MCL-Net Version 3
User Documentation

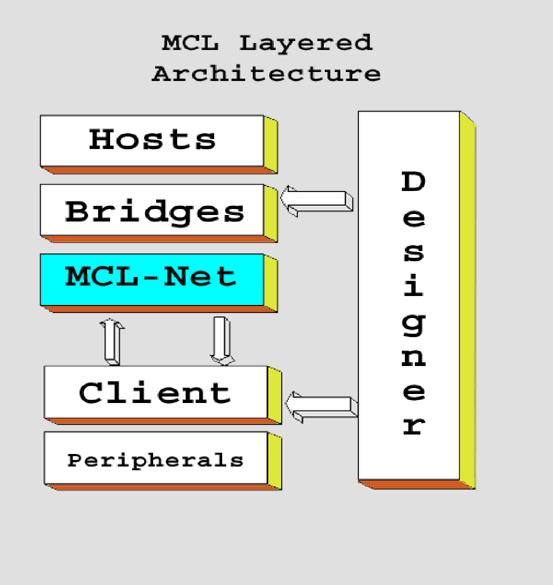
MCL-Net Version 3.07
May 2003

Table of Contents

1 Introduction

MCL-Net is part of the MCL Collection of software. It is a software product that should be installed on a Server somewhere on your Network. It is described as a Communications Server. Its basic function is to support communications between MCL-Client devices and host systems/databases. It is designed to work with MCL-Clients running applications created using MCL-Designer. It can support up to 250 MCL-Client Devices per copy of MCL-Net, however this also depending on one having a suitable licence to support this. MCL-Net controls a variety of vital activities for the MCL-Clients across a network. It provides a flexible communications environment to interface with transaction-driven data collection applications.

Using MCL-Net, the MCL-Client Devices can upload or download data packets, files or programs anywhere as long as they can connect to the Server through a wireless 802.11 or Ethernet 802.3 network. Users can append, copy, rename or delete files on the Server from a remote location. MCL-Net provides time and date synchronization of MCL-Client Devices as well as a variety of interfaces to host applications through a Dynamic Link Library (DLL) on Windows or a shared library MCLNetV3 on UNIX. MCL-Net also includes an ODBC/ SQL engine that runs on Windows platforms that allows the user to access and perform SQL requests on ODBC compliant databases. MCL-Net traces network activity, checks MCL-Client status on the Ethernet/802.11 wireless networks, and manages both data paths and network devices.



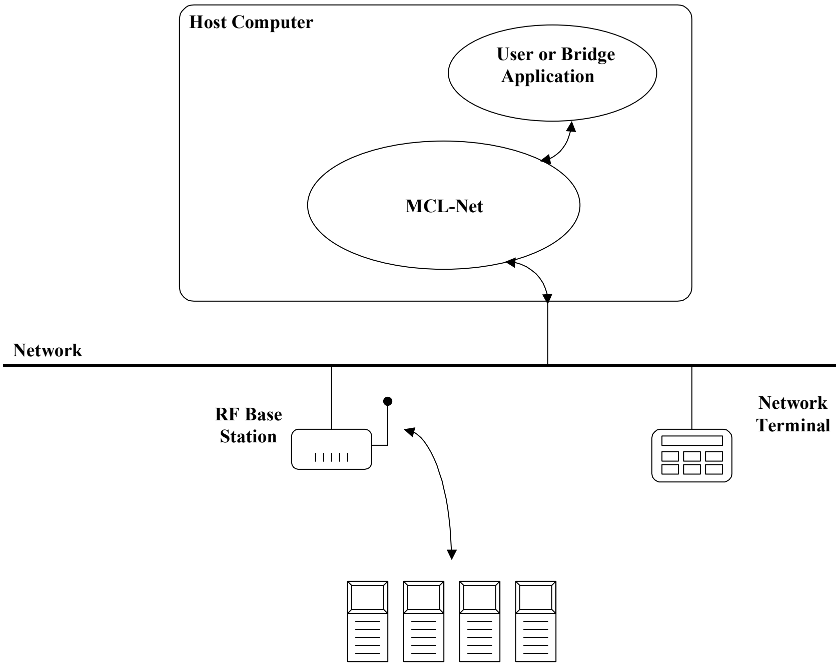
Operational data may be easily saved on networked systems or drives by making declarations in MCL-Net to set up the paths and destinations of data records and files. MCL-Net constructs a MCL-Client ID configuration list that includes Hosts and MCL-Client Devices. Using the master list of all devices connected to the MCL-Net communications Server, enabling or disabling any device on the network becomes quick and easy. Network activity may be traced for devices identified in the MCL-Client ID configuration list and this trace data can be viewed on a screen or written to a file for review

The main functionality of MCL-Net is to:

1. Receive and save data coming from MCL-Clients
2. Manage all "file Server" functionality
3. Transfer files and MCL-Code programs to MCL-Clients
4. Manage auxiliary functions like reset, date and time setting of MCL-Clients.

1.1 Overview

MCL-Net can be seen as a Server, handling messages from client applications written in MCL and running on remote computing devices. It handles basic file operations and will forward specific messages to user applications and Bridge applications using a library of functions.

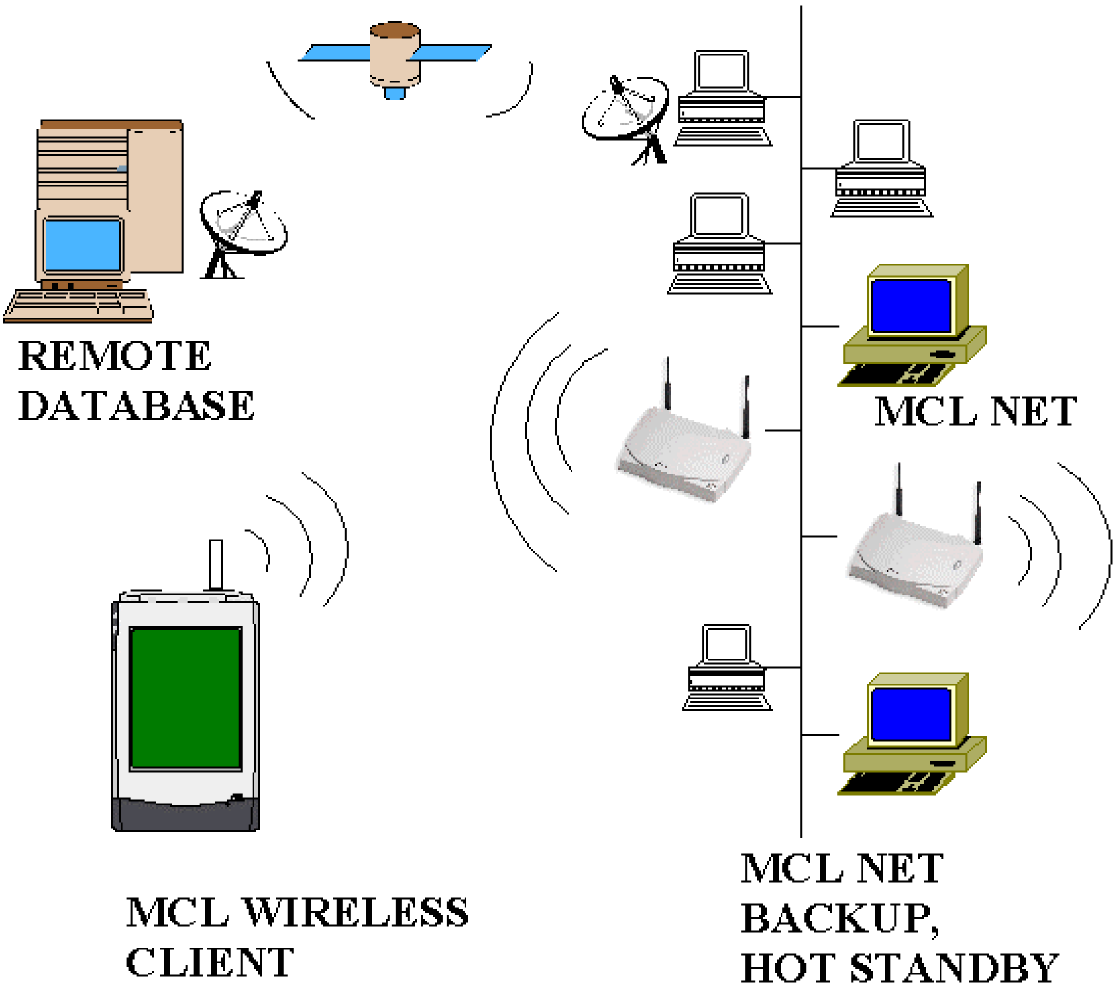


Operations are typically initiated at the MCL-Client; it sends a message to MCL-Net via a base station and/or a TCP/IP Network. On receiving the message, MCL-Net may handle the message internally or forward the message to a User Application or Bridge Application. In fact, all messages except the D0-D9 User messages and Bridge Specific messages are handled internally. D0-D9 messages are specified in MCL-Designer using the “Host Communication” – “Send a Packet” processing. Most messages require a response in some form so MCL-Net stores the Senders IP Address and sends its response to that address. What happens to messages that are not handled internally by MCL-Net is configured using the Dispatcher Tab of the set-up program.

A network diagram below clarifies how MCL-Net might be used in practice

A PC on a network hosts MCL-Net. Wireless devices access this network via Wireless Access Points (Radio Transceiver MAC layer bridges) Packets from the wireless devices are directed at the PC running MCL-Net (There might well be a backup PC running MCL-Net on the network as well) The MCL-Net PC forwards those packets to a database somewhere on the Network, perhaps in a another country over a Wide Area connection. Information from the Database would then return back to the wireless device from the database via the MCL-Net PC.

Possible MCL Net Architecture



2 Installing MCL-Net on Windows

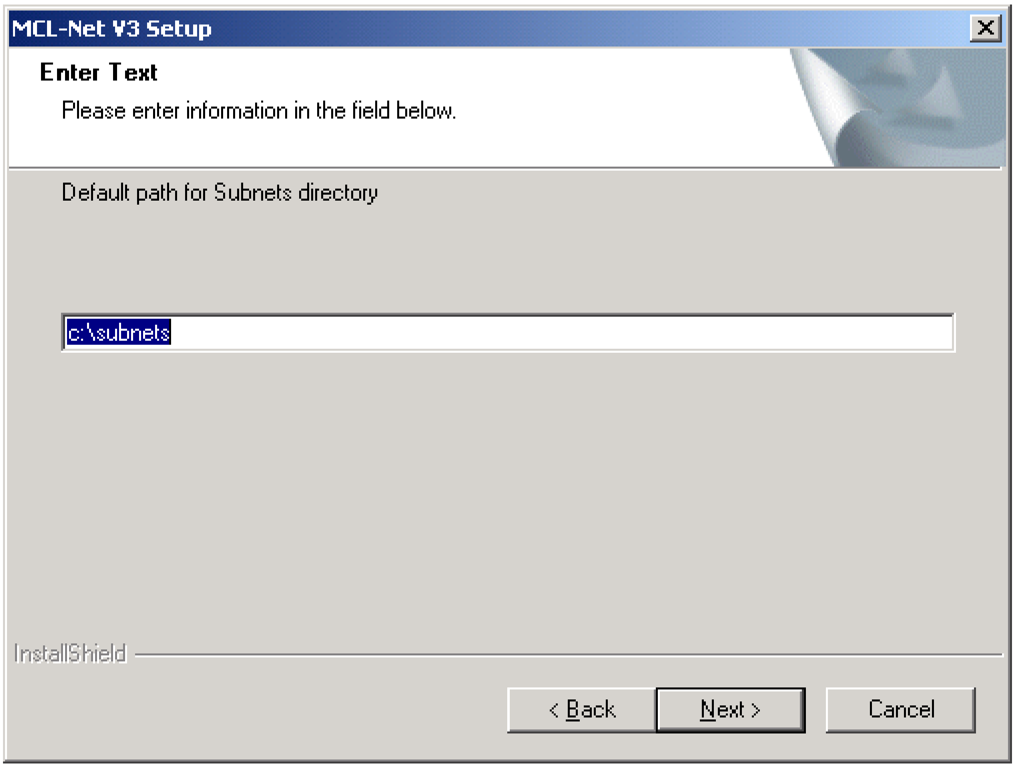
Typically MCL-Net is installed on one PC on the company network. In larger systems, with in excess of 250 MCL-Clients then it may be necessary to invoke a second copy of MCL-Net on the same machine. (A second instance)

This is perfectly possible. In fact; a single computer can run up to 250 instances of MCL-Net. Since each instance of MCL-Net can handle 250 MCL-Clients, this allows a single computer to theoretically handle 62,500 MCL-Clients. For each Instance on MCL-Net, a new Subnet must be created.

Since most installations require less than 250 MCL-Clients, then only a single subnet will be used. If you are installing MCL-Net at the most basic level and want to adopt the most straightforward installation see chapter 10 of this document entitled A Quickstart installation of MCL-Net.

2.1 Installing MCL-Net

Once the setup.exe program for MCL-Net V3 has been extracted from the installation media, simply start this program. During the installation process the operator is asked to select the target directory on the Machine for the MCL-Net Files. A short time later the user is presented with the following screen;



MCL-Net V3 Setup

Enter Text

Please enter information in the field below.

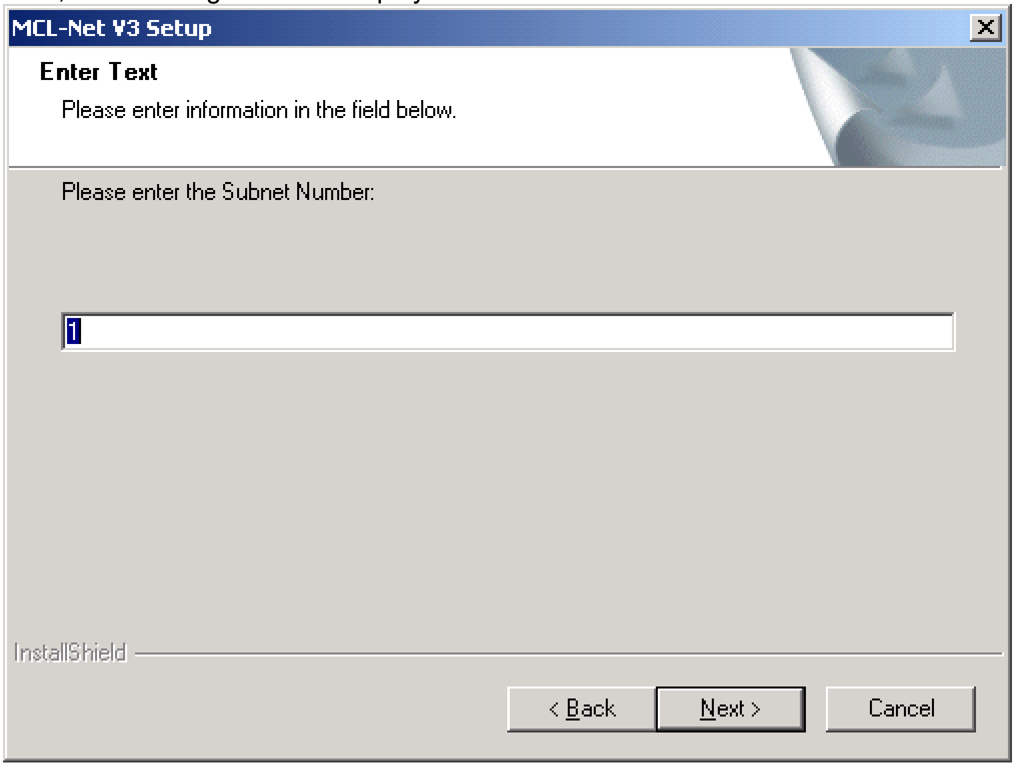
Default path for Subnets directory

c:\subnets

This information is required for the installation process to create the first “Subnet” for the machine. (If you have less than 250 MCL devices then you will only need the one subnet)

Default Path for Subnets Directory; this is the location where the Subnets directories will be created. The first directory is subnet1. However if the user changes the subnet number, a different directory will be created.

Next, the following screen is displayed



Please enter information in the field below.

Please enter the Subnet Number:

1

Subnet Number; This number affects the name of the subnet directory created and the UDP port that MCL-Net will use. This number when added to 5000 gives the actual port that MCL-Net uses. 5000 is the default base port and should only be changed when the user has no other choice. Changing the base port is done after installation and must be done on both MCL-Net and all MCL-Client Devices connecting to MCL-Net.

Additional subnets can be created after installation using the subnet creation wizard

2.2 Licensing of MCL-Net (May 2003)

Having installed MCL-Net you will be asked to run the product in Demo mode (Runs for 1 hour before automatically shutting down) or to enter license details (Activation, see note 1 below*)

When purchasing your license, you need to decide how many simultaneously connected MCL- Clients you expect to use. Licenses are available for the following quantities of simultaneous users: 5, 25,50,100,250. In addition you can buy add on licenses such that you may increase the number of MCL-Clients by 5, 25 or 250 MCL Devices at a time. (NB. The denominations for add on licenses may vary)

If you intend to use more than 250 devices then you need to configure MCL-Net to run multiple instances of itself (Subnets). You may have up to 250 subnets per PC. A single PC can therefore handle thousands of MCL-Clients.

