CSL CS-461 EPC Class 1 Gen 2 RFID Fixed Reader
User's Manual

Version 4.0

CSL: The One-Stop-Shop for RFID Solutions

CONTENT

1 INTRODUCTION 7

1.1 HOW TO USE THIS MANUAL 7

1.2 PRODUCT PACKAGE 8

1.2.1 Basic Package Content 8

1.2.2 Unpacking Instructions 8

1.3 PRODUCT SPECIFICATION 9

2 INSTALLATION 11

2.1 DEVICES 11

2.1.1 Reader Connection 11

2.1.2 Antenna Cable Connection 13

2.2 INSTALLATION STEPS 14

2.3 INSTALLATION RECOMMENDATION 16

2.3.1 Antenna Installation 16

2.3.2 IO Connection 19

2.4 VERIFICATION AND VALIDATION 24

2.5 CAUTIONS 27

3 QUICK START 28

3.1 READER LOGIN 28

3.2 FIRMWARE VERSION UPGRADE 29

3.3 SETUP ACCESS MODE 30

3.4 SETUP OPERATION PROFILE 31

3.5 SETUP TRIGGER 33

3.6 SETUP EVENT 35

3.7 READ TAGS 37

4 WEB BROWSER INTERFACE 38

4.1 HOME PAGE 38

4.2 STATUS 39

4.3 USERS MANAGEMENT 40

4.3.1 Add User 41

4.3.2 Delete User 42

4.3.3 Modify Password 43

4.3.4 List Users 44

4.3.5 Set Auto Logout Time 45

4.3.6 Login/Logout 45

4.4 SYSTEM MANAGEMENT 46

Copyright © Convergence Systems Limited, All Rights Reserved

4.4.1 Reader ID *47*
4.4.2 Access Mode *49*
4.4.3 Frequency Configuration *51*
4.4.4 Operation Profile *56*
4.4.5 Memory Information *61*
4.4.6 Configuration Backup/Restore/Purge *62*
4.4.7 Failover Configuration *65*
4.4.8 Restart System *66*
4.5 NETWORK MANAGEMENT 67
4.5.1 Network Configuration *68*
4.5.2 Trusted Server *69*
4.6 TIME AND TIMER SETTING 73
4.6.1 Date/Time *74*
4.6.2 NTP Setup *75*
4.7 VERSION MANAGEMENT 76
4.8 CAPTURE POINT 79
4.9 TAG & TAG FILTER 80
4.9.1 Access Password *81*
4.9.2 Kill Password *84*
4.9.3 Kill Tags Testing *85*
4.9.4 Write Tags Testing *86*
4.9.5 Capture Tags Testing *90*
4.10 I/O MANAGEMENT 93
4.10.1 I/O Port Assignment *94*
4.10.2 List I/O Port Assignment *95*
4.10.3 I/O Port Testing *96*
4.10.4 Serial Port 1 Testing *98*
4.11 EVENT MANAGEMENT 99
4.11.1 Event *100*
4.11.2 Trigger *106*
4.11.3 Resultant Action *110*

5 PROGRAMMING INTERFACE 116

5.1 HIGH LEVEL API 117
5.1.1 HTTP Request Query *118*
5.1.2 XML Response *118*
5.1.3 TCP Notification *119*
5.1.4 Typical Program Flow *119*
5.1.5 Sample Usage Scenario – Access Control *122*

Copyright © Convergence Systems Limited, All Rights Reserved

5.1.6 Sample Usage Scenario – Conveyor Belt *131*
5.1.7 Sample Usage Scenario – Gambling *144*
5.2 LOW LEVEL API 155
5.2.1 Modem States *155*
5.2.2 Sample Usage Scenario – Start Inventory *156*

6 CSL DEMO PROGRAMS 158

6.1 HIGH LEVEL API DEMO PROGRAM 158
6.1.1 Installing Demo Program *158*
6.1.2 Using Demo Program *159*
6.1.2.1 Autonomous Time Trigger Mode *162*
6.1.2.2 Polling Trigger by Client Mode *167*
6.1.2.3 Save Read Tags *170*
6.2 LOW LEVEL API DEMO PROGRAM 171
6.2.1 Installing Demo Program *171*
6.2.2 Configuring Reader(s) *171*
6.2.3 Reading Tags *174*
6.2.4 Tag Reading Graph *176*

7 USAGE TIPS FOR CS461 178

7.1 INTRODUCTION 178
7.2 GENERAL TIPS 178
7.3 SYSTEM TIPS 178
7.4 WRITE TAG TIPS 178
7.5 EVENT ENGINE TIPS 179

8 RFID COOKBOOK 180

8.1 INTRODUCTION 180
8.2 APPLICATION DETAILS 183
8.2.1 Business Process Analysis *183*
8.2.2 Technology Selection *186*
8.2.3 Customer Expectation Management *187*
8.2.4 Hardware Configuration *188*
8.2.5 Software Configuration *189*
8.2.6 System Integration *191*
8.2.7 Pilot Test *192*
8.2.8 Optimization *194*
8.2.9 Customization *195*
8.2.10 Training *196*

Copyright © Convergence Systems Limited, All Rights Reserved

8.2.11 Test & Commissioning *197*
8.2.12 Maintenance & Statistics *198*
8.3 ANTENNAS FOR DIFFERENT BUSINESS APPLICATIONS 199

9 RFID BEST PRACTICES 200

9.1 INTRODUCTION 200
9.2 INTEGRATION PROCESS DETAILS 202
9.2.1 Familiarization Process *202*
9.2.1.1 Familiarizing with Browser Interface *202*
9.2.1.2 Familiarizing with Programming Interface *202*
9.2.1.3 Full Scale Programming and Integration *203*
9.2.1.4 Reader Capability Envelope Discovery *203*
9.2.2 Integration Process *205*
9.2.2.1 Use Cases and Requirements Gathering *205*
9.2.2.2 Draft Solution and In-House Testing *205*
9.2.2.3 API Programming *205*
9.2.2.4 Pilot Testing *205*
9.2.2.5 Middleware Testing *206*
9.2.2.6 Finalizing Solution *206*
9.2.2.7 Scaling *206*

10 RFID USE CASES 208

10.1 WAREHOUSE REAL TIME INVENTORY TRACKING 208
10.2 HIGH TRAFFIC HUMAN ACCESS CONTROL 209
10.3 REUSABLE PALLET TRACKING 210
10.4 WORK-IN-PROGRESS MONITORING 211
10.5 HUMAN ACCESS CONTROL BY AUTONOMOUS TAG GROUPS IN READER 212
10.6 PALLET/CARTON TAGGING VERIFICATION 213
10.7 BLOOD BAG TRACKING 214
10.8 PHARMACEUTICAL BOTTLES TRACKING AND ANTI-COUNTERFEIT 215
10.9 VEHICLE TRACKING IN MAINTENANCE DEPOT 216
10.10 VEHICLE INFORMATION SYSTEM 217
10.11 DOCUMENT TRACKING 218

11 TROUBLESHOOTING GUIDE 219

11.1 COMMON PROBLEMS AND POSSIBLE CAUSES 219
11.2 TROUBLESHOOTING PROCEDURE 221
11.2.1 Hardware *221*
11.2.1.1 Cannot Read Tag From Antenna *221*

Copyright © Convergence Systems Limited, All Rights Reserved

11.2.1.2 Short Read Range 225
11.2.1.3 No Read From Dense Readers 226
11.2.1.4 I/O Device Not Work 228
11.2.2 Web Browser Interface 230
11.2.2.1 Cannot Access Browser Interface 230
11.2.2.2 Health Check Failed 233
11.2.2.3 Write Tag Fail 233
11.2.3 Low Level API Demo Program 234
11.2.3.1 Cannot Connect to Reader 234
11.2.3.2 Cannot Read Tags 236
11.2.4 Programming Interface 239
11.2.4.1 getCaptureTagsRaw Cannot Get Newly Captured Tag 239
11.3 BUG REPORTING: FORMAT & INFORMATION REQUIRED 241
11.3.1 Prerequisite 241
11.3.2 Bug Reporting Procedure 242
APPENDIX A. RFID BASICS 274
APPENDIX B. GLOSSARY 275
APPENDIX C. API TABLE 281

Copyright © Convergence Systems Limited, All Rights Reserved

1 Introduction

1.1 How to Use this Manual

This manual provides a comprehensive introduction to the CSL CS-461 EPC Class1 Gen 2 RFID product (chapter 1), installation guide (chapter 2), quick start guide (chapter 3), web browser interface (chapter 4), CSL demo program (chapter 5), programming interface (chapter 6), usage consideration and recommendation (chapter 7), and troubleshooting guide (chapter 11). Some other information such as RFID application guide (chapter 8), RFID reader integration best practice (chapter 9) and RFID use case are also provided for reference.

