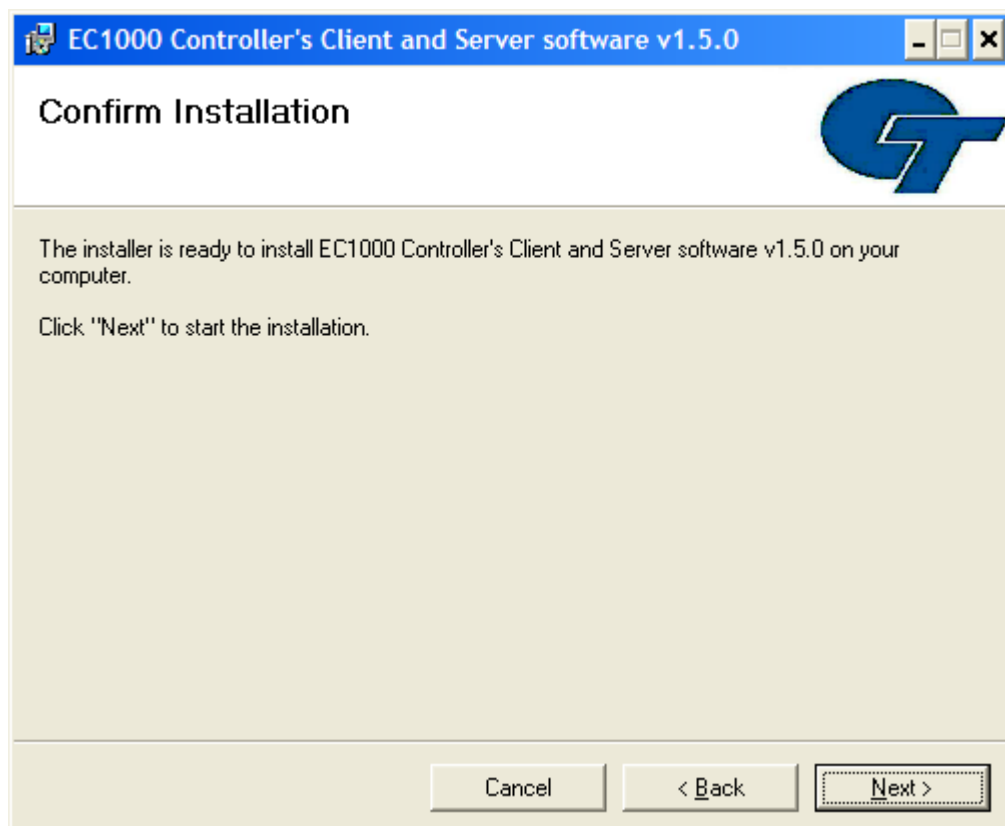
2. Upon reading the license agreement, click the “I agree” button and then press “Next>” to continue.