In most systems (Typical systems less than 100 MCL Clients) then there is very little to be gained by using multiple subnets on the same PC (multiple instances of MCL-Net). It is sometimes suggested that doing this will increase throughput, because MCL-Net might be the bottleneck in the system. Owing to the multithreaded nature of MCL-Net, this is very unlikely to be the case. However, if you believe this might be true in your particular system, please contact MCL International Support to obtain specific advice. Even under these circumstances it may be more prudent to run MCL-Net on a second PC for resilience purposes and divide your MCL-Clients accordingly.

IMPORTANT NOTE FOR LARGER SYSTEMS WITH MULTIPLE SUBNETS:

Your original MCL-Net license determines the number of MCL Clients that can be simultaneous attached to the first subnet plus any “additional user” quantities that you have installed up to 250 Devices. For the second and further subnets **on this same PC**, you should purchase “250 additional users” licenses. However, you are not allowed to exceed the number of licensed users on your first subnet on any additional subsequent subnets. In other words; if your first subnet is licensed for 5 users, then the second and subsequent subnets can also only be accessed by up to 5 units even though you will have purchased an additional 250-user license (Note 2 below *) Therefore to take full advantage of purchasing the “250 additional users” licenses, then your first subnet should also be fully licensed for up to 250 users

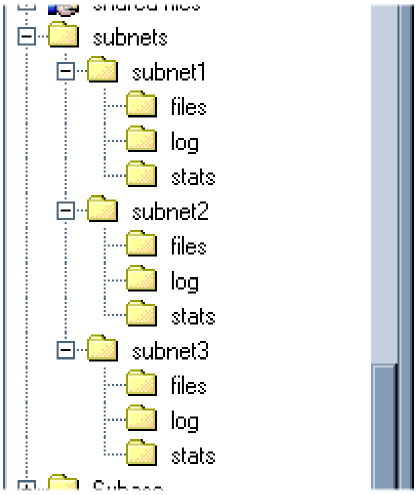
** Note 1: When activating the product, you make a choice to dedicate the license to that particular PC* ***or*** *to use the license in conjunction with an optional “Hardware Key” which gives PC independence/mobility to your license. For specific information on use of the “Hardware Key”, see your license certificate*

**Note 2: When you purchase an “add on” 250 user licenses, you are effectively purchasing another subnet with up to 250 users, but see paragraph above for important restriction)*

2.3 Starting Several MCL-Nets on one machine.

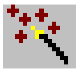
(Only applicable for those installing systems with more than 250 MCL-Clients)

If you need to start several MCL-Nets on one machine and your license allows multiple subnets, then a separate subnet directory needs to be set up for each instance, each containing it's own ini file. In the following example, we're going to have three subnets on the machine, 1, 2 and 3. The recommended directory structure would look something like this.



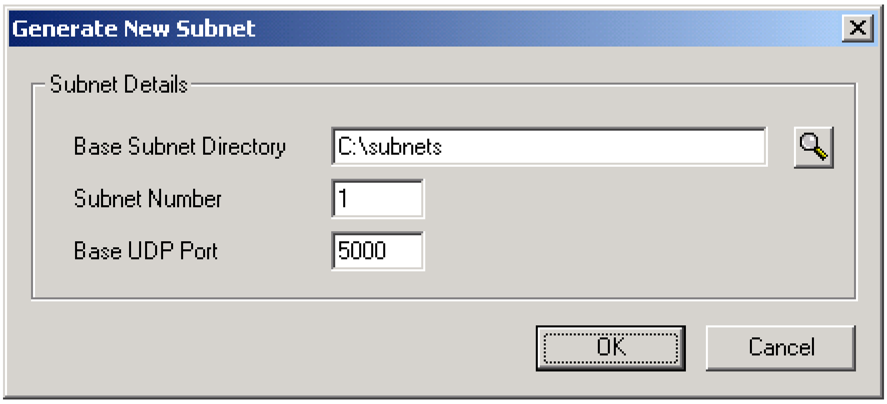
```
subnets
    subnet1
        files
        log
        stats
    subnet2
        files
        log
        stats
    subnet3
        files
        log
        stats
```

The way to create the three MCL-Nets is to use the subnet creation wizard on the MCL-Net setup menu.



This is the subnet creation wizard icon.

The subnet creation wizard will give the following screen,



Generate New Subnet

Subnet Details

Base Subnet Directory: C:\subnets

Subnet Number: 1

Base UDP Port: 5000

On this screen, simply enter the directory where you want to create the subnet, the subnet number and base port. For our example, the only field we change is the subnet number and in here we put 1 and press ok, 2 and press ok, 3 and press ok. This will create three subnets in the following directories

C:\subnets\subnet1
C:\subnets\subnet2

C:\subnets\subnet3

This will create the directories, ini files and shortcuts that you need for each instance of MCL-Net.

When you then double-click on the shortcuts, MCL-Net will be opened using the relevant ini file for that particular instance of MCL-Net.

2.4 Installing MCL-Net as a Windows NT / Windows 2000 Service

MCL-Net can be installed as a Windows NT service.

For this instance of the program, check that you are happy with the settings defined in the MCL-Net Setup drop down menu. Then

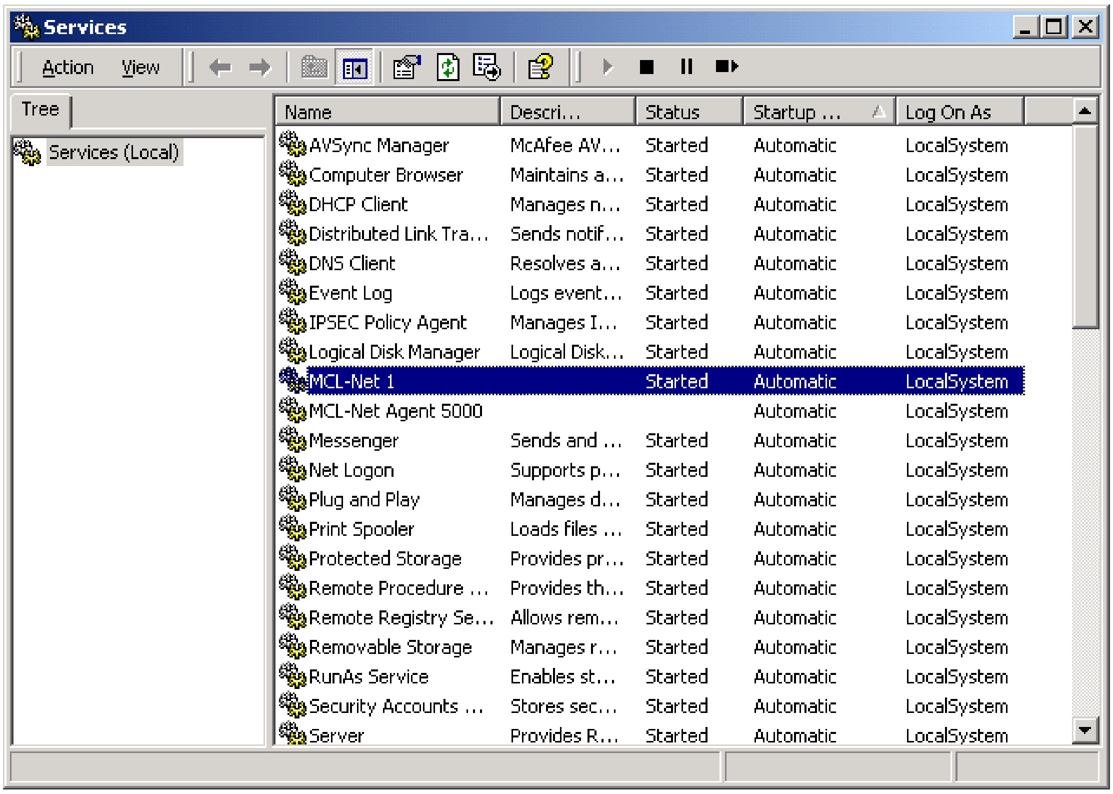
There are then 2 methods to install as a Service;

1. The user can edit the MCLNet.ini file for that subnet and create a Service=TRUE setting in the [Global] section. Then save the file and start the MCL-Net program using the shortcut in the subnet directory. This will install the service if it doesn't exist and start the service.

2. The user can start the MCL-Net program in a MS-DOS window with the arguments –install –i <inifile> to add the MCL-Net to the services list.

```
MCLNet -install -i c:\subnets\subnet1\MCLNet.ini
```

The Subnet entry in the ini file is used to give each Service instance of MCL-Net it's own unique name. The name given to the Service entry will be "MCL-Net" followed by the Subnet number, for example "MCL-Net 3". Each Service entry may use the same MCL-Net executable, but must be installed with different ini files, each having a different Subnet number. The MCL-Net Service appears in the list of Services available on system (Control Panel – Services for NT) or (Control Panel - Administrative Tools – Services for Windows 2000).



Name	Descri...	Status	Startup ...	Log On As
AVSync Manager	McAfee AV...	Started	Automatic	LocalSystem
Computer Browser	Maintains a...	Started	Automatic	LocalSystem
DHCP Client	Manages n...	Started	Automatic	LocalSystem
Distributed Link Tra...	Sends notif...	Started	Automatic	LocalSystem
DNS Client	Resolves a...	Started	Automatic	LocalSystem
Event Log	Logs event...	Started	Automatic	LocalSystem
IPSEC Policy Agent	Manages I...	Started	Automatic	LocalSystem
Logical Disk Manager	Logical Disk...	Started	Automatic	LocalSystem
MCL-Net 1		Started	Automatic	LocalSystem
MCL-Net Agent 5000			Automatic	LocalSystem
Messenger	Sends and ...	Started	Automatic	LocalSystem
Net Logon	Supports p...	Started	Automatic	LocalSystem
Plug and Play	Manages d...	Started	Automatic	LocalSystem
Print Spooler	Loads files ...	Started	Automatic	LocalSystem
Protected Storage	Provides pr...	Started	Automatic	LocalSystem
Remote Procedure ...	Provides th...	Started	Automatic	LocalSystem
Remote Registry Se...	Allows rem...	Started	Automatic	LocalSystem
Removable Storage	Manages r...	Started	Automatic	LocalSystem
RunAs Service	Enables st...	Started	Automatic	LocalSystem
Security Accounts ...	Stores sec...	Started	Automatic	LocalSystem
Server	Provides R...	Started	Automatic	LocalSystem

The user can modify the startup mode of the service (manual or automatic), start and stop the service, or use the Windows NT standard command:

```
net start "MCL-Net 3"
```

Which will start the MCL-Net 3 service.

Because the MCLNet Setup is a remote client application, it can connect to a MCL-Net Service and administer it. This allows the settings of an instance of MCL-Net running as a Service to be administered and then bounced to allow the changes to take affect.

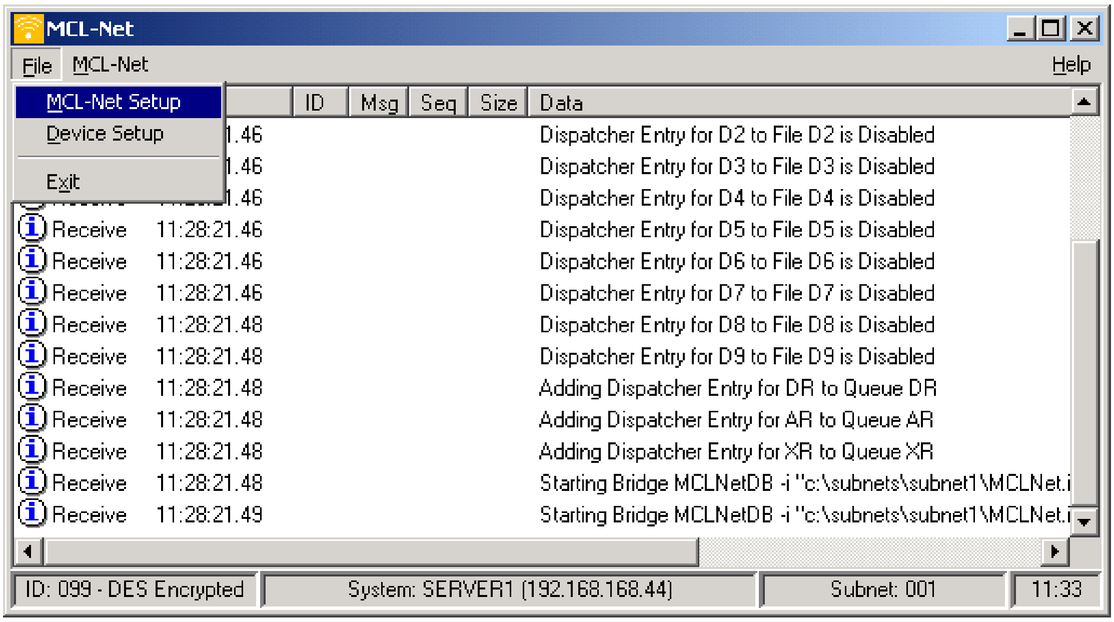
You can also run the MCL-Net Viewer against a running Service, by using the command

```
      MCLNet –I c:\subnets\subnet1\MCLNet.ini
```

OR simply double-clicking the shortcut already created by the install process in the appropriate subnet.

3 MCL-Net Setup Introduction

During the installation we encountered the FILE drop down menu which included in its list the MCL-Net Setup option. Selecting this option actually runs a program called MCLNetSetup



```
              1.46           Dispatcher Entry for D2 to File D2 is Disabled
              1.46           Dispatcher Entry for D3 to File D3 is Disabled
              1.46           Dispatcher Entry for D4 to File D4 is Disabled
Receive    11:28:21.46       Dispatcher Entry for D5 to File D5 is Disabled
Receive    11:28:21.46       Dispatcher Entry for D6 to File D6 is Disabled
Receive    11:28:21.46       Dispatcher Entry for D7 to File D7 is Disabled
Receive    11:28:21.48       Dispatcher Entry for D8 to File D8 is Disabled
Receive    11:28:21.48       Dispatcher Entry for D9 to File D9 is Disabled
Receive    11:28:21.48       Adding Dispatcher Entry for DR to Queue DR
Receive    11:28:21.48       Adding Dispatcher Entry for AR to Queue AR
Receive    11:28:21.48       Adding Dispatcher Entry for XR to Queue XR
Receive    11:28:21.48       Starting Bridge MCLNetDB -i "c:\subnets\subnet1\MCLNet.i
Receive    11:28:21.49       Starting Bridge MCLNetDB -i "c:\subnets\subnet1\MCLNet.i
```

ID: 099 - DES Encrypted | System: SERVER1 (192.168.168.44) | Subnet: 001 | 11:33

The main functionality of the MCLNetSetup program is to connect to an instance of MCL-Net and change it’s configuration.

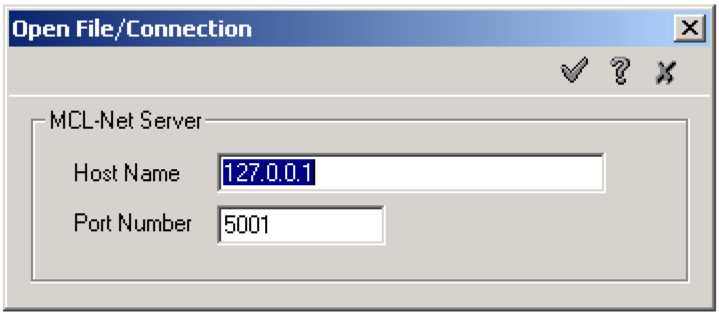
3.1 Starting MCLNetSetup

There are a number of methods that can be used to start MCLNetSetup. One is from the menu of each MCL-Net application. However it can also be started from the Tree View in the MCL-Net Manager. Or it can be run in stand-alone mode from within the C:/MCLNET directory. (Default Path)

Depending on which MCL-Net Instance it connects to, MCLNetSetup will display one or more of the following tabs in a tabbed dialog window.

Global
Dispatcher
Bridges
Logging.