In addition to this user's manual, there are other programmer's manuals for system integrators and software houses that develop their own software and would like to interface directly with this reader. Please refer to these manuals for the details of using the command sets.

There are two ways of accessing the reader, High Level access (HTTP-based) and Low Level access (TCP/IP socket based).

The High Level access method is described in the CSL High Level API Manual, and the Low Level access method is described in the CSL Low Level API Manual.

Copyright © Convergence Systems Limited, All Rights Reserved

1.2 Product Package

1.2.1 Basic Package Content

The reader package contains:

- Reader
- User Manual (in CD format)
- Power Adapter
- Power Cord
- Plastic Cover

1.2.2 Unpacking Instructions

Unpacking of the reader is very simple. The only caution is that the RF connector sockets should be handled with care. The TNC reverse RF connectors come covered with plastic cap. They should remain covered when not in use to reduce chance of ESD entering the ports via the center conductors.

Copyright © Convergence Systems Limited, All Rights Reserved

1.3 Product Specification

Figure 1-1 CS-461 Reader

Features:

- Certified to the EPCglobal Class 1 Gen 2 UHF RFID protocol including dense reader mode
- Sophisticated data handling for efficient management of large streams of tag data on LAN resources
- Highly configurable buffering and tag filtering modes to eliminate the redundant tag data so as to reduce LAN traffic and server loading
- Compliant to the ISO 18000-6 type-C UHF RFID Standard
- 640 kbps tag-to-reader data rates
- Robust performance in dense-reader environments
- Excellent in transmit and receive mode – generates a different combination of unique reader-to-tag command rate, tag-to-reader backscatter rate, modulation format, and backscatter type
- Tremendous savings by using a single transmit/receive antenna for each of its four points ultra high inventory rate, read rate and tag velocity
- Settable and configurable parameters offer maximum throughput and optimal performance
- Supports all Gen 2 commands, including write, lock and kill

Copyright © Convergence Systems Limited, All Rights Reserved

Specifications:

Physical Characteristics:	Length: 29.5 cm; Width: 30 cm; Height: 8 cm; Weight: 3 Kg
Mounting	Vertical orientation
Environment:	Operating Temp: -20^{0}C to 55^{0}C Storage Temp: -40^{0}C to 85^{0}C Humidity: 10% to 95% non-condensing Enclosure: IP-53
Antenna:	4 TNC duplex antenna ports, each single unit antenna for transmit and receive per port
Power:	Power adaptor for 110-240VAC auto-ranging to DC24Volts, 60Watt
RFID Frequency Ranges:	800 or 900 MHz band
Interfaces	10/100 BASE-T Ethernet RJ45 connector Configurable to use fixed IP address or DHCP RS-232 (DB9 connector) HTTP web server Tag air interface: EPC Class 1 Gen 2
Networking Protocols:	High Level: CSL High Level API Low Level: CSL Low Level API
Hardware Platform:	Xscale
Operating System:	Monta Vista Linux
Maximum Tag Read Rate:	1000 tag/sec.
Maximum Speed of Tag:	660 ft/min
Accessories:	Power cord
Order Code:	CS-461-P (P=1: 865-869MHz; P=2: 902-929MHz; P=3: 950-956MHz)
Restrictions on Use:	Approvals, features and parameters may vary depending on country legislation and may change without notice

Copyright © Convergence Systems Limited, All Rights Reserved

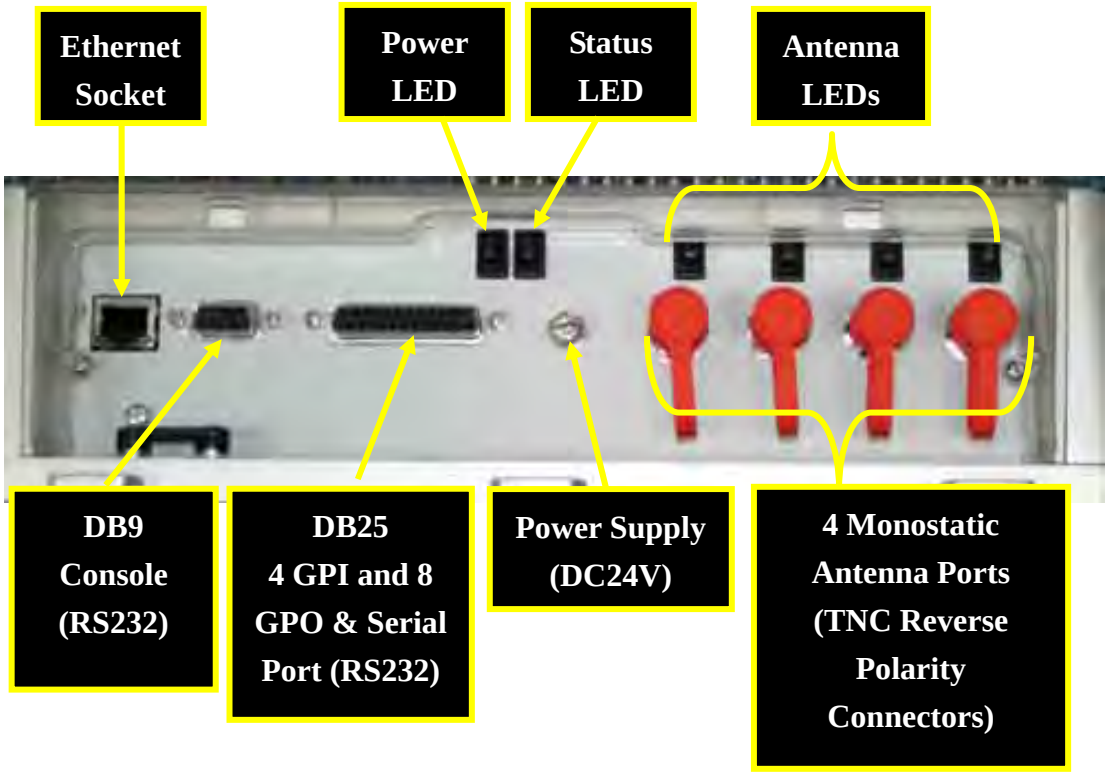
2 Installation

2.1 Devices

2.1.1 Reader Connection

The CSL CS-461 RFID Reader is a EPCglobal Class 1 Gen 2 certified fixed reader product. This reader is powered by Impinj technology, with extremely high inventory rate, tag velocity and true dense reader mode.

This reader can connect to and control four UHF antennas from its 4 TNC duplex antenna ports.

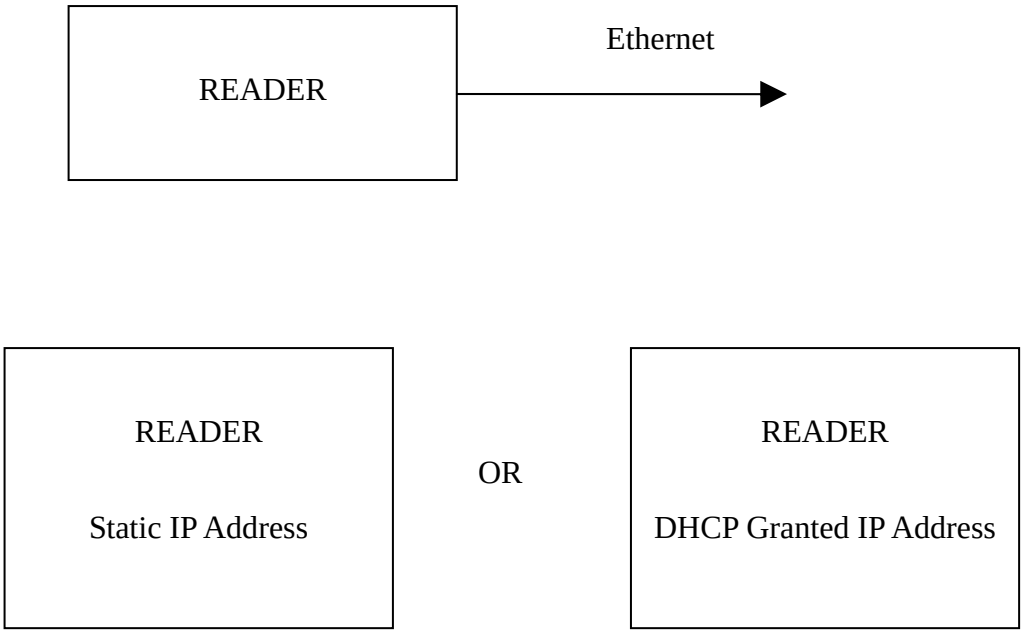


Note: Minimum cable length is 1.5m for this reader.