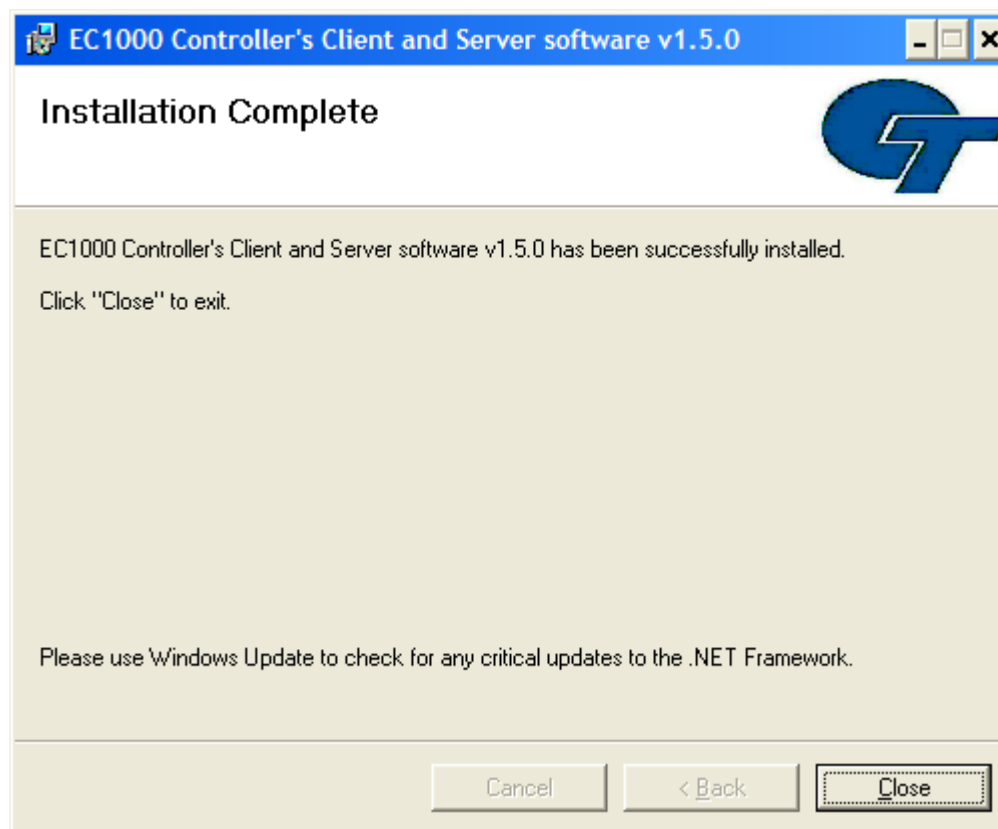
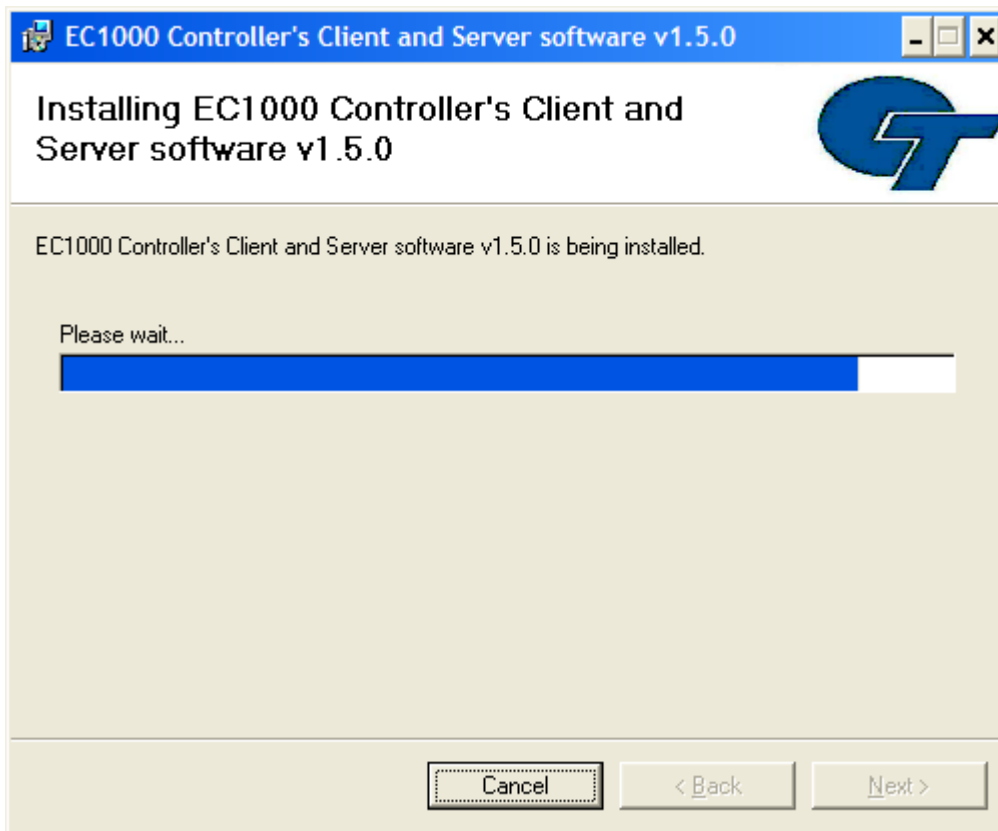
3. Select the items you want to install. Normally the defaults are sufficient but if you also would like to have access to the SCANLAB RTCx emulation library, then chek it too. Press “Next>” to continue.