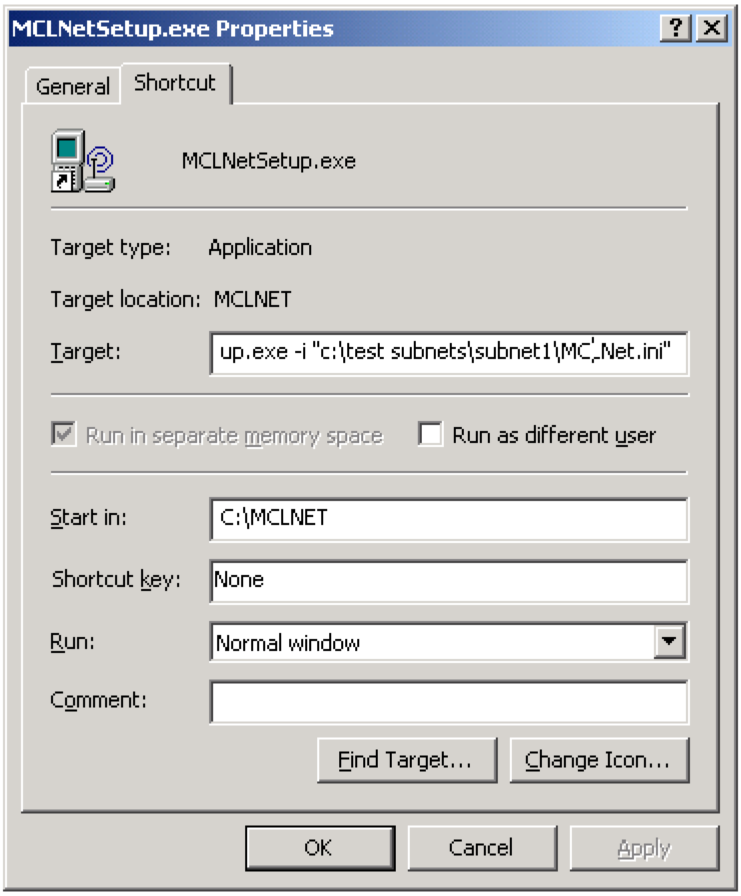
MCLNetSetup when started as a stand-alone application will search the current directory for the MCLNet.ini file. If it exists, it will use the connection parameters from the file to attempt to connect to MCL-Net. If the connection fails, or there’s no INI file the program prompts the user to enter the connection parameters for the particular instance of MCL-Net that you wish to reconfigure.



Host Name: 127.0.0.1

Port Number: 5001

The user can also force the MCLNetSetup program to use a specific MCLNet.ini by creating a short cut to the exe. In the properties of the shortcut specify the INI file as part of the target line;



Target type: Application

Target location: MCLNET

Target: up.exe -i "c:\test subnets\subnet1\MCLNet.ini"

☑ Run in separate memory space ☐ Run as different user

Start in: C:\MCLNET

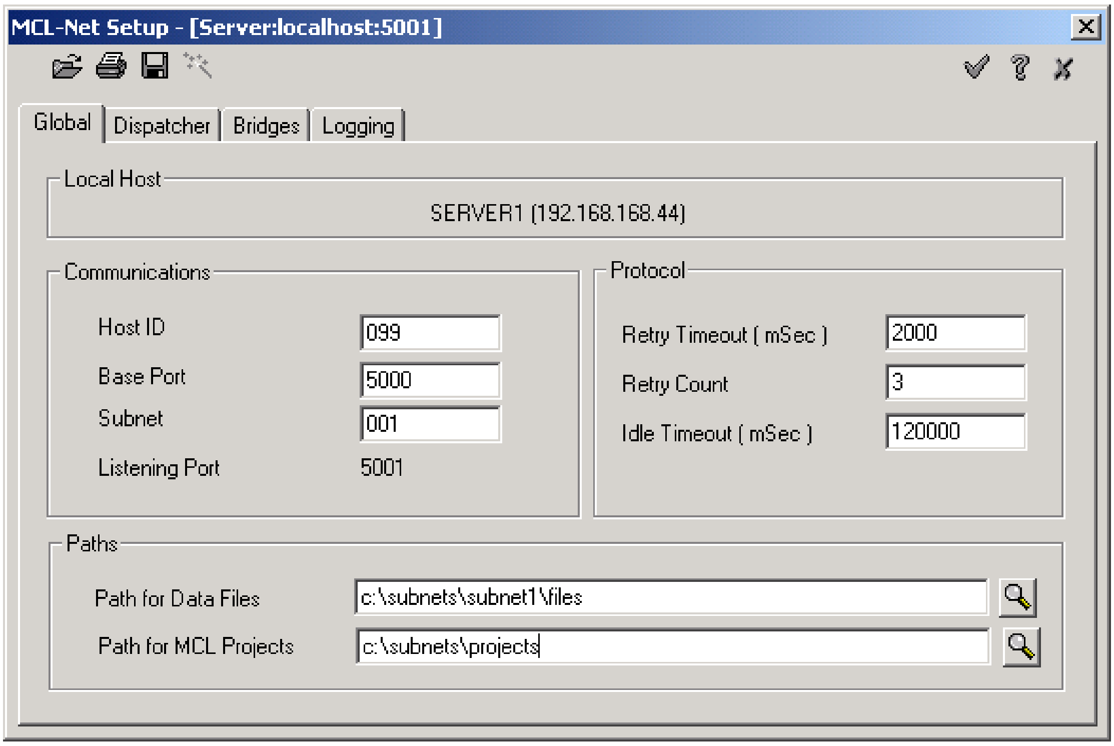
Shortcut key: None

Run: Normal window

Comment:

3.2 Global Set-up

The Global tab accesses information about Communication, Current Host, Protocol and the File Paths.



MCL-Net Setup - [Server:localhost:5001]

Local Host

SERVER1 (192.168.168.44)

Communications

Host ID 099

Base Port 5000

Subnet 001

Listening Port 5001

Protocol

Retry Timeout (mSec) 2000

Retry Count 3

Idle Timeout (mSec) 120000

Paths

Path for Data Files c:\subnets\subnet1\files

Path for MCL Projects c:\subnets\projects

3.2.1 Local Host:

The current host that you are connected to.

3.2.2 Communication:

Host ID (1-254)	The Host ID of MCL-Net Server.
Base Port (default 5000)	The base port number of MCL-Net. The actual port that MCL-Net listens on is the Base Port added to the Subnet number.
Subnet (1-255)	The Subnet of MCL-Net. There may be several instances of MCL-Net on one machine, each having a unique Subnet number. The actual port that MCL-Net listens on is the Base Port added to the Subnet number.
Port (display only)	The listening port number of MCL-Net

3.2.3 Protocol:

Retry Interval

After MCL-Net has failed to post a message to a Queue, if Retries > 0, it will wait this amount of Milliseconds before the retry.

Retry Timeout

After MCL-Net has sent a message to a MCL-Client. It will wait this amount of Milliseconds for a response before checking if a retry is required.

Retry Count

The number of times MCL-Net will retry posting a message.

Idle Timeout

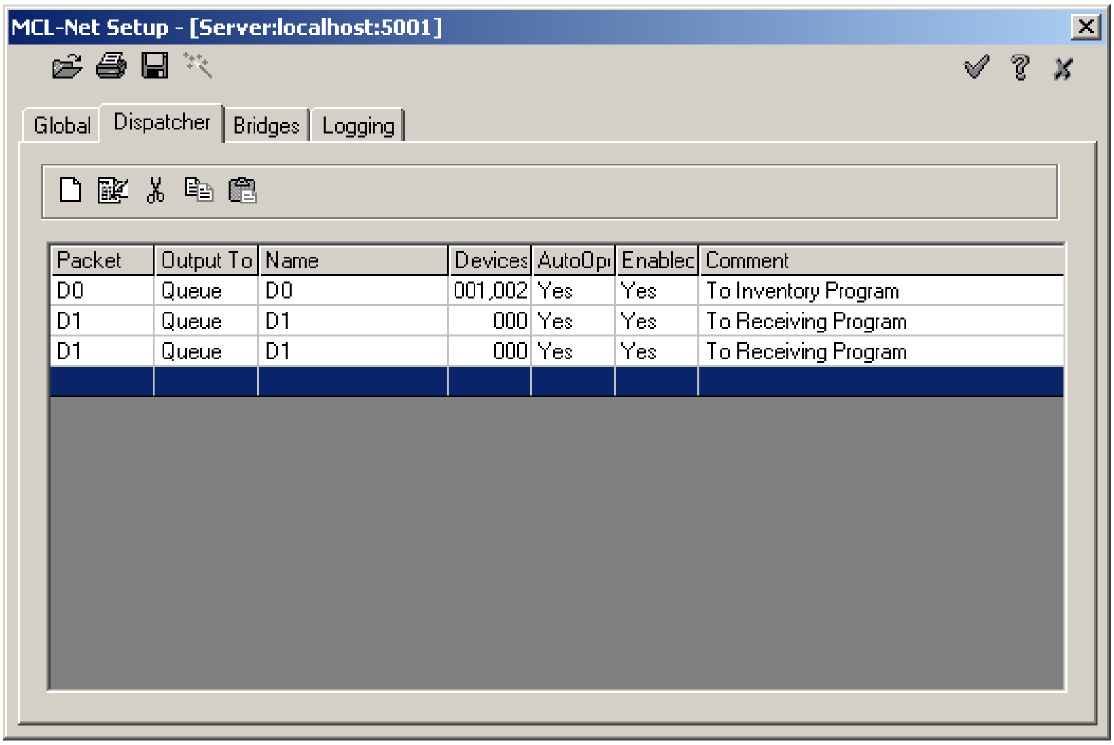
This is the amount of time in Milliseconds a Send Thread must be idle before it will terminate and free any resources it's using.

3.2.4 Paths:

The default Path for Project files defines the path of the directory where Project files are stored. This path is necessary to find the correct SQL script file (PROJECT.MQD) related to a Project. The default Path for Data Files is the path of the directory where data files are stored.

3.3 Dispatcher Set-up

The Dispatcher tab sets up the data packet dispatcher to route the incoming packets to the corresponding application or file. The user can decide to dispatch selected incoming data (following packet type and MCL-Client ID) to a file or to a queue that is attached to an application.
The data dispatcher is a module that is used to define what happens when MCL-Net receives specific user definable messages. The dispatcher enables incoming data packets to be “routed” to the right host application or file, depending on MCL-Client ID (sender) and Packet Type. For instance, all type “D0” packets from MCL-Client number 001 and 002 can be routed to an inventory program (via a queue) while all type “D1” packets from all MCL-Clients are routed to the receiving goods program (via another queue).



Packet	Output To	Name	Devices	AutoOp	Enablec	Comment
D0	Queue	D0	001,002	Yes	Yes	To Inventory Program
D1	Queue	D1	000	Yes	Yes	To Receiving Program
D1	Queue	D1	000	Yes	Yes	To Receiving Program

The Data Dispatcher allows you to perform a “Load Balance” by giving the possibility to start multiple applications for the same queue for the same criteria (MCL-Client ID and Packet Type). This parallel implementation gives better response when several MCL-Clients perform the same transaction at the same time. The Data Dispatcher will then route each incoming packet to the relevant queue. Any number of applications can be watching the same queue (up to operating system limitations).

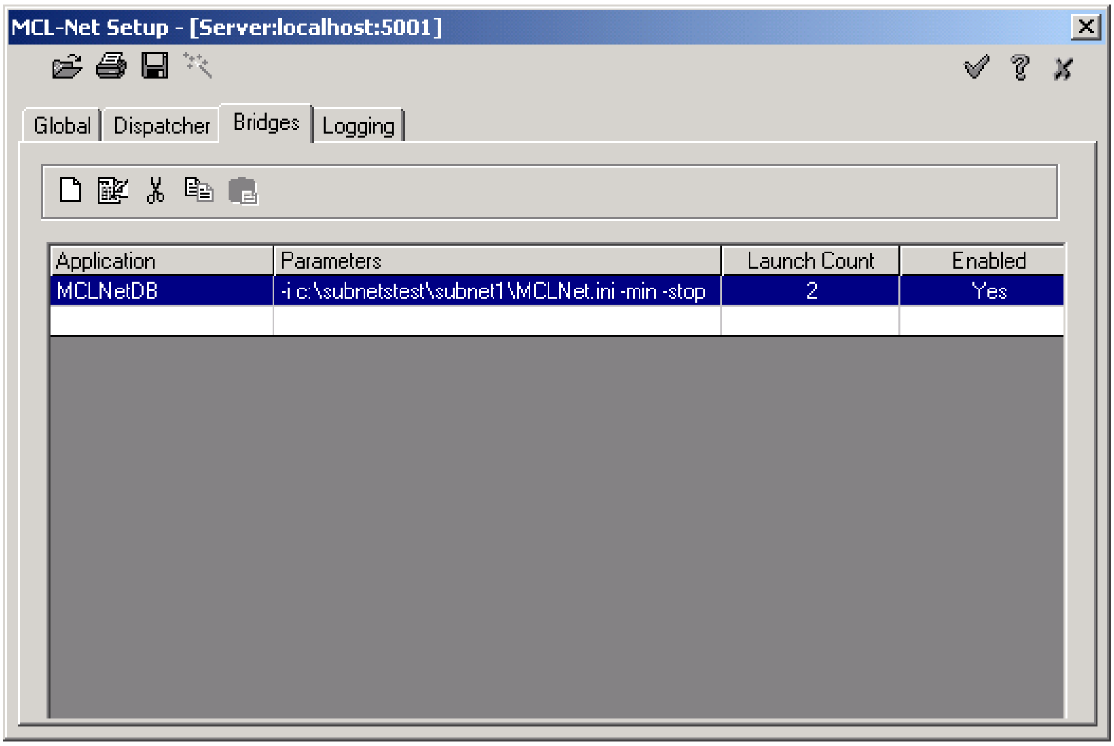
The Data Dispatcher allows you to route incoming packets to text type data files. Each incoming packet will then be appended into the corresponding file, the host application can then, at any time, read the content of this file or rename the file for further operation without file sharing problems. The Dispatcher will create the file if it doesn’t exist. This technique is usually applied when the MCL-Clients are only sending data to the host without any information coming back.

The AutoStart function: When this is set to YES, this indicates that MCL-Net will automatically create the specified queue or file at startup. If it set to no, the application program that requires the use of the specified data must create this queue.

To add or edit an entry in the dispatcher section, double click on the corresponding entry

3.4 Bridges Set-up

The Bridges tab lists the applications that will be run by MCL-Net on startup. In its default mode MCL-Net will run two copies of the MCLNetDB program. (See ODBC section for details) To successfully launch an application, the application name, the parameters, the Launch count and Enabled must be set. The Launch count specifies how many instances of the application you wish to run. Enabled must be set to yes, if not it will be ignored.



Application	Parameters	Launch Count	Enabled
MCLNetDB	-i c:\subnetstest\subnet1\MCLNet.ini -min -stop	2	Yes

To enter the above fields double click on the grid.

3.4.1 ODBC Setup

To access an SQL database through ODBC, the MCL-Client sends a specific request (In fact a “DR” data packet) that contains a filename, the SQL query number and the data to MCL-Net. In other words the MCL-Client does not send a full SQL statement, instead it sends information to MCL-Net telling it where to look for the predefined SQL statement that the MCL-Client wishes to execute. All the SQL statements for the MCL application are stored in a single file known as the MQD file that is created by the MCL-Designer.

MCL-Net transfers this request to the MCL-Net OBDC/SQL engine that extracts the correct SQL query from the MQD file. (The MQD file is stored in the specified Project directory. This directory is defined in the global setup)

When the MCL-Net ODBC engine finds (or cannot find) the corresponding data in the database, it sends return codes and data to the MCL-Client back through MCL-Net. The return code indicates the number of records that has found in the database.

The MCL-Net OBDC/SQL engine is an executable file named MCLNetDB.EXE and is located in the working directory of MCL-Net. Its syntax is shown below:

MCLNetdb queuename [-argument]

Where:

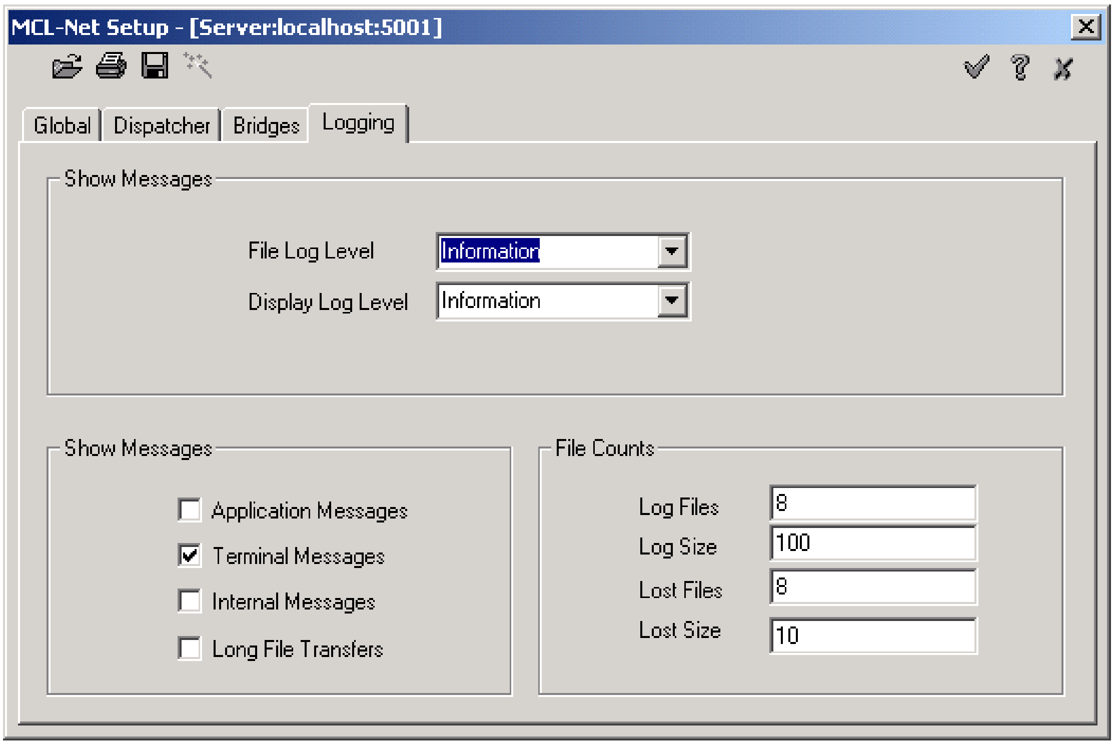
queuename: The name of the pipe that can be open automatically (Auto-Open set to Yes) by MCL-Net dispatcher or by the user application. The Default queuename is DR and the default packet type is DR.