The reader is connected to the network via Ethernet cable (RJ45 socket). The reader can have a static IP address or can obtain an IP address using DHCP. Normally, a static IP address is

Copyright © Convergence Systems Limited, All Rights Reserved

more convenient to use because it does not change when the reader reboots, but the user has to make sure there is no collision with other network devices in the network. If the reader is configured to be DHCP, then a separate discovery program that runs on the PC side can help the user find all readers in the same local area network.



Copyright © Convergence Systems Limited, All Rights Reserved

2.1.2 Antenna Cable Connection

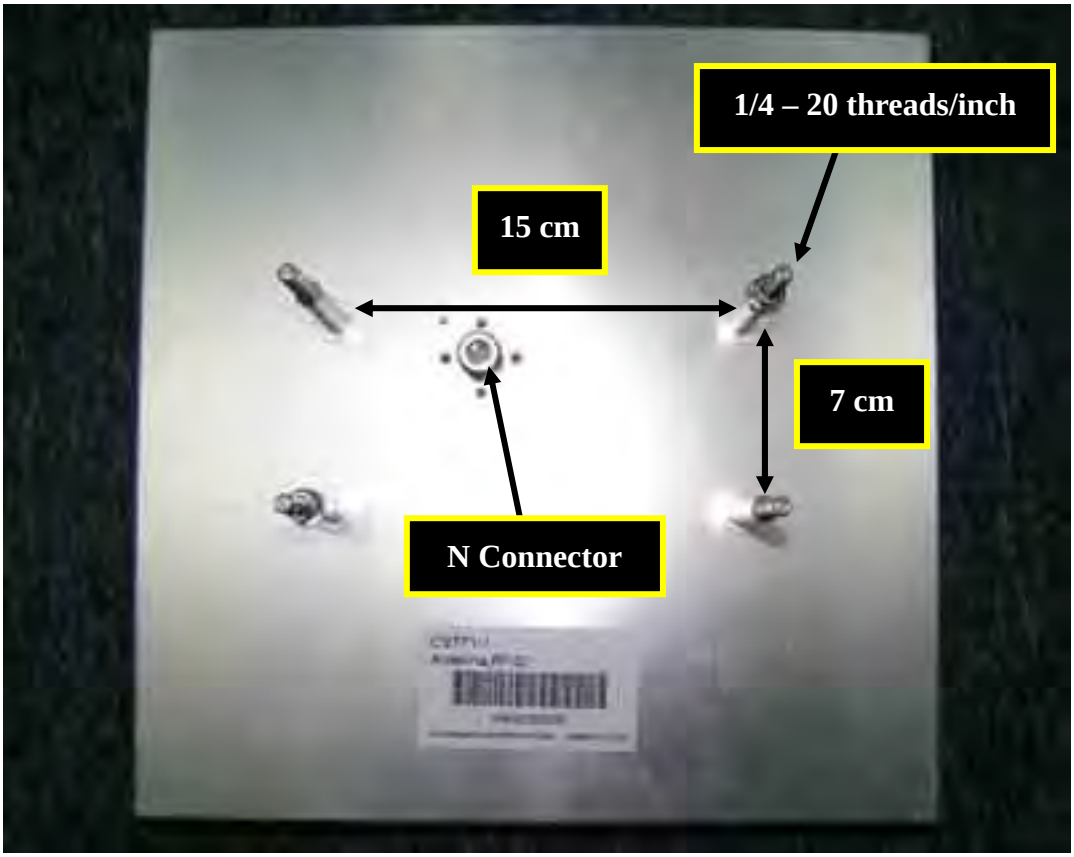


Figure 2-1 CS-771 Mono-Static Antenna

Copyright © Convergence Systems Limited, All Rights Reserved

2.2 Installation Steps

The reader can be setup easily as described below:

1. Connect the antenna(s) to the reader using the appropriate antenna cables (TNC reverse male connector to the reader side and N straight male connector to the antenna side).

2. Connect the reader to your network or computer using LAN cable on the LAN port. Please remember to use cross-over cable if it is direct reader-to-computer connection.

3. Plug in the power cord to the reader and switch on the power supply. Then the reader will boot up automatically. After the LEDs on the reader finished flashing and the Power LED remains in ON state, the reader has been boot up successfully. You can now use the web-based administration page of the reader to configure the reader.

   - In order to access the web-based administration interface of the reader, open a web browser (i.e. Internet Explorer, IE) on your PC and enter the IP address of the reader on the URL field (the default IP address of the reader is printed on the label of the reader). Make sure that the PC is configured in the same subnet as the reader and they are properly connected on the LAN.
   - If a blank page is displayed after entering the web interface, please install the Microsoft XML Core Services (MSXML 4.0 Service Pack 2) and try again. It can be downloaded from:

   http://www.microsoft.com/downloads/details.aspx?FamilyID=3144b72b-b4f2-46da-b4b6-c5d7485f2b42&DisplayLang=en

4. After that, the web-based administration page of the reader will be displayed on the web browser as follows:

Copyright © Convergence Systems Limited, All Rights Reserved

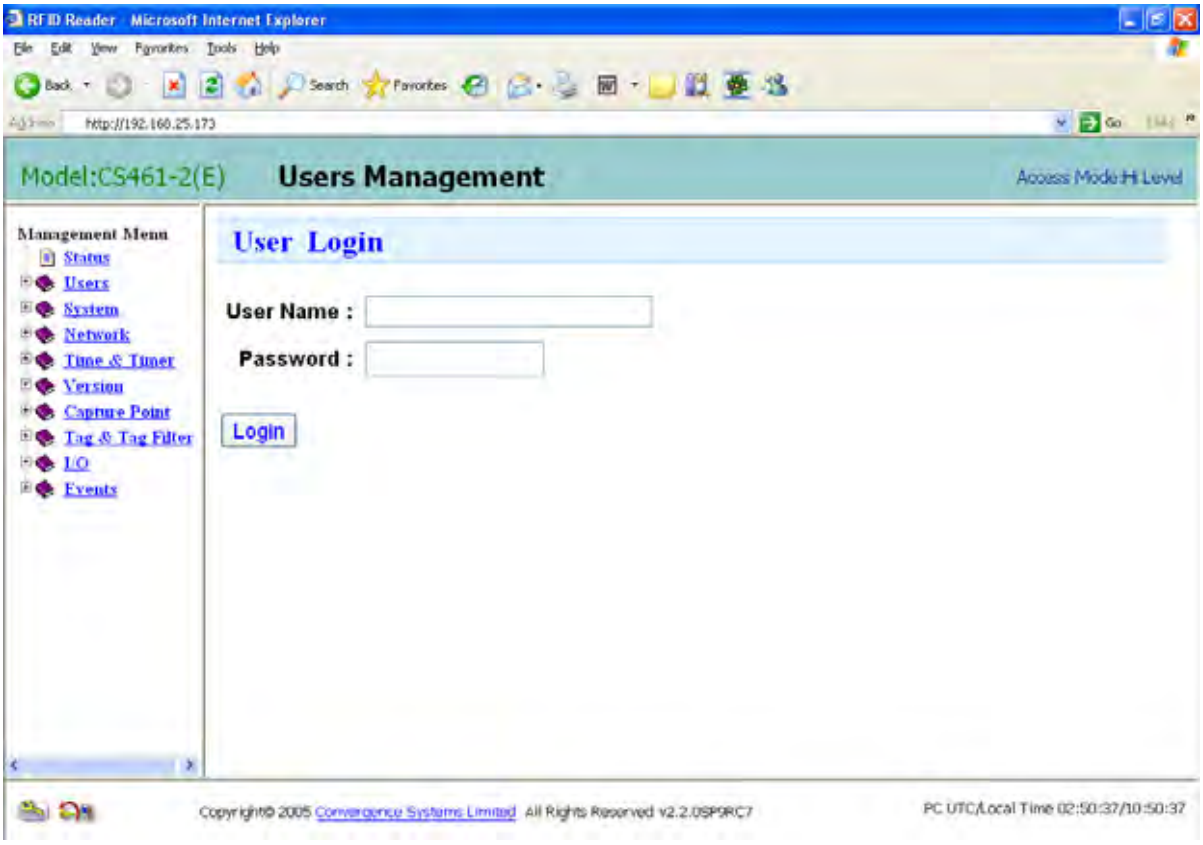


Figure 2-2 User Login

You can now login the configuration tool. The default administrator login name and password are as follows:

Login: root
Password: csl2006

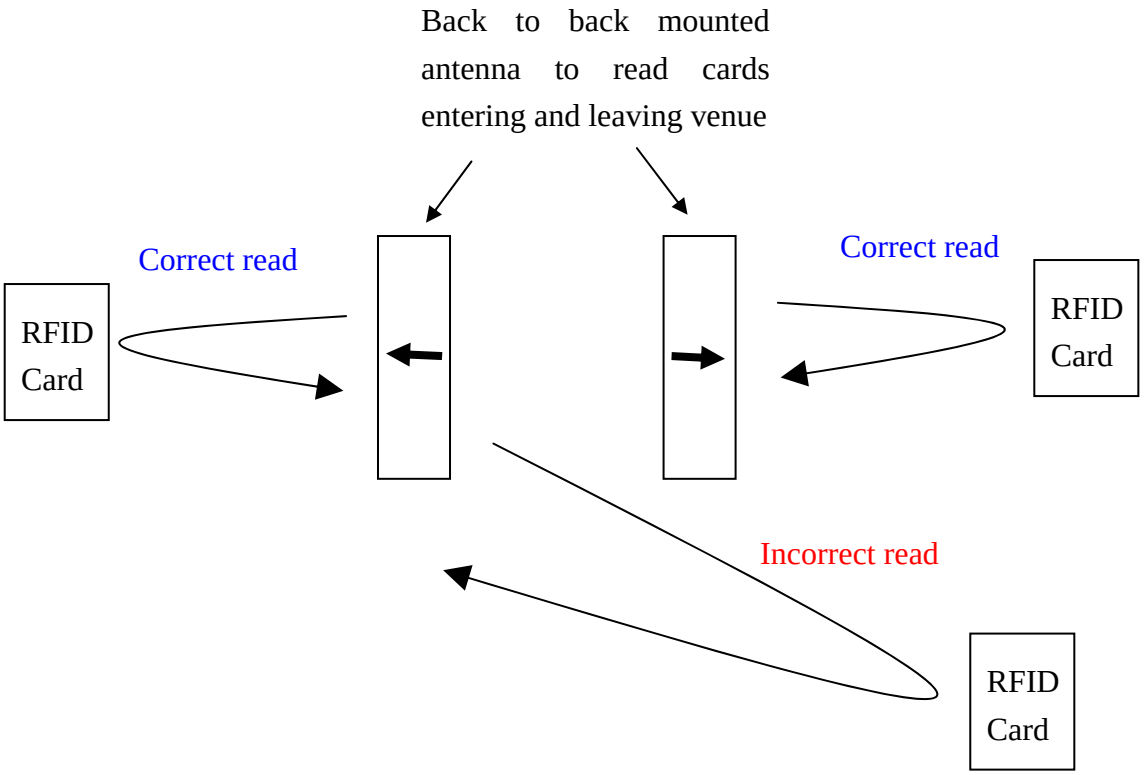
Please refer to chapter 4 for the details of this configuration interface.

5. In addition to the web interface, a “Reader Demo Program” is also provided for your configuration and testing of the reader. Please refer to chapter 錯誤! 找不到參照來源。 for the details.

2.3 Installation Recommendation

2.3.1 Antenna Installation

In antenna installation, especially when the reader is connected to multiple antennas, then the most important and immediate concern is spatial coupling between antenna, or, in other words, the isolation between antennas. If you mount antenna back to back, it is possible that the backlobe of the antenna will be able to transmit enough of the energy to turn on a tag that are on the opposite side and should have only been picked up by the antenna on that opposite side.

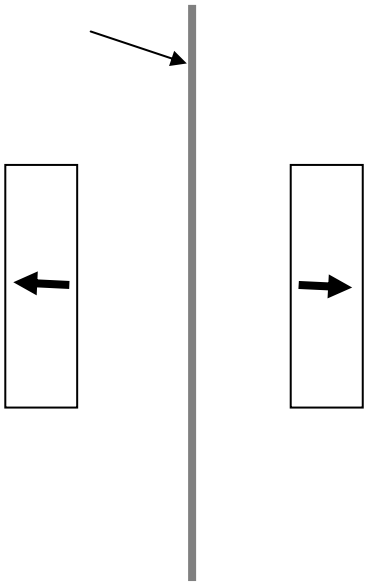


Back to back mounted antenna may erroneously read cards from back due to antenna backlobe

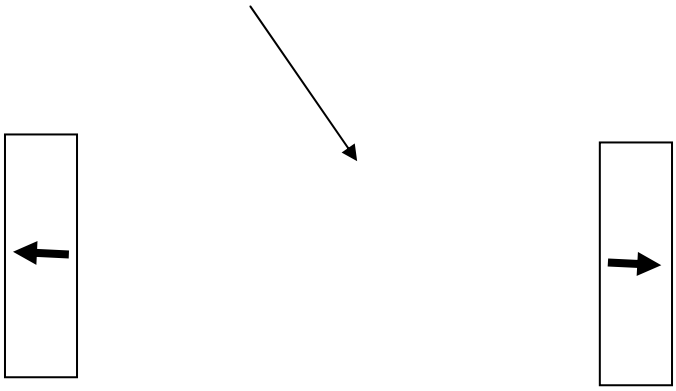
Copyright © Convergence Systems Limited, All Rights Reserved

To prevent this erroneous read from happening, one can put spacer metal plate between the two back to back mounted antenna, or place the antenna farther apart, or a combination of the two methods:

Metal plate in between to isolate the two antennas, each dimension should be at least three times that of antenna



Increase the separation between the two antennas will increase the isolation between them.



Copyright © Convergence Systems Limited, All Rights Reserved

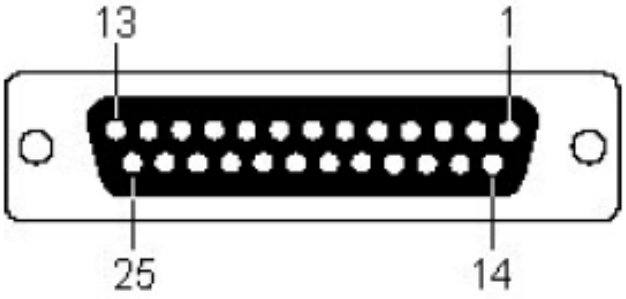
Note also that the isolation between the different ports on the readers is not all the same. There are pairs with better isolation than others. Port 1 has the best isolation with port 3; port 2 has the best isolation with port 4. If isolation between antennas is found to be a problem, put the problem antennas at these best isolated pairs, i.e. connect one to port 1 and the other to port 3, or connect one to port 2 and the other to port 4.