4. The default installation directory is chosen by the installer. Click "Browse..." to select an alternate location. Also choose if this installation is to apply to everyone or just to yourself. Press "Next>" to continue.



5. Confirm your intentions by pressing Next>. The following sequence of screens will appear:



6. Press "Close" to complete the installation. You are now ready to use the API.

2 Accessing the API and Tools software

The API DLLs are placed in the installation directory, normally C:\Program Files\CTI\EC1000\Client

If application development is done under .NET, then the DLLs can be copied to a convenient project working place. The primary interface DLLs are LECBroadcast.dll and LECSession.dll. The other DLLs are required to support these interfaces. If the COM interface is being used, then the API is already registered with the Windows operating system as ILecBroadcast and ILecSession.

3 Sample Programs

The distribution CD contains sample programs including source code to illustrate the proper use of the API. Installation of these programs is optional.

This CD includes three different test programs plus complete source code that demonstrate how to use the EC1000 Client API to communicate with EC1000 modules. The EC1000 client API can be used as either a .NET assembly or as a COM server and the test programs exercise both of these interfaces. These demonstration programs have been tested under the Microsoft Windows 2000 and XP platforms.

Note: Cambridge Technology Inc. makes no warranties, either explicit or implied, with regards to the applicability of these sample source codes for an end-user application. They are provided merely as examples on how to interface with the EC1000 API.

Warning: These sample programs send sample job commands that move the laser galvanometer mirrors and turn on the laser to draw test images. Use caution when running these programs so that the laser is not aimed at a sensitive target. Always use proper laser safety precautions.

The source code for the examples is located in the installation directory which is normally C:\Program Files\CTI\EC1000\Client\Sample Programs. To access the EC1000 Client API as a COM server via C++ you can follow either the \LecTesterCOM sample or the \LecTesterCOM_ui sample depending if you are using MFC support or not. To access the EC1000 Client API as a .NET Assembly via C# you can follow the \LecTesterNET sample.

All examples show how to reference the Broadcast interfaces, Attach, Detach, query a list of EC1000s on the network, and query System broadcast data and Status broadcast data of each EC1000 found. All examples also show how to reference the Session interfaces, to Login and Logout to any EC1000 found, query for and send Fix Configuration data, send multiple Job data, save Job Data to EC1000 module, and send priority data to Abort jobs.

All queried data is returned in XML format and the samples show how to parse that data.

All samples show how to attach to the Session client to receive Event Notification messages by creating and attaching an Event Sink to an EC1000 Session.

Any application making use of the EC1000 client requires the presence of MS .NET Framework V1.1.4322 in the MS Window system. It is also required the presence of a text file with the same name as your application with ".config" extension at the end of the name i.e. "<MyAppName>.exe.config". The content of the file is the same as the content of the example executables found in each of the \debug directories.

LecTesterNET

This test program interfaces with the .NET mode of the EC1000 client and is written in C# and was compiled with VisualStudio.NET 2003.

What this program does:

- Queries the system for the local Ethernet cards' IP addresses and ask user to select the correct one.
- Creates a Broadcast Listening session by calling clientAttachBroadcast()
- Listens for Broadcast messages for 10 seconds by calling getServerCount() and/or getServerList()
- For each target found it queries for both System & Status info data to get Target's IP address.
- Prompts the user to select a Target IP address to start a Session with.
- The selected target is polled for up to 30 second until it reports Status Available.
- Session is started by calling loginSession() and attaching an EventSink object to receive Events.
- It exercises querying & sending different Config data by calling requestFixedData() & sendFixedData().
- It exercises managing & saving a Job file in target's disk by calling saveJobData().
- It exercises sending multiple Job data blocks by calling sendStreamData() in a loop; the user is prompted for number of

time to loop.