[-argument]:	none:	MCLNetDB starts with normal window.
-min:		MCLNetDB starts minimized
-max:		MCLNetDB starts maximized

You can launch MCLNetDB from the bridges tab as shown in the screen shot above; two copies of this are launched by default (Two for added throughput capabilities).

3.5 Logging Set-up

The Logging Setup allows you to define the level of detail recorded in the transaction logging files of MCL-Net.



Show Messages

File Log Level: Information

Display Log Level: Information

Show Messages

☐ Application Messages

☑ Terminal Messages

☐ Internal Messages

☐ Long File Transfers

File Counts

Log Files: 8

Log Size: 100

Lost Files: 8

Lost Size: 10

A different level may be set for the messages that go to the Viewer window (Display Log Level) and to the logging files themselves. There are three levels for each of these options, being ‘Error’, ‘Warning’, and ‘Information’. ‘Error’ will only show Error messages, ‘Warning’ will show both ‘Error’ and ‘Warning’ messages, and ‘Information’ will show all three.

There are also three groups of messages that may be switched on and off, being Application Messages (TCP), MCL-Client Messages (UDP) and Internal Messages (Queue messages). UDP messages are messages that flow between MCL-Net and the MCL-Clients, TCP messages are all other network messages, and Queue messages are messages that flow internally on MCL-Net between its various modules. For example if you are only interested in seeing messages between your 3rd party application and MCL-Net, then enable TCP messages and disable UDP and Queue messages. To show the full path of messages as they pass around the system, enable all three.

One other option in the show messages section is Long File Transfers. If this is switched off all messages for file transfers to and from the Server will be shown. If this is switched on only the first and last file messages of the transfer will be shown. This applies to the Viewer window only. The log files will always contain all the file messages regardless of whether the Long File Transfers is on or off.

The File Counts section: This allows the setting of how many log files are generated, and how large they are. The log files have a limited size, once that size is reached, it is copied to a backup file and a new log file is started. Once the maximum number of log files is reached, the oldest log file is overwritten. This means that you can easily limit the amount of disk space that is used by the log files.

The Lost Files; hold packets that are lost in the system, whether due to not receiving an acknowledgement, or other similar situations and these also overwrite themselves in due course as per the logging files.

4 Revision Administration Introduction

The main functionality of the Revision Administration program is to allow the centralized and automatic deployment of MCL-Client applications (Projects). It also allows you to maintain MCL-Client specific information; this ensures the relevant Project is loaded with a minimum amount effort.

The Revision Administration program is used to connect to an instance of MCL-Net and change the Terminals.ini file. The Terminals.ini file stores information about MCL-Clients served by that instance of MCL-Net and should be located in the same directory as MCLNet.ini.

When initially powering up a MCL-Client the usual set-up is required such as IP settings etc such that it can communicate with MCL-Net. Once the Terminals.ini file is set up within MCL-Net, and MCL-Client that connects to MCL-Net can auto load its default Project.

In the Global section of the Terminals.ini file a TIDControl entry is set which should have one of the following options:

VeryLow
Low
Medium
High
VeryHigh.

Very Low is the default and doesn't strictly need an entry in the Terminals.ini file.

Low allows the MCL-Client to be loaded with a minimum of setup but also has a corresponding minimal level of control. The MCL-Client ID (TID), MCL-Client model (TMDL) and MCL-Client serial number (TSN) are not set on the Server and the TID is not set on the MCL-Client. The Server allocates a free TID, adds an entry in Terminals.ini and sends a list of groups to the MCL-Client for that MCL-Client type. The user can then select the group, and from the selected group can select a Project.

Medium the MCL-Client must exist in Terminals.ini, the group if defined is returned. If the group is not defined, groups set up for that MCL-Client type are returned.

High the TMDL and TSN must exist in Terminals.ini. If these exist a spare TID is allocated, if not it is rejected.

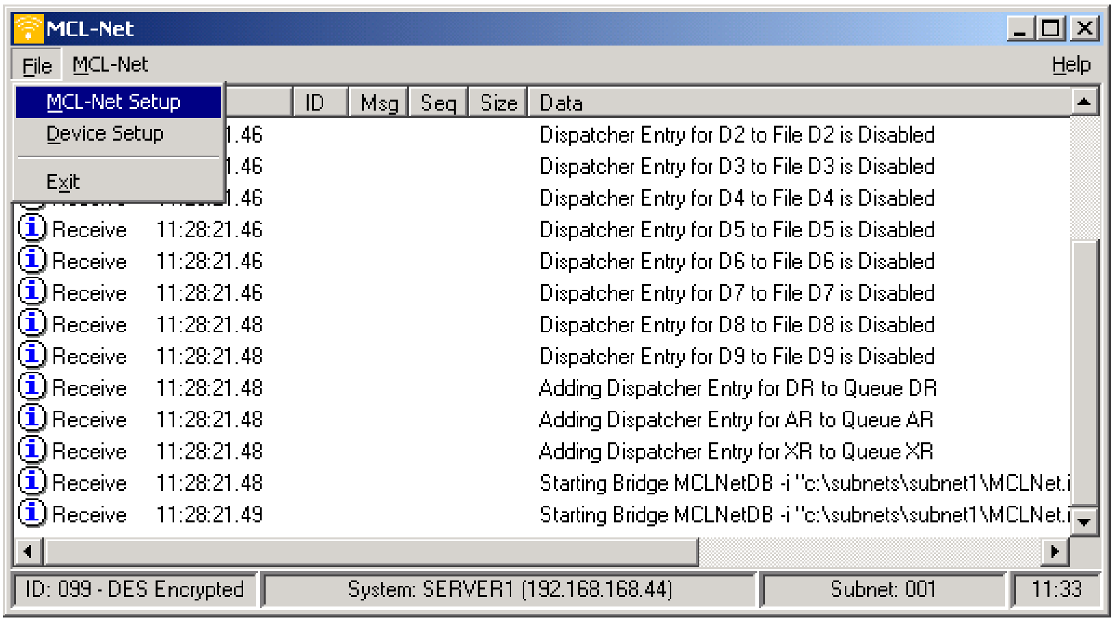
Very High gives the tightest control and the TID, TMDL and TSN must exist in Terminals.ini.

Also in the Global section there is a UseMCL-ClientIP, which can be TRUE or FALSE, if this is true when the MCL-Client connects for the first time its IP is inserted into the ini file else the IP is set to 0.0.0.0.

Entries in the Global section are not edited through the Revision Administration section as they rarely change.

4.1 Starting Revision Administration

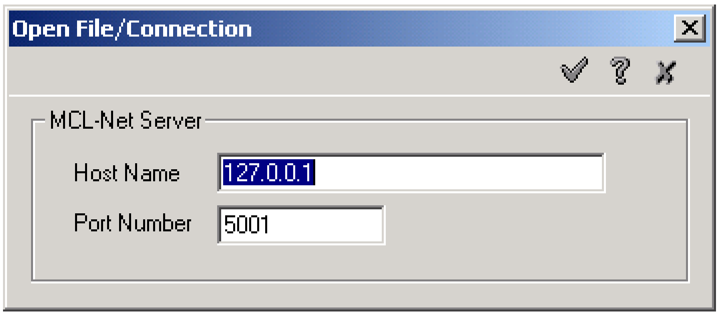
Revision Administration is a stand-alone application, although it can be launched by selecting Device setup from the dropdown menus from within a running instance of MCL-Net.



		ID	Msg	Seq	Size	Data
	1.46					Dispatcher Entry for D2 to File D2 is Disabled
	1.46					Dispatcher Entry for D3 to File D3 is Disabled
	1.46					Dispatcher Entry for D4 to File D4 is Disabled
Receive	11:28:21.46					Dispatcher Entry for D5 to File D5 is Disabled
Receive	11:28:21.46					Dispatcher Entry for D6 to File D6 is Disabled
Receive	11:28:21.46					Dispatcher Entry for D7 to File D7 is Disabled
Receive	11:28:21.48					Dispatcher Entry for D8 to File D8 is Disabled
Receive	11:28:21.48					Dispatcher Entry for D9 to File D9 is Disabled
Receive	11:28:21.48					Adding Dispatcher Entry for DR to Queue DR
Receive	11:28:21.48					Adding Dispatcher Entry for AR to Queue AR
Receive	11:28:21.48					Adding Dispatcher Entry for XR to Queue XR
Receive	11:28:21.48					Starting Bridge MCLNetDB -i "c:\subnets\subnet1\MCLNet.i
Receive	11:28:21.49					Starting Bridge MCLNetDB -i "c:\subnets\subnet1\MCLNet.i

ID: 099 - DES Encrypted | System: SERVER1 (192.168.168.44) | Subnet: 001 | 11:33

When launched as a stand-alone application, you are asked to enter the host and port you want to connect to. Once entered the application will attempt to connect to the machine running MCL-Net, when connected the relevant information will be displayed from the Terminals.ini file.

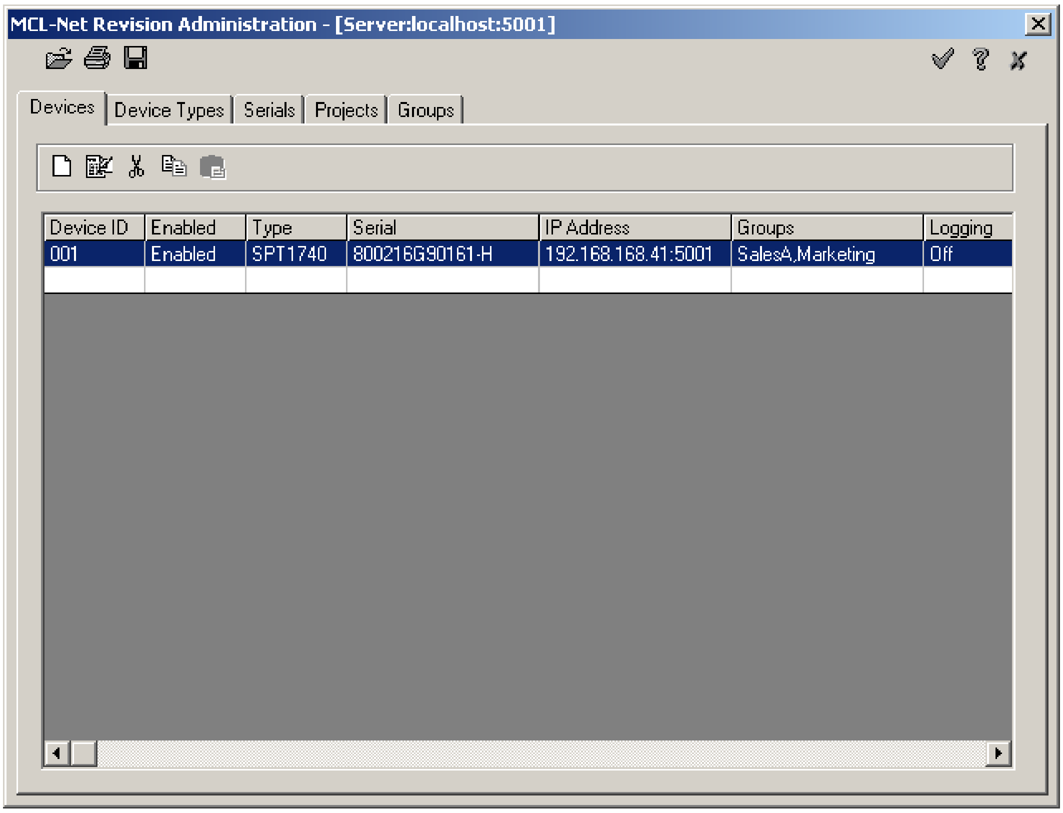


Host Name: 127.0.0.1

Port Number: 5001

Revision Administration will display the following tabs in a tabbed dialog window.

Devices
Device Types
Serials
Projects
Groups



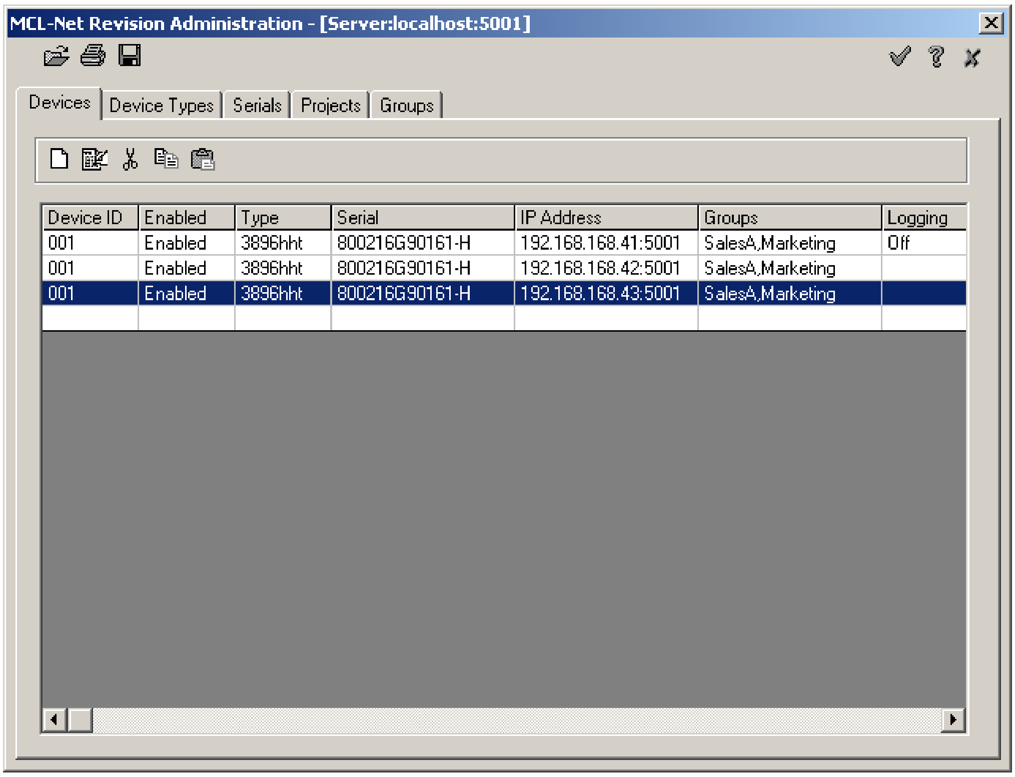
Device ID	Enabled	Type	Serial	IP Address	Groups	Logging
001	Enabled	SPT1740	800216G90161-H	192.168.168.41:5001	SalesA,Marketing	Off

If you want to connect to another host click on the open folder icon in the toolbar, this will allow you to edit another Terminals.ini file for another running instance of MCL-Net.
At any time, if you want to save to the Terminals.ini file click on the disk icon in the toolbar, or quit and you will be prompted to save your changes before continuing.

Revision Administration when started as a stand-alone application will search the current directory for the MCLNet.ini file. If it exists, it will use the connection parameters from the file to attempt to connect to MCL-Net. If the connection fails, or there’s no INI file the program prompts the user to enter the connection parameters for that instance of MCL-Net;

4.2 MCL Devices Tab

The MCL Devices tab accesses information about each individual MCL-Client.



Device ID	Enabled	Type	Serial	IP Address	Groups	Logging
001	Enabled	3896hht	800216G90161-H	192.168.168.41:5001	SalesA,Marketing	Off
001	Enabled	3896hht	800216G90161-H	192.168.168.42:5001	SalesA,Marketing	
001	Enabled	3896hht	800216G90161-H	192.168.168.43:5001	SalesA,Marketing	

This window lists the address information of all the MCL-Client Devices defined for MCL-Net. This list allows MCL-Net to make the correspondence between the MCL-Client ID and the IP address of each device.

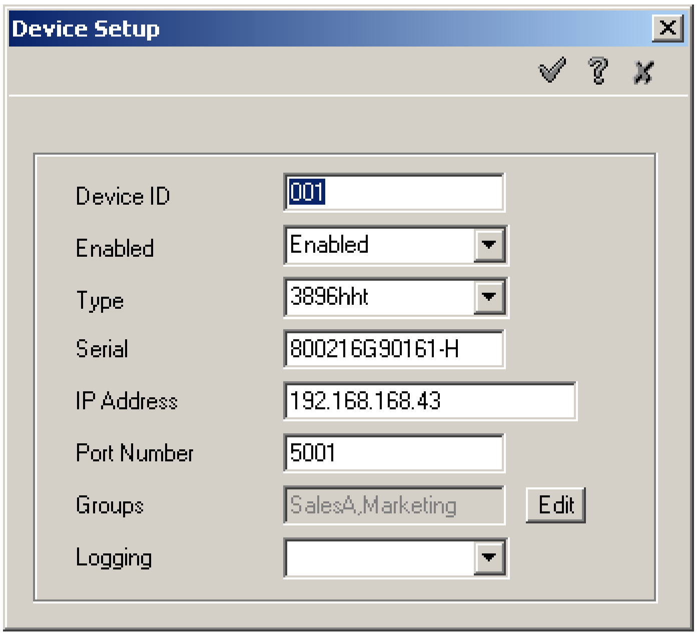
The MCL-Client ID must be unique in the system and in the range from 001 to 255, with one or more ID's used for the host. The IP Address should also be unique and the range depends on your Network Class.