Copyright © Convergence Systems Limited, All Rights Reserved

2.3.2 IO Connection

The IO connector consists of one DB25 connector. The IO pins are 3.3V CMOS signal (i.e., 0V for logic 0 and 3.3V for logic 1). Since the current output from the port is low (please see below for details), users are recommended to connect minimal external circuitry for driving external device. In any case, it is always a good idea to have an external adaptor board to protect the reader from directly affected by external world environmental conditions, such as lightning. The following are the pin-out definition:

Pin Assignment



Front view (female) of CS461

Pin	Function
1	Not Connect
2	RXD input, RS232
3	TXD output, RS232
4	CTS input, RS232
5	RTS output, RS232
6	Not Connect
7	Ground
8	Not Connect
9	Not Connect
10	Data Input 4
11	Data Input 3
12	Data Input 2
13	Data Input 1
14	Data Output 1
15	Data Output 2
16	Data Output 3

Copyright © Convergence Systems Limited, All Rights Reserved

Pin	Function
17	Data Output 4
18	Data Output 5
19	Data Output 6
20	Not Connect
21	Data Output 7
22	Not Connect
23	Data Output 8
24	Not Connect
25	Not Connect

Electrical Specification

a) Output ports are internally driven by 74LVC273 (V_{cc} = 3.0V) with 100Ω current limiting resistors. (See Figure 2-3)

DC characteristic of 74LVC273:

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
V_{OH}	HIGH level output voltage	V_{cc} = 3.0V; V_I = V_{IH} or V_{IL}; I_o = -100 μA	V_{cc} – 0.2	V_{cc}		V
		V_{cc} = 3.0V; V_I = V_{IH} or V_{IL}; I_o = -12mA	V_{cc} – 0.6			
		V_{cc} = 3.0V; V_I = V_{IH} or V_{IL}; I_o = -24mA	V_{cc} – 1.0			
V_{OL}	LOW level output voltage	V_{cc} = 3.0V; V_I = V_{IH} or V_{IL}; I_o = 100 μA			0.20	V
		V_{cc} = 3.0V; V_I = V_{IH} or V_{IL}; I_o = 24mA			0.55	

NOTE:
1. All typical values are at V_{cc} = 3.3V and T_{amb} = 25°C.

b) Input ports are internally connected to inputs of 74LVC244 via 100Ω resistors. (See Figure 2-5)

DC characteristic of 74LVC244:

Copyright © Convergence Systems Limited, All Rights Reserved

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			MIN	TYP[1]	MAX	
V_{IH}	HIGH level input voltage	V_{cc} = 2.7 to 3.6V	2.0			V
V_{IL}	LOW level input voltage	V_{cc} = 2.7 to 3.6V			0.8	V
V_I	DC Input voltage range		0		5.5	V
I_I	Input leakage current	V_{cc} = 3.6V; V_I = 5.5V or GND		±0.1	±5	μA

NOTE:

1. All typical values are at V_{cc} = 3.3V and T_{amb} = 25°C.

Example Circuits

Please refer to the following figures for example circuits.