- Prints out all notifications received that report Job execution status.
- Allows user to test Aborting Jobs while they are running by calling `sendPriorityData()`.
- The Job sends & aborts repeat forever until user hits <Enter> key in console window.
- Session is ended by calling `LogoutSession()`; the Disconnect Event is reported.
- Broadcast listening is ended by calling `clientDetachBroadcast()`.

When creating your own .NET application you need to add a reference to the `LecBroadcast.dll` and `LecSession.dll` via VS.NET's Project menu, and select Add Reference. From that dialog click the Browse button to select both these dlls and making them available to your project.

LecTesterCOM

This test program interfaces with the COM mode of the EC1000 client and is written in C++ and was compiled with Visual Studio 6.0.

What this program does:

- Queries the system for the local Ethernet cards' IP addresses and ask user to select the correct one.
- Creates a Broadcast Listening session by calling `clientAttachBroadcast()`.
- Listens for Broadcast messages for 10 seconds by calling `getServerCount()` and/or `getServerList()`.
- For each target found, it queries for both System & Status info data to get Target's IP address.
- Prompts the user to select a Target IP address to start a Session with.
- The selected target is polled for up to 30 second till it reports Status Available. The <Enter> key can interrupt this poll.
- A Session is started by calling `loginSession()` and attaching an Event Sink object to receive Events.
- It exercises querying & sending different configuration data by calling `requestFixedData()` & `sendFixedData()`.
- It exercises sending multiple Job data blocks by calling `sendStreamData()` in a loop.
- Prints out all notifications received that report Job execution status
- Allows user to test Aborting Jobs while they are running by calling `sendPriorityData()`
- A Session is ended by calling `LogoutSession()`, Disconnect Event is reported.
- Broadcast listening is ended by calling `clientDetachBroadcast()`.

When creating your own COM application, you will simply add an `#IMPORT` statement to your code with the directory and name of each of these two table libraries `LecBroadcast.tlb` and `LecSession.tlb`.

LecTesterCOM_ui

This test program interfaces with the COM mode of the EC1000 client and is written in C++ and was compiled with Visual Studio 6.0. The difference between this version and `LecTesterCOM` besides that uses MFC user interface is that it uses a different mechanism to interface to the EC1000 Events received via the EC1000 Client.

What this program does is:

- Queries the system for the local Ethernet cards' IP addresses and ask user to select the correct one.
- Creates a Broadcast Listening session by calling `clientAttachBroadcast()`
- Listens for Broadcast messages for 10 seconds by calling `getServerCount()` and/or `getServerList()`
- For each target found it queries for both System & Status info data to get Target's IP address.
- Prompts the user to select a Target IP address to start a Session with.
- Application MFC dialog pops up with a button to start a Session which user must click to test.
- Before starting a Session user must select an XML Job file via file Browser in dialog.
- The selected target is polled for up to 30 second till it reports Status Available.
- A session is started by calling `loginSession()` and attaching an `EventSink` object to receive Events.
- It exercises saving a Job data file in Target's disk by calling `saveJobData()`.
- It exercises sending one Job data block by calling `sendStreamData()`.
- Prints out all notifications received that report Job execution status
- Allows user to test Aborting Jobs while they are running by calling `sendPriorityData()`
- User can repeat Session by selecting a different XML file and clicking Session button.
- The session is ended by calling `LogoutSession()`, Disconnect Event is reported.
- Hit Close button so Broadcast listening is ended by calling `clientDetachBroadcast()`.

LecTesterUTILS

This directory contains utility functions used by all three sample programs above. It provides utilities for both the COM and .NET programs under separate subdirectories (\cpp and \CSharp). These utilities provide a way for parsing XML data, a function for finding your local Ethernet IP address, a function to interpret Client function errors and other utilities for displaying information.