Note: The ID and IP address values also must be defined on the MCL-Client itself.

Double Clicking on a MCL-Client line calls the MCL-Client entry screen. This screen allows you to define the MCL-Client ID, the IP address, the MCL-Client type, the device status (enable or disable), and the serial number of the MCL-Client, logging (on or off) and the default group assigned to the MCL-Client (optional parameter). Each field is enterable except for groups; only groups defined in the groups tab can be selected. Clicking the edit button does this. When a MCL-Client is added, if its IP is not unique a warning message will appear. The MCL-Client ID should also be unique, but if the MCL-Client ID is not unique, only the last entry for that MCL-Client ID will be saved to the Terminals.ini file.

To call a blank MCL-Client Setup screen, highlight the last line (the blank row) in the grid and click the first icon in the icon menu. On the MCL-Client Setup screen the only mandatory field is the MCL-Client ID field. The entry defaults to being enabled if the enabled field in left blank.

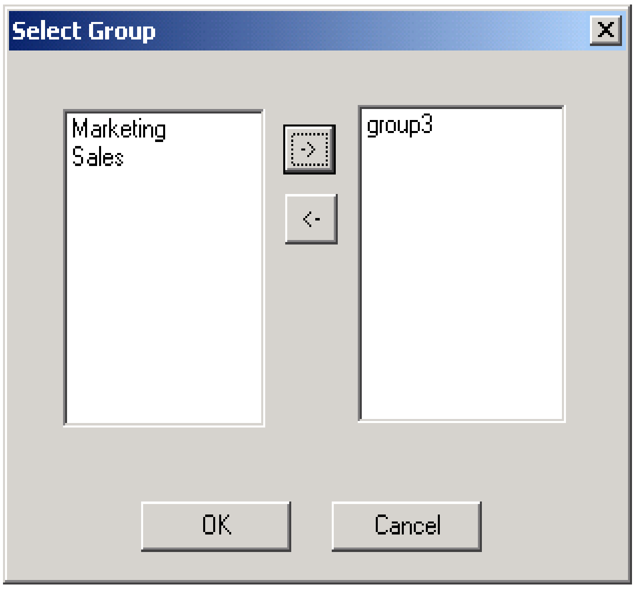
Once a MCL-Client Device ID is defined it can be associated with a MCL-Client type range. This range in turn can hold one or more groups up to a maximum of 20. Each group can be associated with up to 10 Projects.



Device Setup

Device ID	001
Enabled	Enabled
Type	3896hht
Serial	800216G90161-H
IP Address	192.168.168.43
Port Number	5001
Groups	SalesA,Marketing
Logging	

On the MCL-Client Device entry screen the edit button lists the available groups, and the groups assigned to this MCL-Client ID, highlighting the entry and clicking on the directional buttons can change these.



Select Group

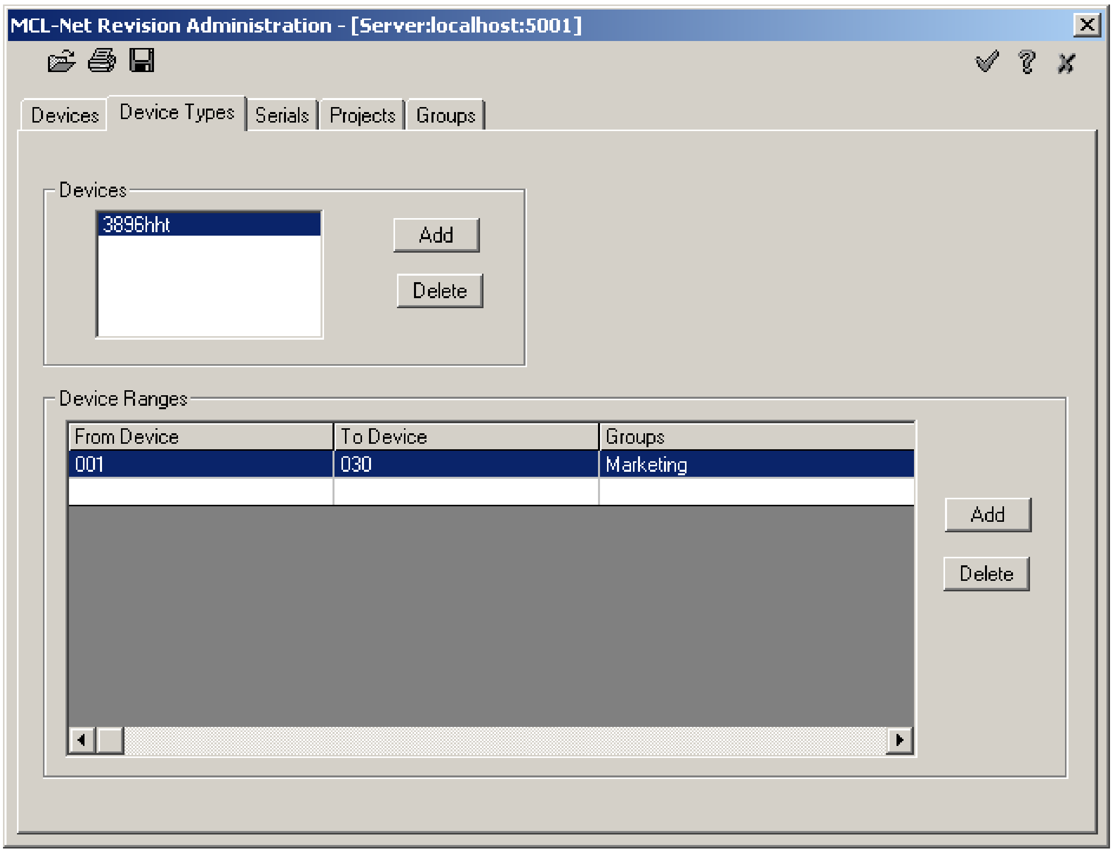
Marketing
Sales

group3

4.3 The Devices Types Tab

The Devices Types tab allows the user to define the various MCL-Client types that can communicate with MCL-Net. For each MCL-Client type a MCL-Client range can be specified. These ranges cannot overlap. If a MCL-Client ID of a certain type connects with MCL-Net, the terminals.ini file can be checked for the associated group. If a MCL-Client ID is not in a specified range and a default for the MCL-Client type is specified, the group associated with the default, is then associated with this MCL-Client ID.

For each range a group has to be specified, if no group is specified, the MCL-Client type and range will not be saved to the Terminals.ini file.

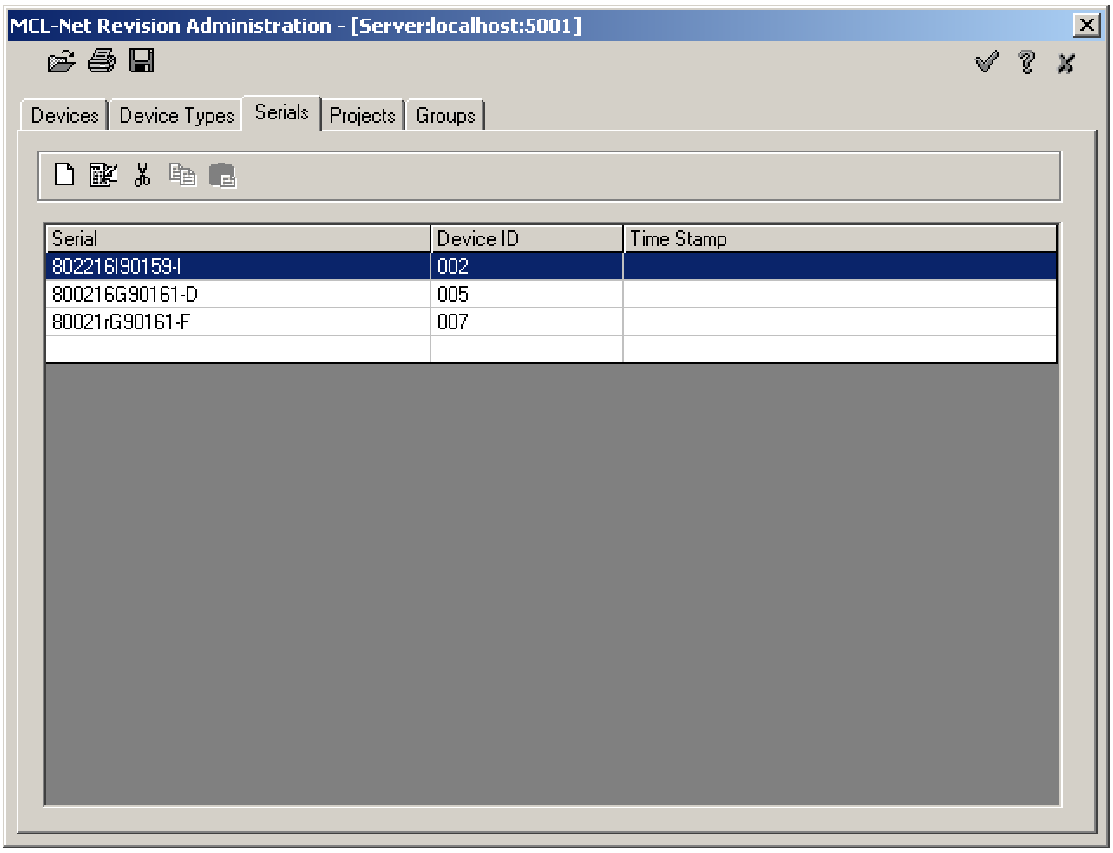

MCL-Net Revision Administration - [Server:localhost:5001]

From Device	To Device	Groups
001	030	Marketing

Double clicking on the MCL-Client Device Ranges grid brings up the select groups screen as defined above.

4.4 Serials Tab

The Serials tab is simply used to maintain a cross reference of the physical Serial Number against an MCL-Client ID.



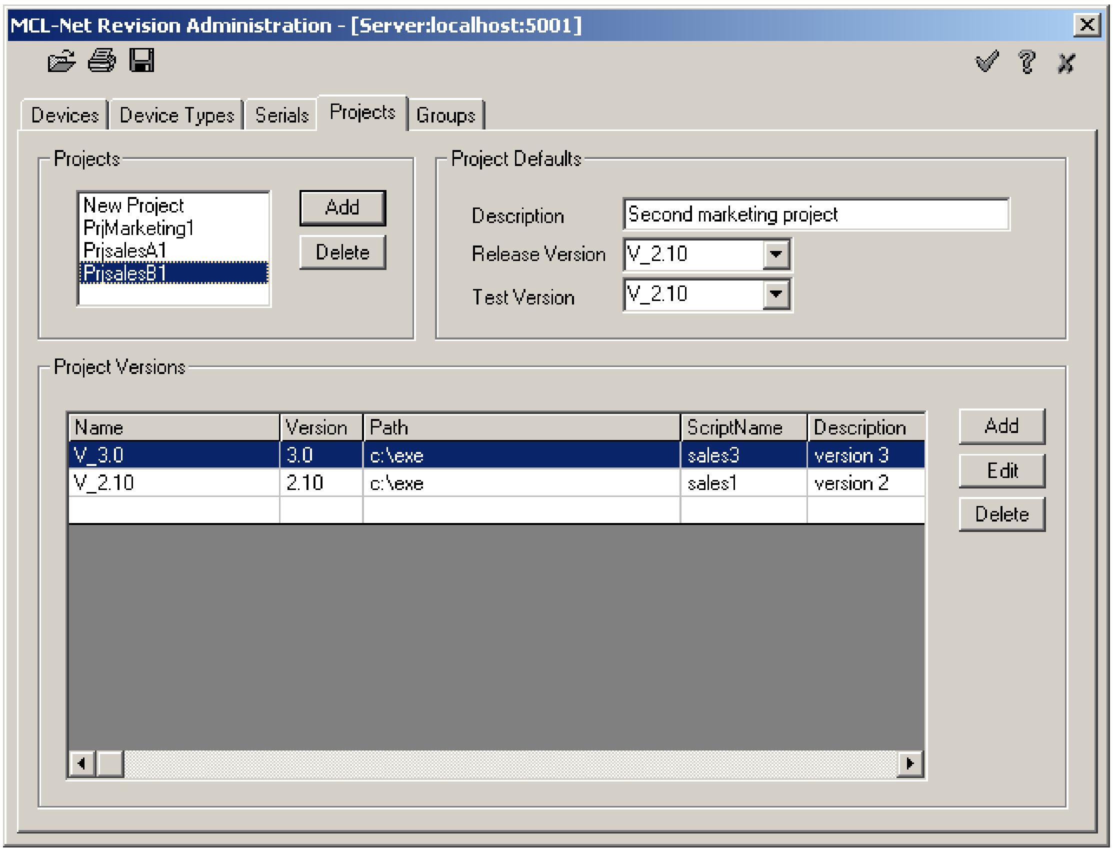
Serial	Device ID	Time Stamp
802216I90159-I	002	
800216G90161-D	005	
80021rG90161-F	007	

To edit the Serial or TID double click on the grid

4.5 Projects Tab

The Project tab is used to define the Projects. Each Project name must be unique. For each defined Project there can be a number of different versions, these versions are displayed in a grid, which can be added to and deleted as the Project is modified.

Once a Project is added to the Project list it is accessible to the groups, alternatively if a Project is deleted, it is removed from the group. If a group is associated with no Projects it will be removed. So removal of Projects should be done carefully.



Projects

- New Project
- PrjMarketing1
- PrjsalesA1
- PrjsalesB1

Project Defaults

Description: Second marketing project

Release Version: V_2.10

Test Version: V_2.10

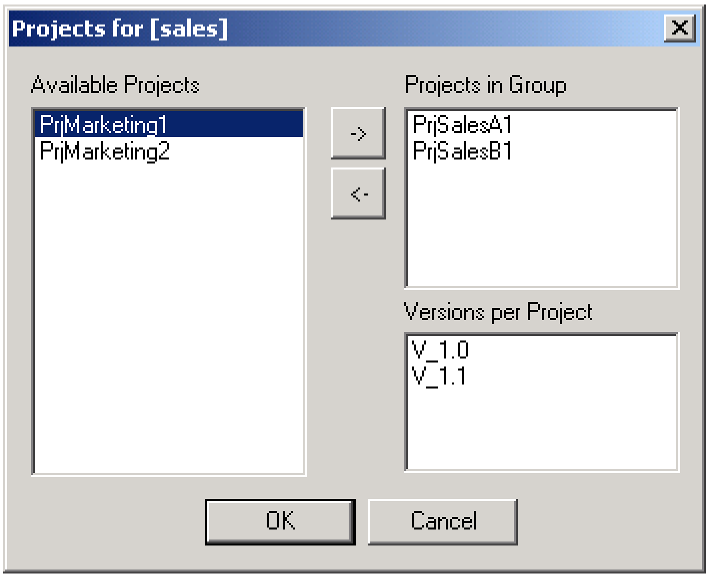
Project Versions

Name	Version	Path	ScriptName	Description
V_3.0	3.0	c:\exe	sales3	version 3
V_2.10	2.10	c:\exe	sales1	version 2

To edit a Project version, either double click on the grid or highlight a row in the group and click the edit button.

4.6 Groups Tab

The Groups tab is used to maintain the group entries in the Terminals.ini file.
Projects are associated with groups on the screen, to edit a group double click on the group entry or highlight the group and click the edit button. This calls the select Project screen.



Projects for [sales]

Available Projects

PrjMarketing1
PrjMarketing2

Projects in Group

PrjSalesA1
PrjSalesB1

Versions per Project

V_1.0
V_1.1

To add a Project, highlight an available Project, highlight a version of this Project and use the directional arrow to add. These Projects are added to the TGroup section of Terminals.ini on saving or when exiting the application.

Each group name must be unique and be associated with at least one Project. If a group is removed it is no longer available to MCL-Client Type ranges, if a mistake is made exit the application without saving.