a) Output port example with optical coupler

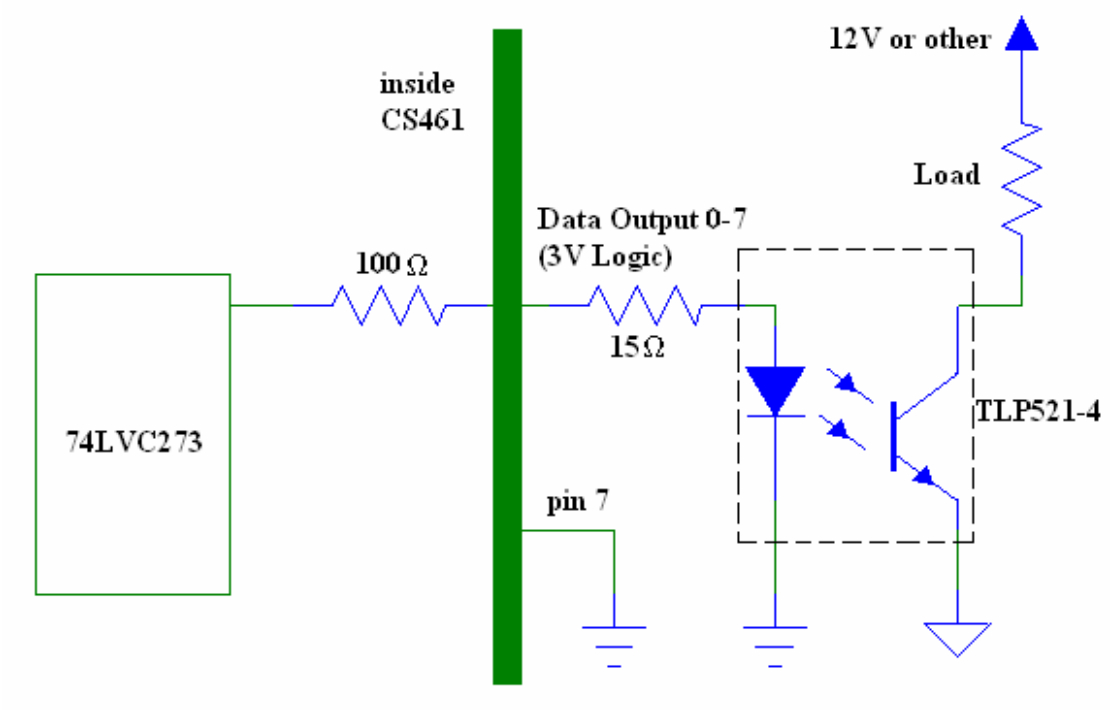


Figure 2-3

Copyright © Convergence Systems Limited, All Rights Reserved

b) Output port example with transistor

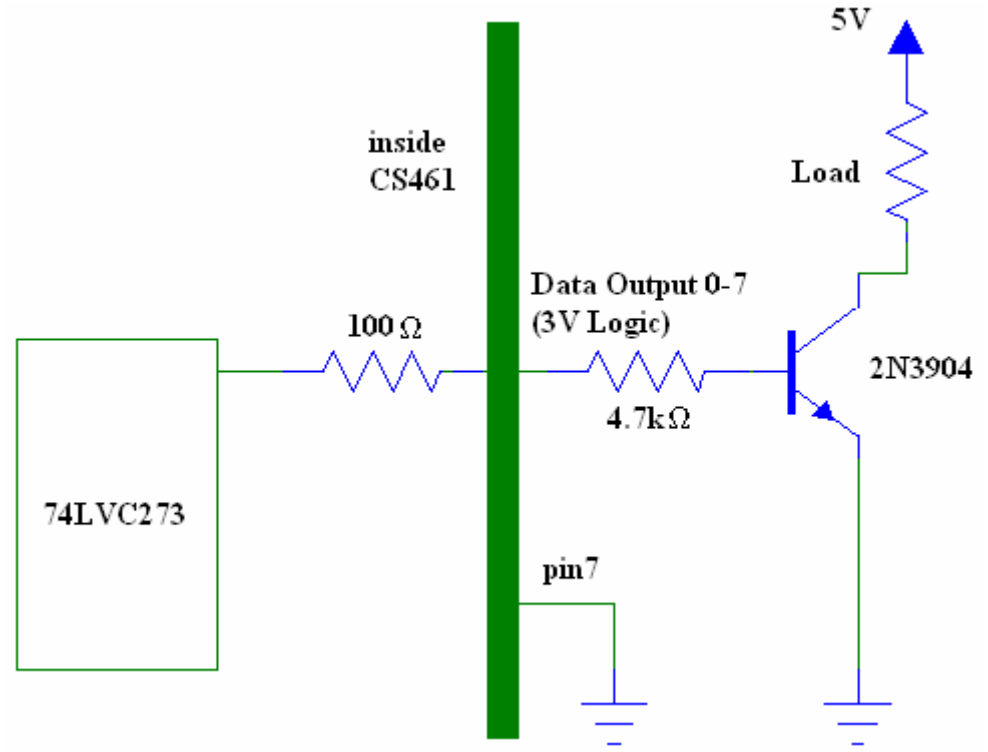


Figure 2-4

c) Input port example

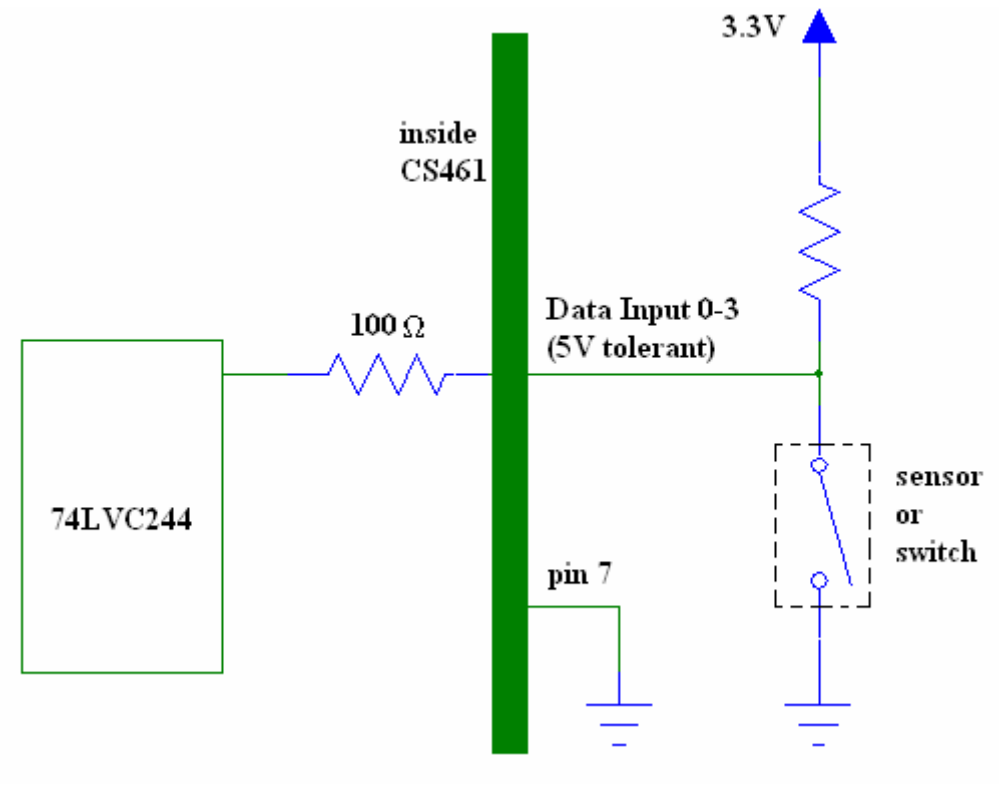


Figure 2-5

Copyright © Convergence Systems Limited, All Rights Reserved

d) Input port example with optical coupler

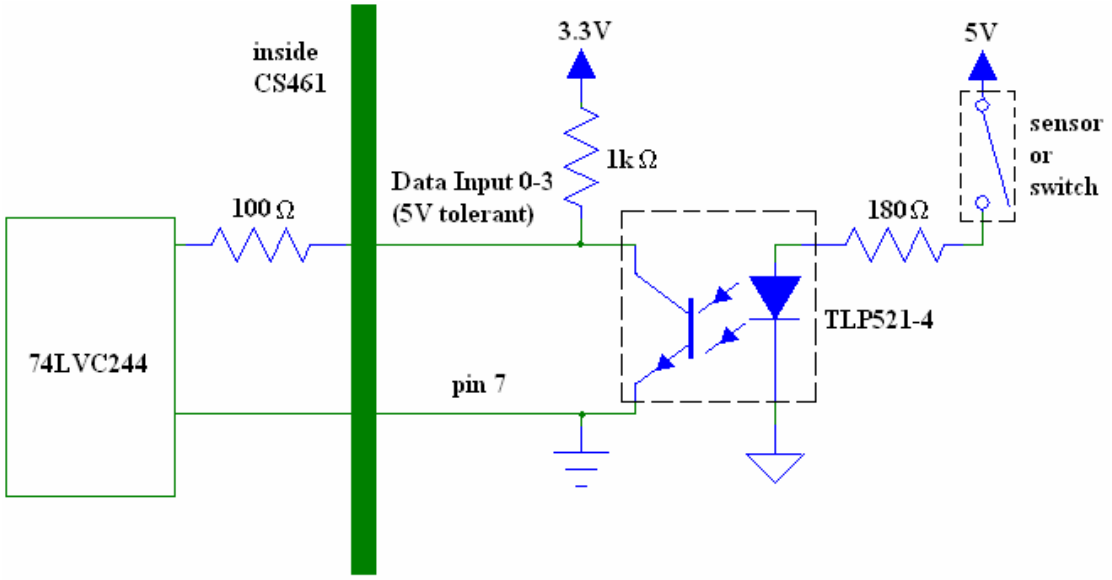


Figure 2-6

Adapter Board Schematics

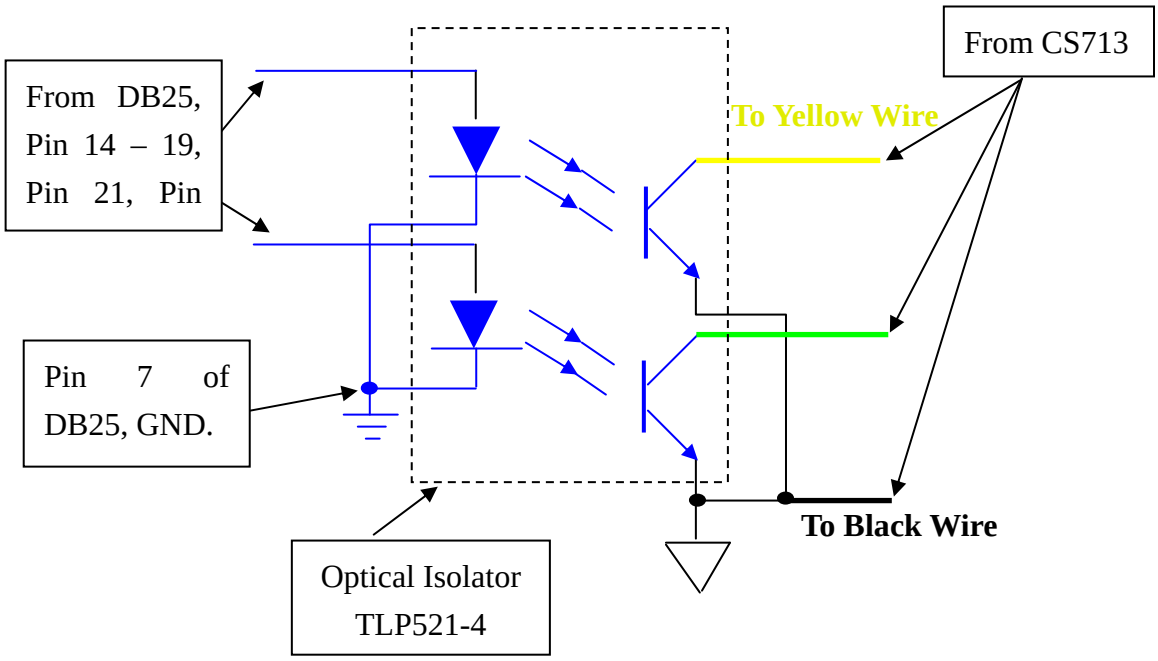


Figure 2-7

Copyright © Convergence Systems Limited, All Rights Reserved

2.4 Verification and Validation

The reader comes with a pre-configured event enabled to read tags from all 4 ports. So once the reader is taken out of the box, just by connecting any one of the antenna ports, and putting some tags in front of the antenna, and then going to the Capture Tag Testing page, one should be able to observe tags coming in. Please follow the steps below to verify the reader is functioning properly:

1. Hook up all four ports to four antennas.

2. Set up the reader to read all four ports alternately by modifying the Trigger in each try:

Go to the "Modify Trigger" page by clicking "Events -> Trigger -> Modify Trigger":