5 MCL-Net Viewer

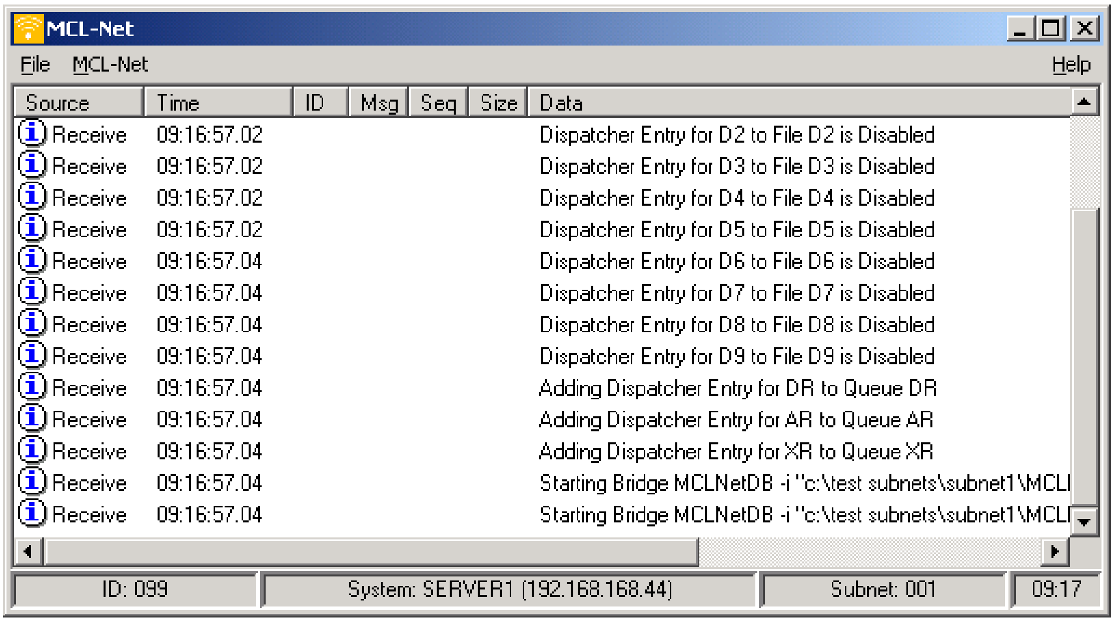
The MCL-Net Viewer window appears when MCL-Net is launched on a Windows platform. The window displays MCL-Net traffic in a user-friendly environment with directional icons to show the flow of messages through the Server. The types of messages displayed are Application Messages (TCP), MCL-Client Messages (UDP) and Internal Messages (Queue messages). UDP messages are messages that flow between MCL-Net and the MCL-Clients (the messages with blue directional arrows), TCP messages flow between Host applications and MCL-Net, and Queue messages are messages that flow internally within MCL-Net between its various modules.

MCL-Net can run in a number of different modes;

1. As a normal Windows process with an interface window
2. As a Windows Service with no interface window
3. As a process started by a different Service, the MCL-Agent, which again has no window
4. As a Window on a process, service or remote MCL-Net

The MCL-Net core Server is the same in each instance. What is different is the existence or lack of a user interface window. The Window also displays different information depending how the MCL-Net is running.
When MCL-Net is first installed, it may be run in demonstration mode for one hour, after which it will automatically shuts down. If running in demonstration mode, the Title Bar displays MCL-Net – Demonstration Mode.

The following is the MCL-Net window that is displayed when a user starts MCL-Net subnet 25 from a typical shortcut when it is not installed as a service



Source	Time	ID	Msg	Seq	Size	Data
Receive	09:16:57.02					Dispatcher Entry for D2 to File D2 is Disabled
Receive	09:16:57.02					Dispatcher Entry for D3 to File D3 is Disabled
Receive	09:16:57.02					Dispatcher Entry for D4 to File D4 is Disabled
Receive	09:16:57.02					Dispatcher Entry for D5 to File D5 is Disabled
Receive	09:16:57.04					Dispatcher Entry for D6 to File D6 is Disabled
Receive	09:16:57.04					Dispatcher Entry for D7 to File D7 is Disabled
Receive	09:16:57.04					Dispatcher Entry for D8 to File D8 is Disabled
Receive	09:16:57.04					Dispatcher Entry for D9 to File D9 is Disabled
Receive	09:16:57.04					Adding Dispatcher Entry for DR to Queue DR
Receive	09:16:57.04					Adding Dispatcher Entry for AR to Queue AR
Receive	09:16:57.04					Adding Dispatcher Entry for XR to Queue XR
Receive	09:16:57.04					Starting Bridge MCLNetDB -i "c:\test subnets\subnet1\MCLI
Receive	09:16:57.04					Starting Bridge MCLNetDB -i "c:\test subnets\subnet1\MCLI

ID: 099 | System: SERVER1 (192.168.168.44) | Subnet: 001 | 09:17

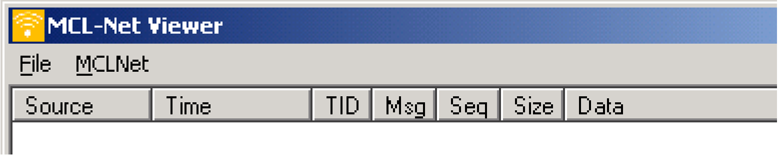
The Status Bar at the bottom is split into 3 sections;

The Subnet Number
The System Information
The Host ID number

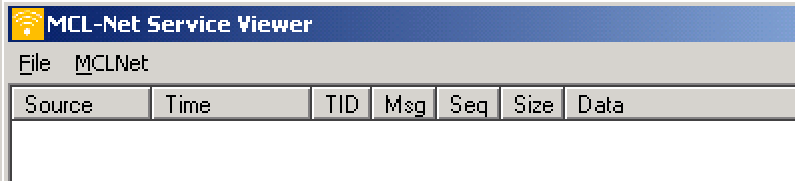
The main window area displays Error, Warning and Information messages as well as MCL-Client Traffic between MCL-Net and MCL-Clients.

The data on the Title bar will change depending on how MCL-Net is started. The above window is a normal foreground Windows process.

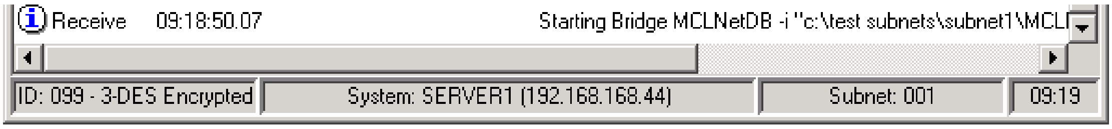
If the user starts the same MCL-Net a second time, the second instance will only create a viewer window on the first MCL-Net. The title bar changes as shown below (MCL-Net Viewer is follows;



If this viewer is a viewer on a service rather than a viewer on a foreground MCL-Net then the title bar changes as follow;

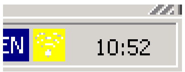


The Status Bar also changes. It will display the Encryption Level if MCL-Net is running in Encrypted Mode;

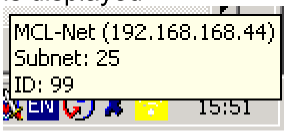

Receive 09:18:50.07 Starting Bridge MCLNetDB -i "c:\test subnets\subnet1\MCL

ID: 099 - 3-DES Encrypted | System: SERVER1 (192.168.168.44) | Subnet: 001 | 09:19

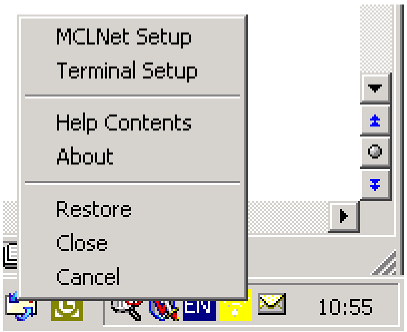
When MCL-Net starts, it also creates a tray icon;



On “hovering” over the Icon, the MCL-Net Host IP Address, Subnet Information and ID string is displayed


MCL-Net (192.168.168.44)
Subnet: 25
ID: 99

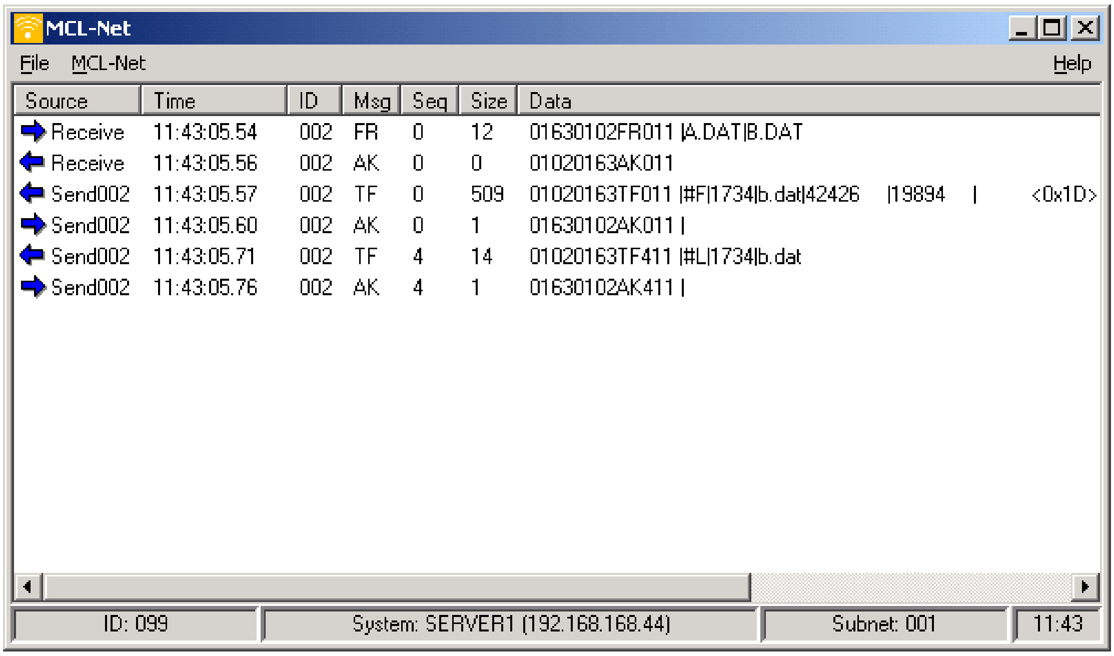
This Icon also contains the Menu Items from the MCL-Net Main Window (Right Click)



In the MCL-Net window the screen is divided into seven categories

Source
Time
ID (may be shown at TID depending on version of MCL Net)
Msg
Seq
Size
Data.

The **Source** tells the user what MCL-Net thread is handling the message and in what direction it is going.



```
Source     Time         ID    Msg  Seq  Size  Data
Receive    11:43:05.54  002   FR   0    12    01630102FR011 |A.DAT|B.DAT
Receive    11:43:05.56  002   AK   0    0     01020163AK011
Send002    11:43:05.57  002   TF   0    509   01020163TF011 |#F|1734|b.dat|42426    |19894    |        <0x1D>
Send002    11:43:05.60  002   AK   0    1     01630102AK011 |
Send002    11:43:05.71  002   TF   4    14    01020163TF411 |#L|1734|b.dat
Send002    11:43:05.76  002   AK   4    1     01630102AK411 |

ID: 099     System: SERVER1 (192.168.168.44)     Subnet: 001     11:43
```

For line one of our example window above there is a FR (File Request) message from MCL-Client ID 33 coming into the receive thread, while line two has an AK message leaving the receive thread in acknowledgement of the FR. The **Time** section displays when the message is handled. The **ID** shows the MCL-Client ID that is sending or requesting data, the **Msg** is the type of message that is being handled and the **Seq** shows the message sequence number.

The **Size** shows the size of the Data in the message (not including any header information). Finally, the **Data** section shows the entire Packet, including the message type, the sequence number, the message length and the message packet.

The on-screen display length of the messages can be modified by changing the LogMessageSize value in the Logging section of the MCLNet.ini file.

On the File menu there are three options;

- MCL-Net Setup calls the MCL-Net Setup application and allows the user to change the configuration of MCL-Net.

- MCL-Client Setup calls the Revision Administration application, which allows users to administer MCL-Clients.

- Exit stops MCL Net and closes the application.

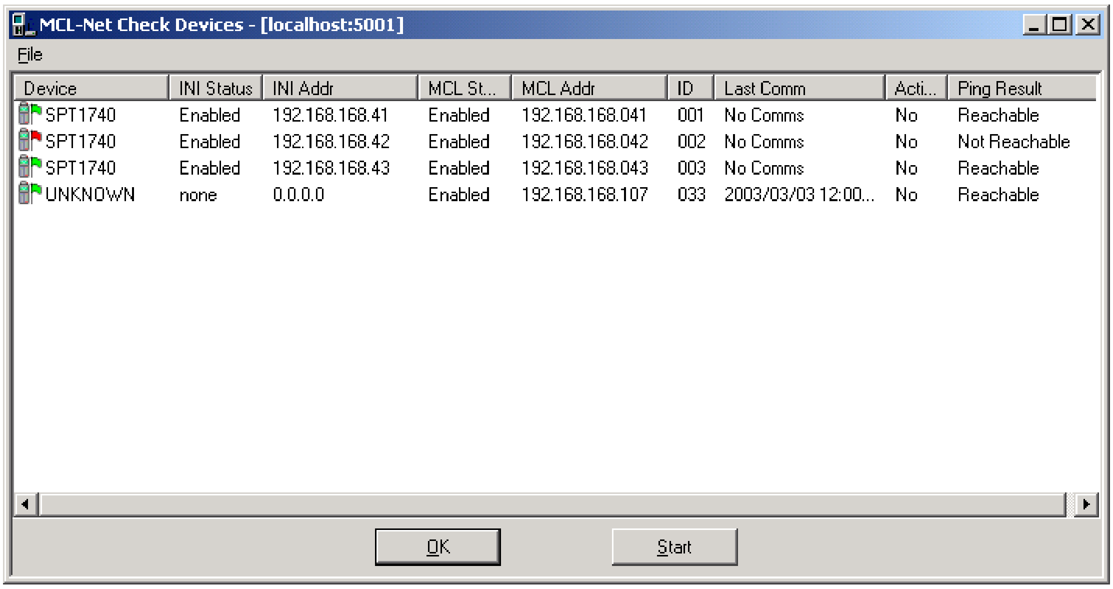
On the MCL-Net menu there are four options:

- Clear – clears the current messages from the MCL-Net window.
- Check MCL-Clients – Starts the MCLPing application
- Install Product – see help on MCLKey.
- Install Add-On – see help on MCLKey.

To check the version of MCL-Net that is running, click on help, then about.

5.1 MCL-Ping

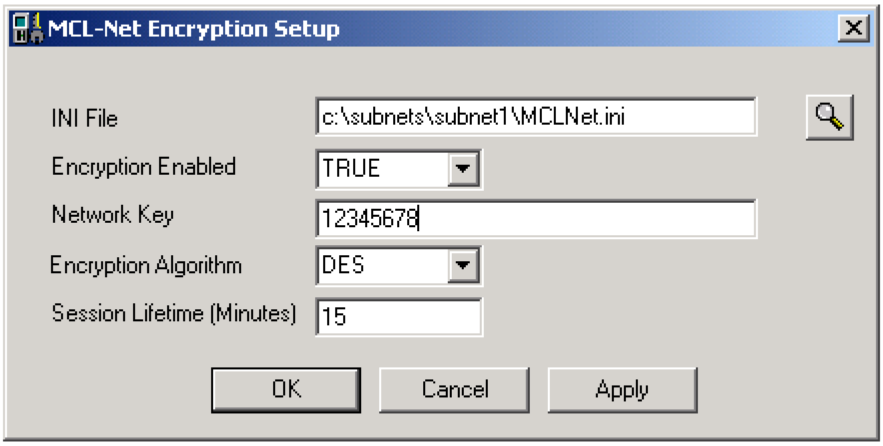
MCL-Ping is an application that can be started from the MCL-Net Viewer window using: MCL-Net->Check Devices. On starting the application the following screen is displayed. Periodically the application sends a message to MCL-Net that causes MCL-Net to ping the MCL-Client in question. The result of the ping is displayed on screen. Other useful information is also displayed including the INI settings for the device as well as the MCL-Net settings for a device. In the following example, the fourth device does not actually have an INI entry, yet the device at some stage successfully communicated with MCL-Net.



Device	INI Status	INI Addr	MCL St...	MCL Addr	ID	Last Comm	Acti...	Ping Result
SPT1740	Enabled	192.168.168.41	Enabled	192.168.168.041	001	No Comms	No	Reachable
SPT1740	Enabled	192.168.168.42	Enabled	192.168.168.042	002	No Comms	No	Not Reachable
SPT1740	Enabled	192.168.168.43	Enabled	192.168.168.043	003	No Comms	No	Reachable
UNKNOWN	none	0.0.0.0	Enabled	192.168.168.107	033	2003/03/03 12:00...	No	Reachable

6 Network Key

This is a stand-alone application that is used to edit the Network Key used for encryption on the local machine. In MCLNet.ini the Network Key is encrypted to give a further level of security. For a user to edit the Network key this application should be used and the Key can be viewed and changed if necessary.



INI File: c:\subnets\subnet1\MCLNet.ini

Encryption Enabled: TRUE

Network Key: 12345678

Encryption Algorithm: DES

Session Lifetime (Minutes): 15

To change Network Key, type in the path of the ini file or locate it using the magnifying key, once located the Network Key for this file will appear. To change, type in new key and ok (save and exit), this will exit the application. To change a number of keys in different ini files, locate the file, change the key and hit apply (save), once finished exit the application.

7 MCLSamples

The MCLSamples application is provided for both Windows and Unix platforms, to give users sample applications and code, which provide examples of how to interface with MCL-Net at packet level. The function calls and parameters are defined in the MCL-Net Technical documentation. Here we will outline the MCLSample application itself.

7.1 To Load and Run MCLSamples

As part of the standard installation process the code for MCLSamples is also installed. If you have Visual Studio or Microsoft Visual C++ installed on your PC, you can load the MCLSamples Project workspace by clicking on MCLSamples.dsw. Once loaded in the workspace the following appear, Source Files, Header files and resource files as part of the MCLSamples file. By right clicking on the MCLSamples file and choosing option settings you bring up the Project settings of MCLSamples.