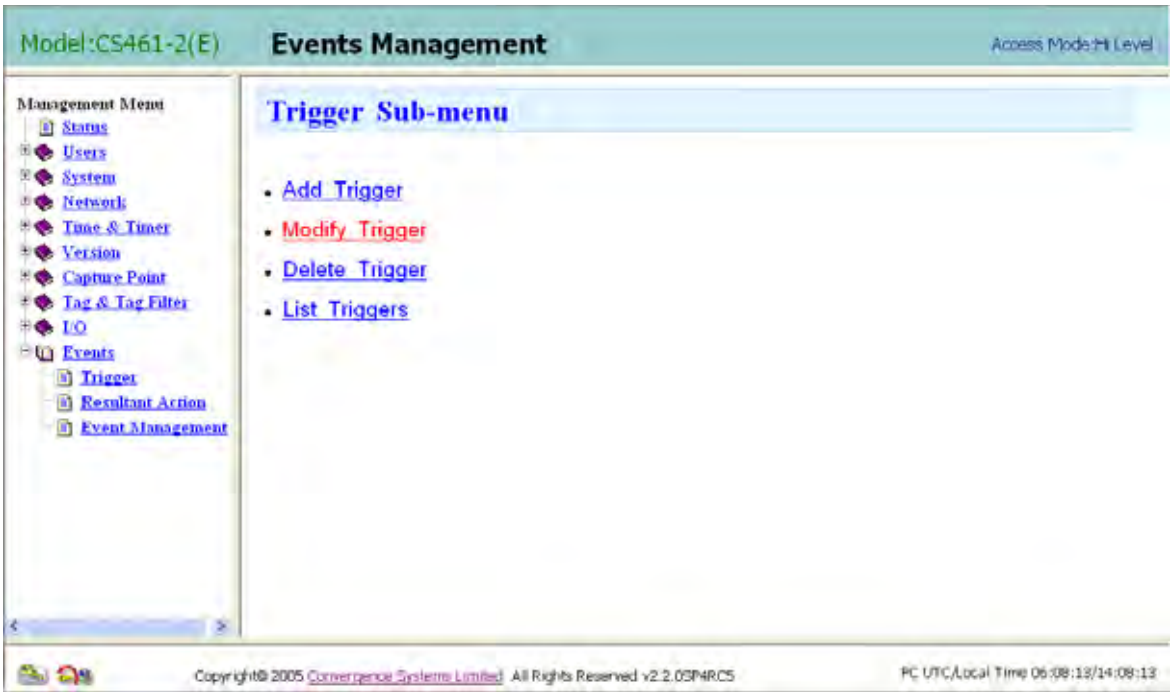


Figure 2-8 Trigger

Select one antenna alternately in each try and save the modification by clicking "Modify":

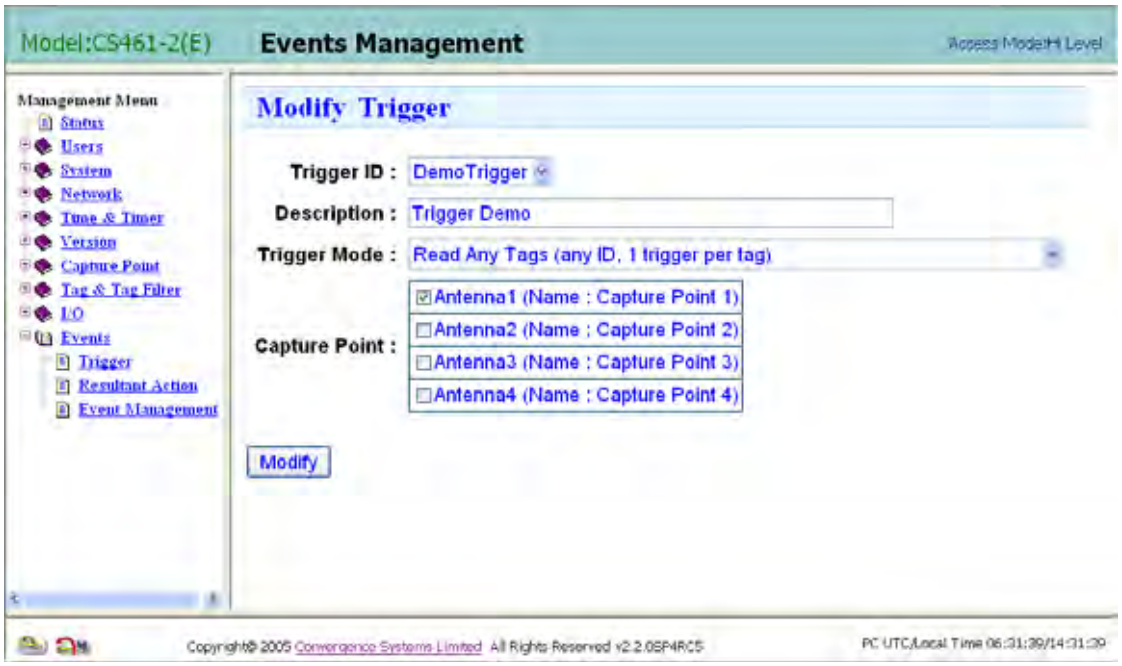


Figure 2-9 Trigger - Modify

3. Take the sample tags and read them from the antenna selected in the previous step. Make sure there is no other reader operating nearby. Verify the tags are read in the "Capture Tags Testing" page by clicking "Tag & Tag Filter -> Capture Tags Testing -> Capture Tags (*Time Window Mode Event Driven*) – EPC":

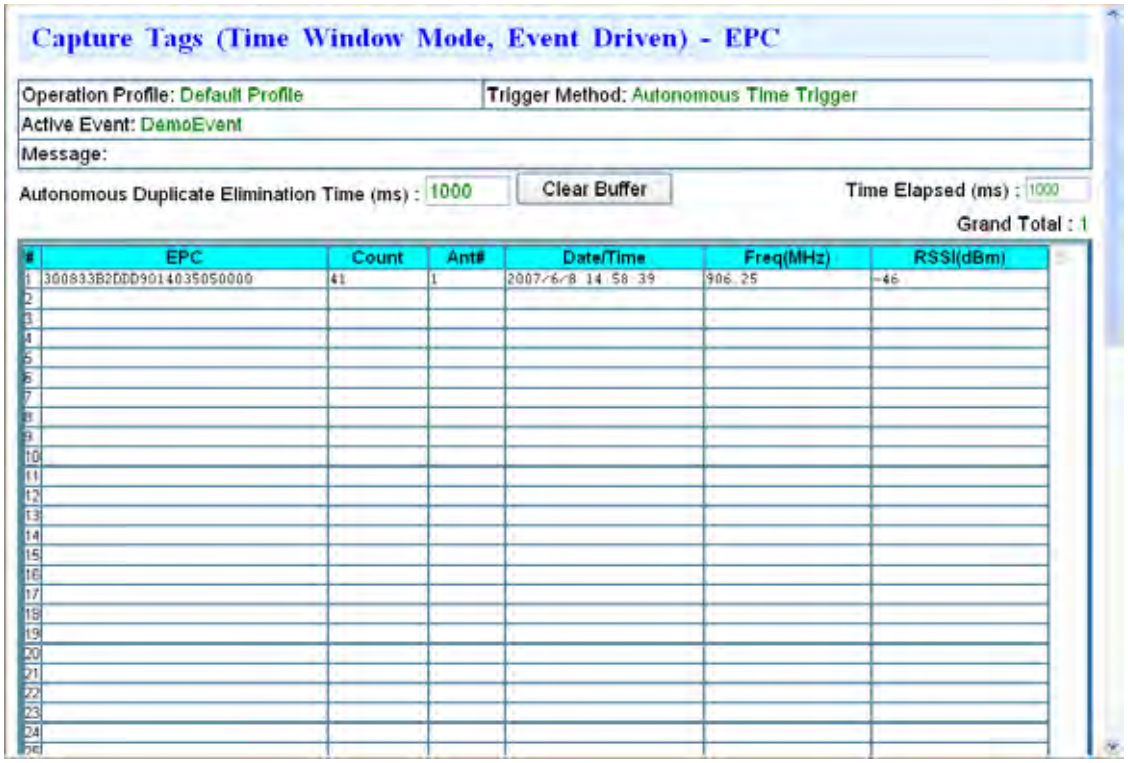


Figure 2-10 Capture Tags

4. Take the read range of the tags and check them against the standard performance.

Copyright © Convergence Systems Limited, All Rights Reserved

2.5 Cautions

The reader default IP address is printed on the reader label. To change this IP address, installer must connect the reader to a PC and modify the IP address using the web browser interface.