Under the general tab in the output files field you specify where you are creating your executable MCLSamples application.

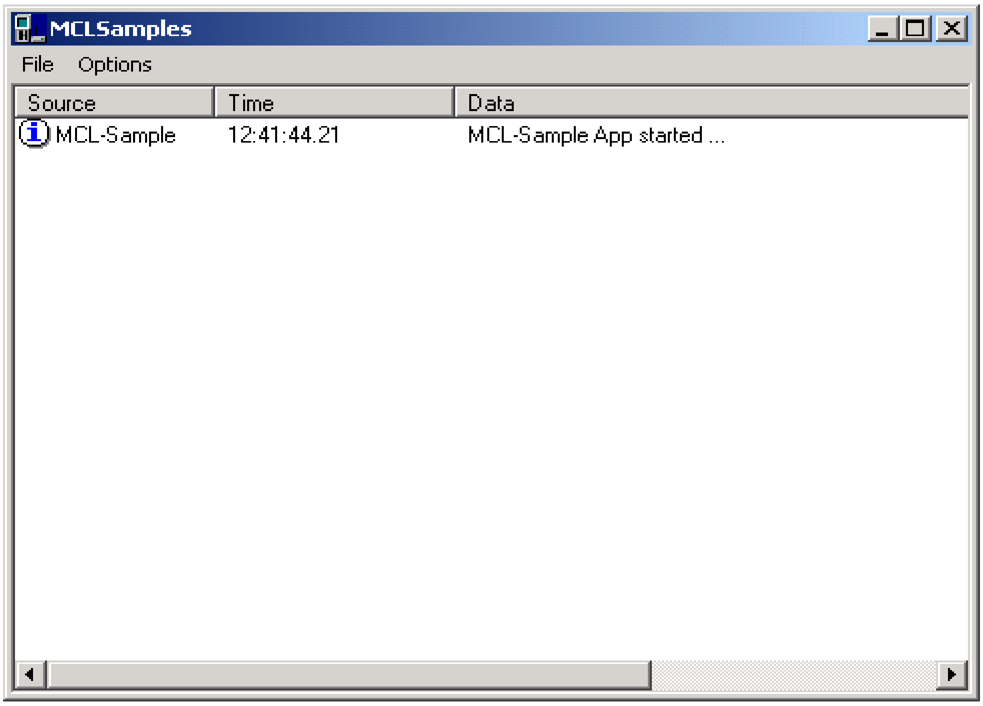
Under Debug in the program arguments field you can specify the ini file to use, for example if it is in the current directory –i MCLApp.ini. A preferable way to specify this is to leave this field blank and create a short cut to the executable and in its properties/target specify the ini path after the executable programs path. This method is more robust and easier to change.

Under the C/C++ tab and category pre-processor, the path for the mcl.h file needs to be specified, if it is in the current directory where the MCLSamples.dsw resides this field can be blank.

You also need to link to the NetS24V3.lib, this should be specified under the Link tab in the Object/Modules library field. NetS24V3.dll also need to be in the path or reside it the directory where you are running the Samples application.

7.2 Starting MCLSamples for windows

MCLSamples is a stand-alone application, which connects to MCL-Net using parameters from an ini file called MCLApp.ini. In MCLApp.ini you will need to specify the host name and port where the MCL-Net is running, which you want to connect to. Once this is done the application can be started and the following window should appear.



Source	Time	Data
MCL-Sample	12:41:44.21	MCL-Sample App started ...

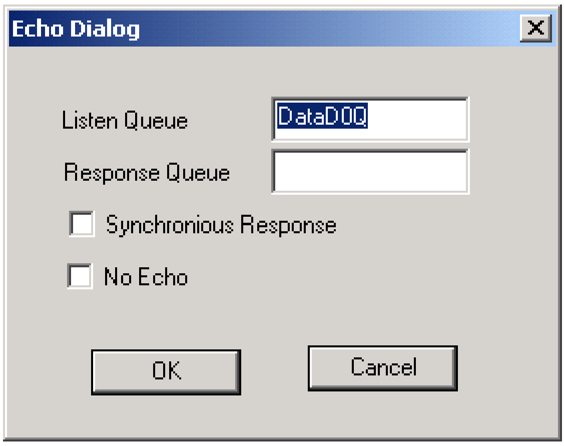
This is the main MCLSamples window. There are two menu options, File and Options. From the File menu you can clear the screen and exit the application. From the options you can choose one of the following:

Echo	- used to read from a specified Queue running in MCL-Net
Check	- Check the status of the MCL-Net you are trying to connect to
Check and Start	- Check the status, if MCL-Net is not running, try to start the Server
Remove Files	- deletes a file on the machine the Server is running
Check MCL-Client	- checks if a MCL-Client is connected
Shutdown	- stops MCL-Net you are connecting to
Reboot	- sends a reboot command to MCL-Net
Retrieve IP	- retrieves the IP and port number of a MCL-Client
List Files	- creates a file in the DEFPATH on the host machine with a list of files which exist in a specified directory
App File to Net	- app File To Net is used to transfer a file on the local machine to the connected host machine
Net File to App	- net File To App works the same as App File to Net, but in the opposite direction
Wait MCL-Client	- wait for a MCL-Client to connect to MCL-Net

7.3 Echo

The Echo sample is used to read from a specified Queue running in MCL-Net. Once a message is read it is displayed and can be echoed back to MCL-Net.

To start the echo click on options – echo – start echo; this will display the following screen.


Echo Dialog

Listen Queue DataD0Q

Response Queue

□ Synchronious Response

□ No Echo

The listen Queue is the queue on which you want to listen for a particular message. The queue should correspond to a Queue defined in the dispatcher section of the MCLNet.ini for the MCL-Net which you are communicating with. For the given example DataD0Q could correspond to the following entry:

```
[Dispatcher]
Dispatch_001=D0|Queue|DataD0Q|000|No|
```

The Response Queue is the Queue, which the Echo program will send its response to. Synchronous Response responds in Synchronous mode ensuring the MCL-Client received the packet.
No Echo - does not send an echo response to MCL-Net.

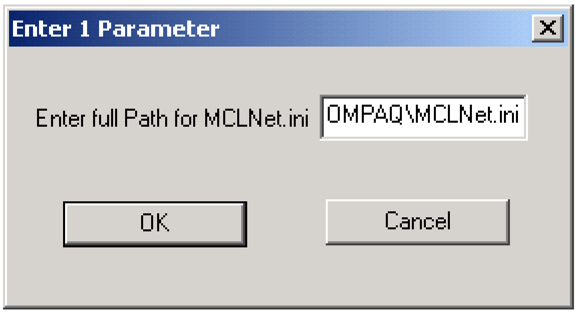
Once start Echo is initiated it attempts to connect to the running MCL-Net, if it is successful it will wait on the specified queue for incoming messages until the Stop echo option is chosen. If Start Echo cannot connect to MCL-Net, it logs a message and stops.

7.4 Check

If you want to check the status of the MCL-Net you are trying to connect to, use the check option. If MCL-Net is running, The Command Completed OK will be returned; else a message like MCL-Net not started will be displayed.

7.5 Check and Start

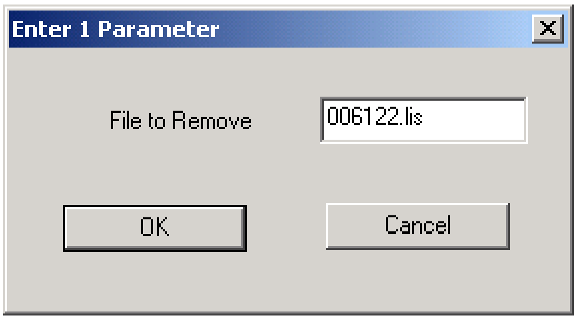
Check and Start is similar to Check, but if MCL-Net is not running, this option will try to start the Server. On the local machine the BINPATH in MCLApp.ini needs to specify the path to where the MCL-Net executables resides. The path for the MCLNet.ini file for MCL-Net you wish to start also needs to be entered on the following screen.


Enter 1 Parameter

Enter full Path for MCLNet.ini OMPAQ\MCLNet.ini

7.6 Remove Files

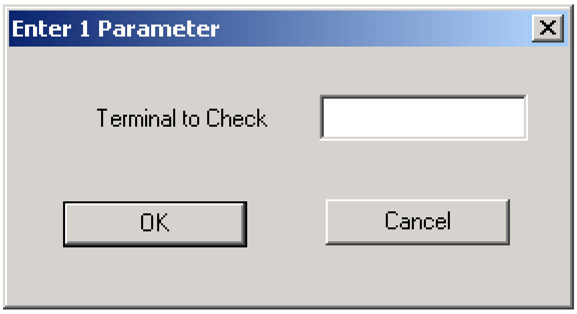
Remove Files deletes a file on the machine on which MCL-Net is running. If no path is specified the file is expected to reside in the DEFPATH of the connected MCL-Net machine.



File to Remove 006122.lis

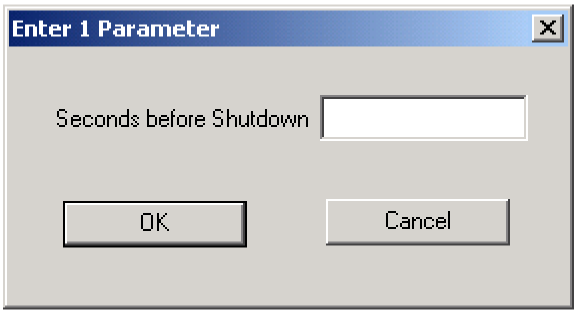
7.7 Check MCL-Client

When a MCL-Client connects to MCL-Net, it connects with a TC command. When Check MCL-Client is ran the user enters the MCL-Client number it wants to check, if this MCL-Client has connected, the message ‘the Command Completed OK’ is displayed, else The command returned with a not found error.



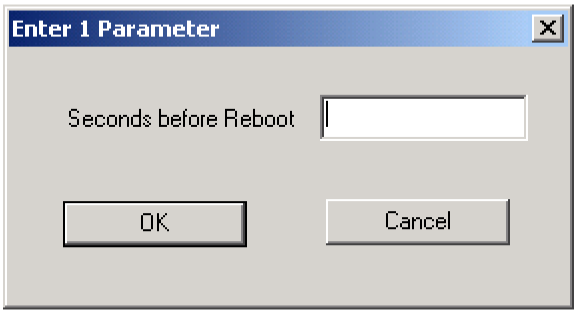
7.8 Shutdown

The Shutdown command stops the MCL-Net instance with which you are connected according to the MCLApp.ini file.



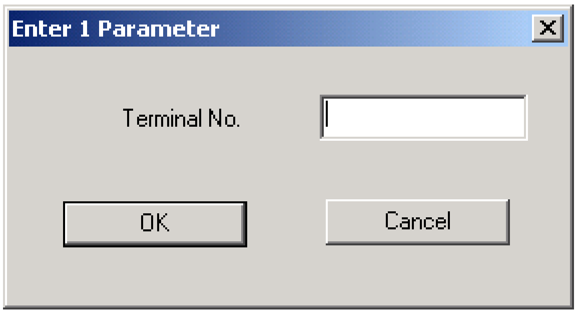
7.9 Reboot

Reboot sends a reboot command to MCL-Net. Once the reboot is done the samples window displays the Command Completely successfully.



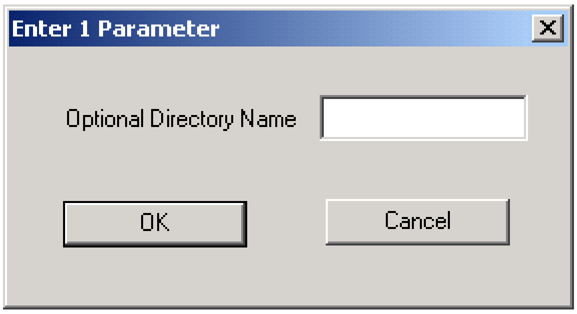
7.10 Retrieve IP

Retrieves the IP and port number of the MCL-Client entered in the pop up window. If The MCL-Client IP returns as 000.000.000.000:5001, which is the default for an undefined MCL-Client, then the MCL-Client has never communicated with MCL-Net since the last Reboot/start.



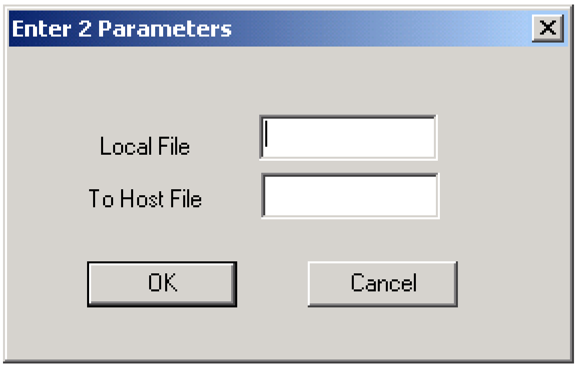
7.11 List Files

Creates a file in the DEFPATH on the host machine with a list of files, which exist, in a specified directory. The created file name is shown in the samples window and has an extension of .lis. The list of files will also specify the file size and the date and time created. If no directory path is given, the created file will show the files in the DEFPATH directory. If the directory does not exist a file is created with no entry.



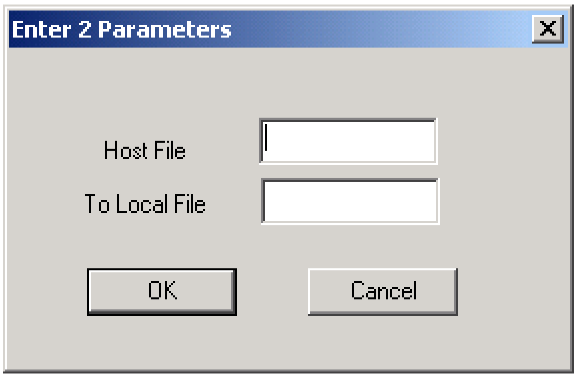
7.12 App File to Net

App File To Net is used to transfer a file on the local machine to the connected host machine. If no directory path is given for the local file, DEFPATH defined in MCLApp.ini is the default. If no file can be found for the given file name, a file of size 0 will try to be created on the host machine in its DEFPATH. If no host file is specified the file cannot be created on the host machine. IF you want to put the file in another directory besides DEFPATH, you need to specify the full correct path and file name in the To Host File field.



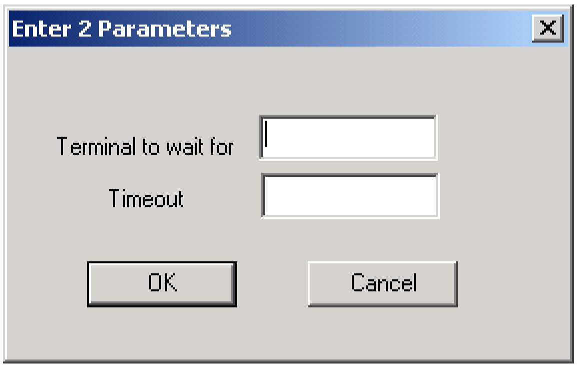
7.13 Net File to App

Net File To App works the same as App File to Net, but in the opposite direction.



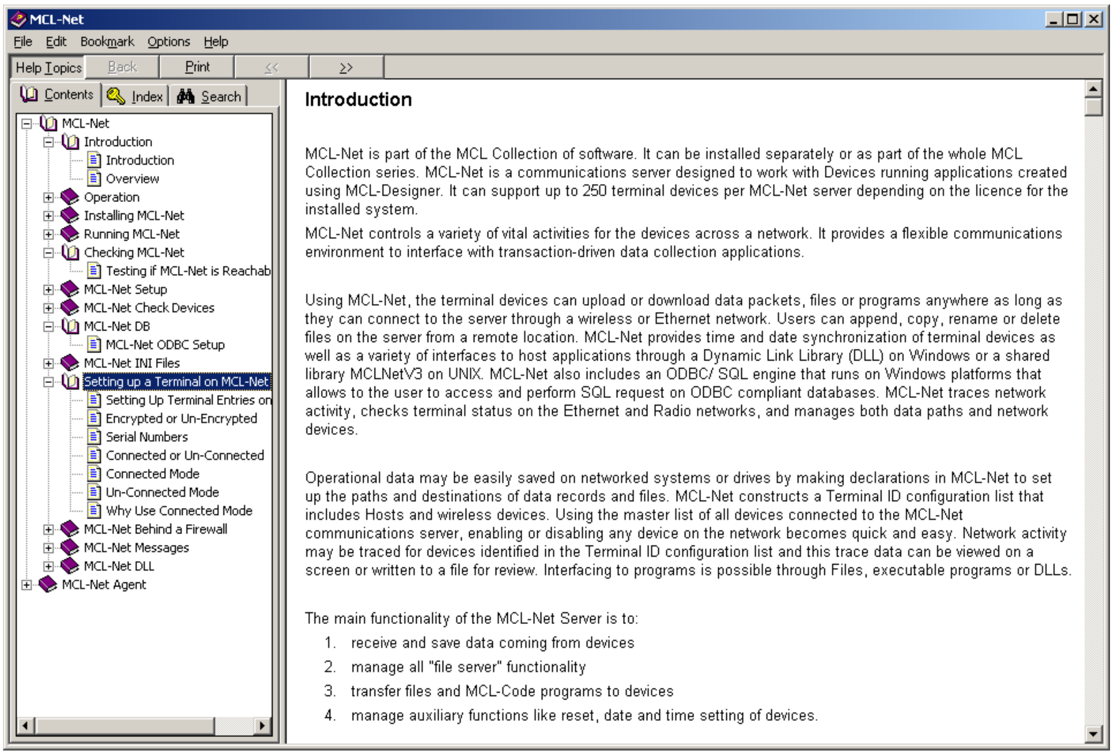
7.14 Wait MCL-Client

Wait MCL-Client sends a wait MCL-Client message to MCL-Net until a MCL-Client responds or the timeout expires. The timeout is in milliseconds.



8 Help Files

MCL-Net includes a comprehensive help file, which encompasses everything in this manual and much more. It can be accessed by selecting the drop down menu entitled “Help” from the top right hand corner of the MCL-Net Viewer window



Introduction

MCL-Net is part of the MCL Collection of software. It can be installed separately or as part of the whole MCL Collection series. MCL-Net is a communications server designed to work with Devices running applications created using MCL-Designer. It can support up to 250 terminal devices per MCL-Net server depending on the licence for the installed system.

MCL-Net controls a variety of vital activities for the devices across a network. It provides a flexible communications environment to interface with transaction-driven data collection applications.

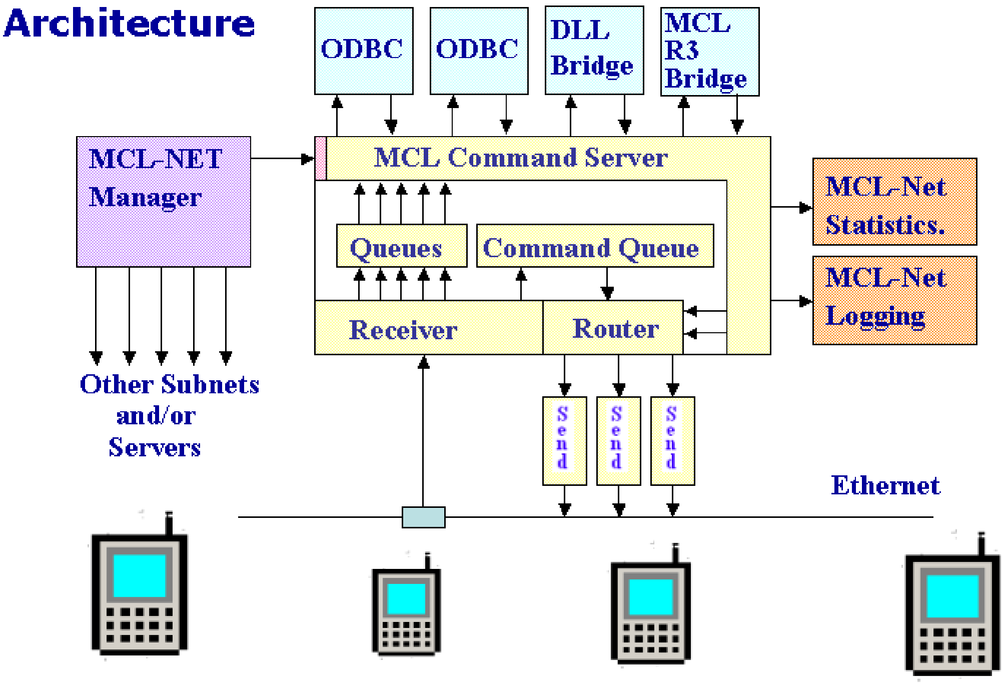
Using MCL-Net, the terminal devices can upload or download data packets, files or programs anywhere as long as they can connect to the server through a wireless or Ethernet network. Users can append, copy, rename or delete files on the server from a remote location. MCL-Net provides time and date synchronization of terminal devices as well as a variety of interfaces to host applications through a Dynamic Link Library (DLL) on Windows or a shared library MCLNetV3 on UNIX. MCL-Net also includes an ODBC/ SQL engine that runs on Windows platforms that allows to the user to access and perform SQL request on ODBC compliant databases. MCL-Net traces network activity, checks terminal status on the Ethernet and Radio networks, and manages both data paths and network devices.

Operational data may be easily saved on networked systems or drives by making declarations in MCL-Net to set up the paths and destinations of data records and files. MCL-Net constructs a Terminal ID configuration list that includes Hosts and wireless devices. Using the master list of all devices connected to the MCL-Net communications server, enabling or disabling any device on the network becomes quick and easy. Network activity may be traced for devices identified in the Terminal ID configuration list and this trace data can be viewed on a screen or written to a file for review. Interfacing to programs is possible through Files, executable programs or DLLs.

The main functionality of the MCL-Net Server is to:

1. receive and save data coming from devices
2. manage all "file server" functionality
3. transfer files and MCL-Code programs to devices
4. manage auxiliary functions like reset, date and time setting of devices.

9 Message Flow and Internal Architecture of MCL Net



In an MCL world, transactions are typically driven by the MCL-Client. Messages originating from the Client are passed over the Ethernet network destined for a particular instance of MCL-Net. They are received at the **Receiver** and forwarded to the appropriate queue. Messages that MCL-Net can immediately deal with, without any further interfacing are sent to the **Command Queue** and responded to immediately back to the MCL-Client via the **Router Module**. Other messages are queued for the **Command Server**, which decides what to do with each message based on its packet type and then forwards the message to the appropriate MCL bridge component or application. Statistical data and message logging is also collected by the **Command Server.** Information resulting back from the various MCL Bridge products and Applications is accepted back by the **Command Server** and forwarded to the **Router** for onward transmission to the MCL Client.

Also included within MCL-Net is the Net Agent that allows the **MCL-Net Manager** software module to administer and view MCL-Net and MCL-Client activity.

MCL-Net Manager

This is a stand-alone management program that presents a “Tree View” of all computers on the network that have MCL-Net installed on them and reports the current status of those Subnets. It also presents a view of all devices on each subnet and can query the status of each device. It can change the configuration of each MCL-Net Subnet.

The Net Agent (represented in pink in the diagram above) is a program that runs as a Service on the host computer where one or more instances of MCL-Net (Subnets) are running. Both local and remote MCL-Net Manager programs use it to administer all MCL-Net Subnets (instances) on the local computer.

For more details on the MCL-Net Manager see separate manual or onscreen help files.

UDP Messages
UDP messages are messages that flow between the MCL-Net and the MCL D-Clients. These messages have blue directional arrows in the MCL-Net Viewer Window.

TCP Messages
TCP messages flow between Host applications and MCL-Net.

Queue Messages
Queue messages are messages that flow internally on MCL-Net between its various modules.

Other Messages
There are other general messages used in MCL-Net, these messages do not show network traffic, but inform the user what is happening in the MCL-Net Viewer window. These messages include Warning messages, Information messages and Error message:

10 A Quick Start Installation of MCL-Net

Owing to the scalable nature and complexities of MCL-Net, it is easy to get confused by such a manual. This section has been created to accommodate in a couple of pages the set up of MCL-Net in its simplest form.

10.1 Starting MCL-Net

This section is dependant on you accepting the default values presented during the installation of MCL-Net.

After installation, the following items will exist on the computer;

1. An MCLNET directory containing all binaries, DLLs and Clean INI files required by MCL-Net
2. A subnets directory will exist with 2 sub-directories;
    a. Subnet1 that contains the files and directories required by a single subnet, and a shortcut to start MCL-Net.
    b. Agent, which contains the files and directories required by the MCL-Net Agent Service
3. A shortcut on the desktop to start the MCL-Net subnet1.
4. A MCL-Net Menu item on the Program Menu that contains the MCL-Net Manager and MCL-Net subnet1.
5. The MCL-Agent will be installed as a service and will have been automatically started

To start the first MCL-Net, simply double click on the MCL-Net1 icon defined by 3 above. If MCL-Net is not licensed on the computer, the activation process will automatically start. If MCL-Net is not to be activated immediately, the user can start in Demonstration mode. Note that MCL-Net will automatically terminate after one hours operation in Demonstration mode.

10.2 First Terminal Communication

MCL-Net should by now be started on the computer. In order to check that a terminal can communicate with MCL-Net take the following steps;

1. On Installation, the new subnet is configured with a default Device 001. Using the “File -> Device Set-up” menu, change the IP Address to the actual address of the device or 0.0.0.0 if the device is using DHCP. If you do not need the sample entry, you can delete the entry completely.
2. On the device, from the System Menu;
    a. Define it as ID 001 on Subnet 001 using “Set-up TID”
    b. Selecting “Host Set-up”;
        i. Ensure HostID is 99
        ii. Define the IP Address of the Host
    c. If possible, check that the “Base Port” is set at 5000
3. Again using the System Menu;
    a. Choose the “Tests” Item,
    b. From the lists of tests, choose the Host Connection.

At this point the terminal will attempts to communicate with MCL-Net. If successful, the device will respond with a success message and traffic messages will appear on the MCL-Net Viewer window.

If for some reason the Host Connection test fails, the user should use the MCL-NetDiagnostic application to determine if MCL-Net is responding correctly. If it is, then the problem lies with the device set-up. If it is not, the MCL-NetDiagnostic application will present probably causes for the failure. Each should be checked individually.

10.3 Testing if MCL-Net is Reachable (MCL-NetDiagnostic)

(Please note that in earlier versions, MCLNetdiagnostic was known as MCLNetTest)

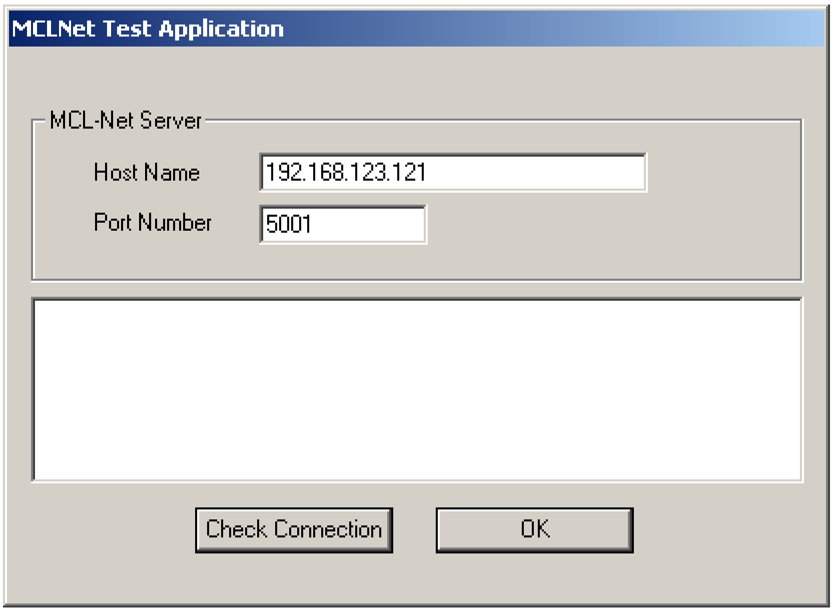
Checking if MCL-Net will respond correctly to a MCL-Client can sometimes be quite difficult, especially it the user is unfamiliar with that particular MCL-Client, the MCL-Client is on a different area of the network or there's a firewall between the MCL-Client and MCL-Net.

A simple application has been provided called MCLNetDiagnostic.exe which runs on a Windows type PC. This application should be taken to the area of the network where the MCL-Client will be connected and installed/launched. The MCLNetDiagnostic program then attempts to send data packets to the MCL-Net. If packets are missed or not returned to the MCL-Net Test program, it offers hints and checks that the user should make to identify the cause of the communication failure.

How to run the Tests

The MCL Net Diagnostic program should not be started on the same computer as MCL-Net as both products will attempt to use the same Network Port. Instead, copy the MCLNetDiagnostic.exe program and the SX32W.DLL to a folder on the remote PC.

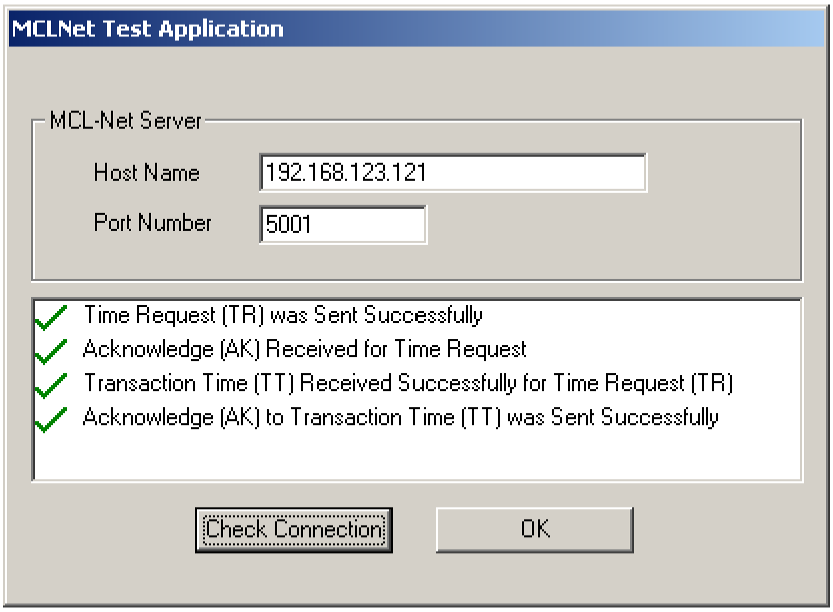
Enter the Host Name or IP Address and the Port on which MCL-Net is listening and click the "Check Connection" button.



Host Name: 192.168.123.121

Port Number: 5001

On Success the following is displayed…



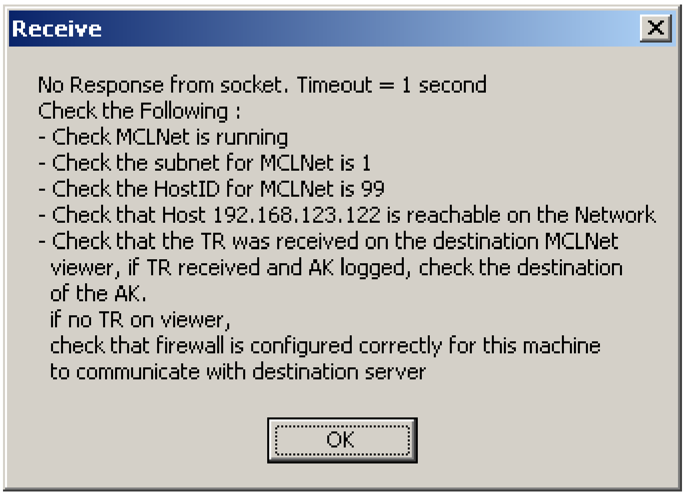
Host Name 192.168.123.121

Port Number 5001

✓ Time Request (TR) was Sent Successfully
✓ Acknowledge (AK) Received for Time Request
✓ Transaction Time (TT) Received Successfully for Time Request (TR)
✓ Acknowledge (AK) to Transaction Time (TT) was Sent Successfully

Errors in communication can be the result of a number of issues;

There may be a firewall between the 2 PCs
MCL-Net may not be listening or not started
MCL-Net may be responding to a different MCL-Client at a different IP Address.
The Checks a user should perform are detailed in a pop-up message box and these checks differ depending on where the error occurred.



No Response from socket. Timeout = 1 second
Check the Following :
- Check MCLNet is running
- Check the subnet for MCLNet is 1
- Check the HostID for MCLNet is 99
- Check that Host 192.168.123.122 is reachable on the Network
- Check that the TR was received on the destination MCLNet
  viewer, if TR received and AK logged, check the destination
  of the AK.
  if no TR on viewer,
  check that firewall is configured correctly for this machine
  to communicate with destination server

11 Glossary of Unfamiliar Terms

MCL-Net

This is the central communication Server. Terminals send messages to this Server and the Server may in turn respond to the messages. Terminals typically communicate with the Server to perform database queries, upload or download files or perform other host type operations. MCL-Net can run as a normal Windows program with a foreground window or as a service. Multiple instances of MCL-Net can run simultaneously on the same host computer. Each instance is known as a Subnet.

MCL-Net Agent

The agent is a program that can runs as a service on the host computer where one or more instances of MCL-Net (Subnets) are running. Both local and remote MCL-Net Manager programs use it to administer all MCL-Net Subnets on the local computer.

MCL-Net Manager

The manager program presents a “Tree View” of all computers that may have MCL-Net Subnets running and the Current Status of those Subnets. It also presents a view of all devices on each subnet and can query the status of each device. It is also possible to change the configuration of a MCL-Net Subnet using the Setup programs available in the MCL-Net Manager.

Subnet

Typically MCL-Net is installed on one PC on the company network. In larger systems, with in excess of 250 MCL-Clients then it may be necessary to invoke a second copy of MCL-Net on the same machine. (A second instance, also known as a subnet)

You may have up to 250 MCL subnets running on a single computer, each with it’s own independent ini files and directory structure.