Copyright © Convergence Systems Limited, All Rights Reserved

3 Quick Start

3.1 Reader Login

- Power up the reader.
- Open an Internet Explorer and browse the reader using its IP Address (e.g. http://192.168.25.173). The browser screen should look like Figure 3-1.
- The version of current running firmware is shown at bottom. The firmware version in Figure 3-1 is v2.2.0.
- To login, input the “User Name” and “Password”, then click the “Login” button. The default administrator login name and password are as follows:

  Login: root

  Password: csl2006

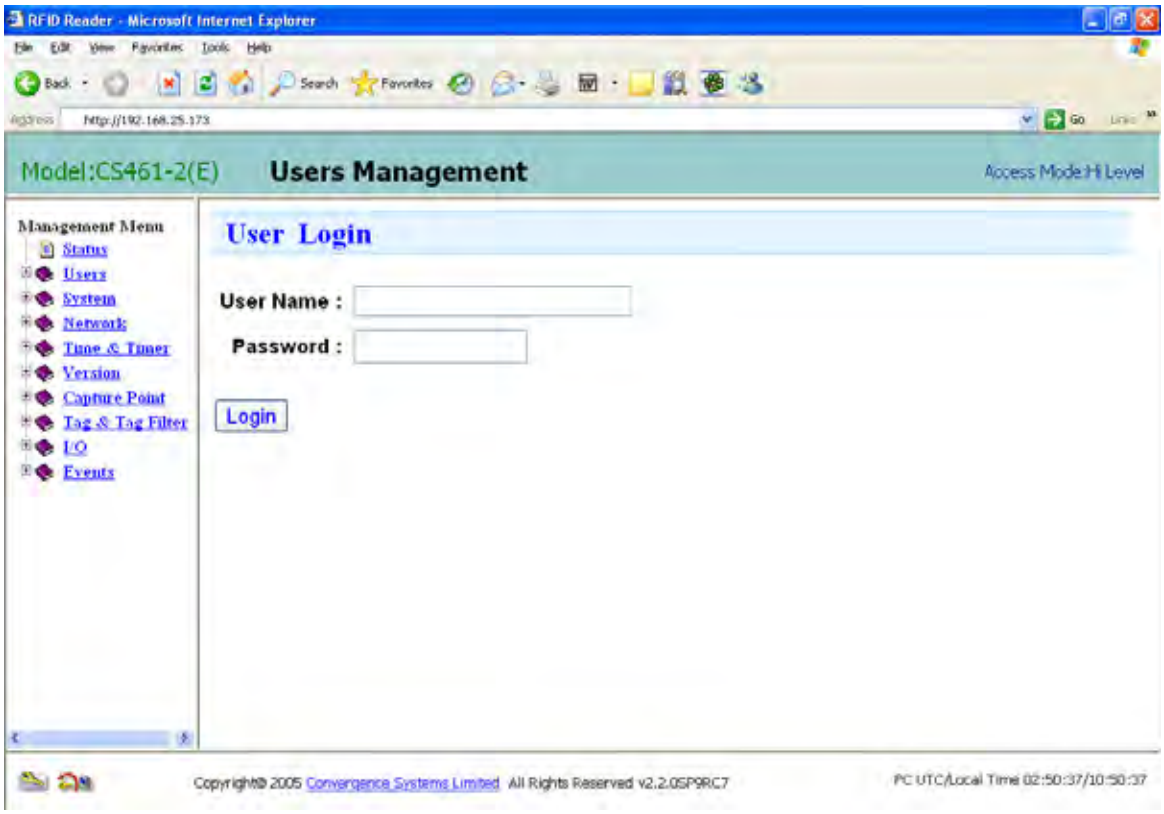


Figure 3-1 Login Screen

3.2 Firmware Version Upgrade

In case you want to make sure the firmware version is the latest possible, please do the following actions.

- Get firmware files from our ftp site. You can contact us for login details.

Item	Filename	Remark
1	lib1-2.0.4-461.0-3AE09359.cne	For firmware version before v2.0.7
2	lib2-2.0.4-461.0-01832089.cne	For firmware version before v2.0.7
3	patch-2.0.4-461.0-5FA8A972.cne	For firmware version before v2.0.7
4	reader-2.1.1-461.0-5BE0558B.cne	

- Please open page “Firmware Upgrade” as shown in Figure 3-2. You can reach the page by clicking “Version -> Firmware Upgrade”.
- Enter full path of upgrade file in “Firmware file location” entry using “Browse…” button. Then, click “Firmware Upgrade” button to send file to reader. Please repeat this action until all necessary files are upgraded.
- Restart the reader.

Figure 3-2 Firmware Upgrade page

3.3 Setup Access Mode

If access mode showing at the top of the screen is not “Hi Level”, please do the following actions.

- Please open page “Set Access Mode” as shown in Figure 3-3. You can reach the page by clicking “System -> Access Mode -> Set Access Mode”.
- Select “High Level HTTP API Mode” and click “Set” Button.

Figure 3-3 Set Access Mode

3.4 Setup Operation Profile

Item	Parameter	Value
1	Modulation Profile	Profile0
2	Population Est	50
3	Session #	3
4	Estimated Tag Time in Field	1000
5	Capture Mode Prefilter	Time window
6	Duplicate Elimination Triggering Method	Autonomous Time Trigger
7	Autonomous Duplicate Elimination Time	2000
8	Antenna1	Checked, 30 dBm
9	Antenna2	Checked, 22 dBm
10	Antenna3	Checked, 20 dBm
11	Antenna4	Checked, 15 dBm
12	Enable	Checked

Please set above parameters to default operation profile.

- Please open page “Operation Profile” as shown in Figure 3-4. You can reach the page by clicking “System -> Operation Profile”.
- Select correct values and then click “Set” button.

Copyright © Convergence Systems Limited, All Rights Reserved

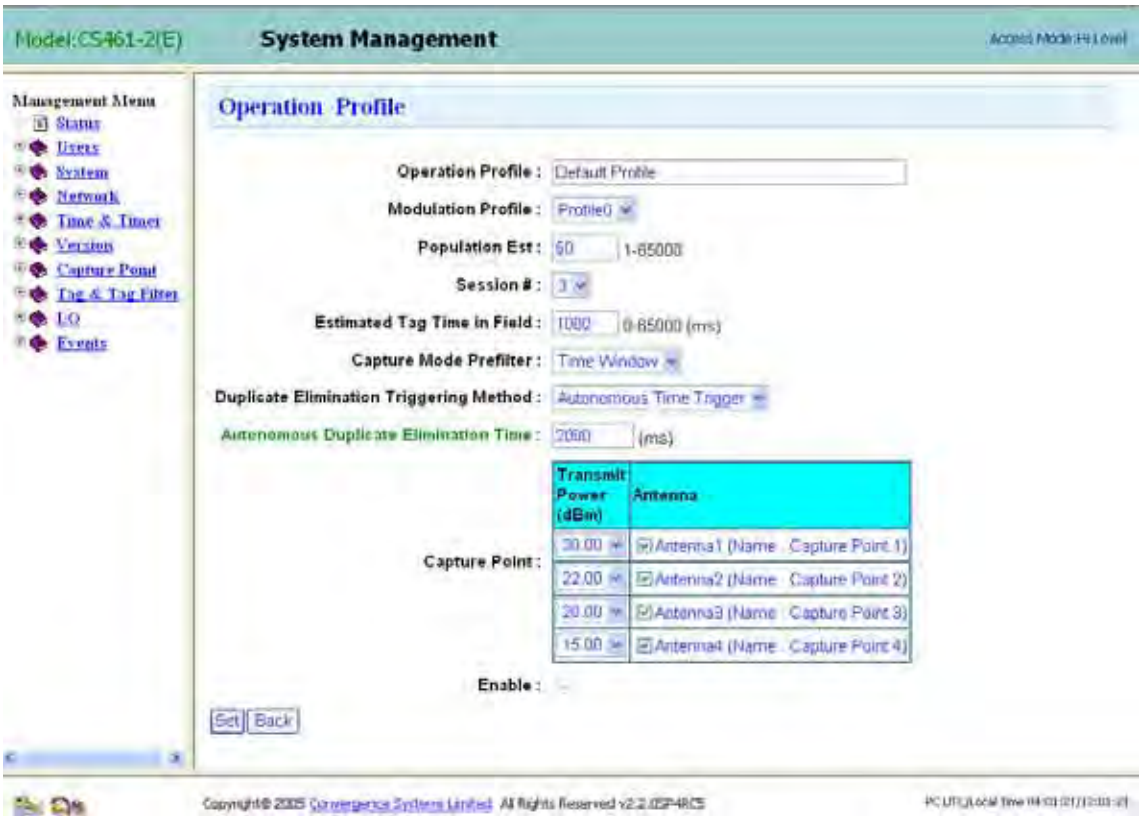


Figure 3-4 Operation Profile

3.5 Setup Trigger

Item	Parameter	Value
1	Trigger ID	DemoTrigger
2	Description	Trigger Demo
3	Trigger Mode	Read Any Tags (any ID, 1 trigger per tag)
4	Capture Point	Antenna1 to Antenna 4 are checked

Please verify the settings of DemoTrigger are identical to Figure 3-5. Otherwise, please modify the settings of DemoTrigger.

- Please open page “Modify Trigger” as shown in Figure 3-6. You can reach the page by clicking “Events -> Trigger -> Modify Trigger”
- Fill in correct values and click “Modify” button.

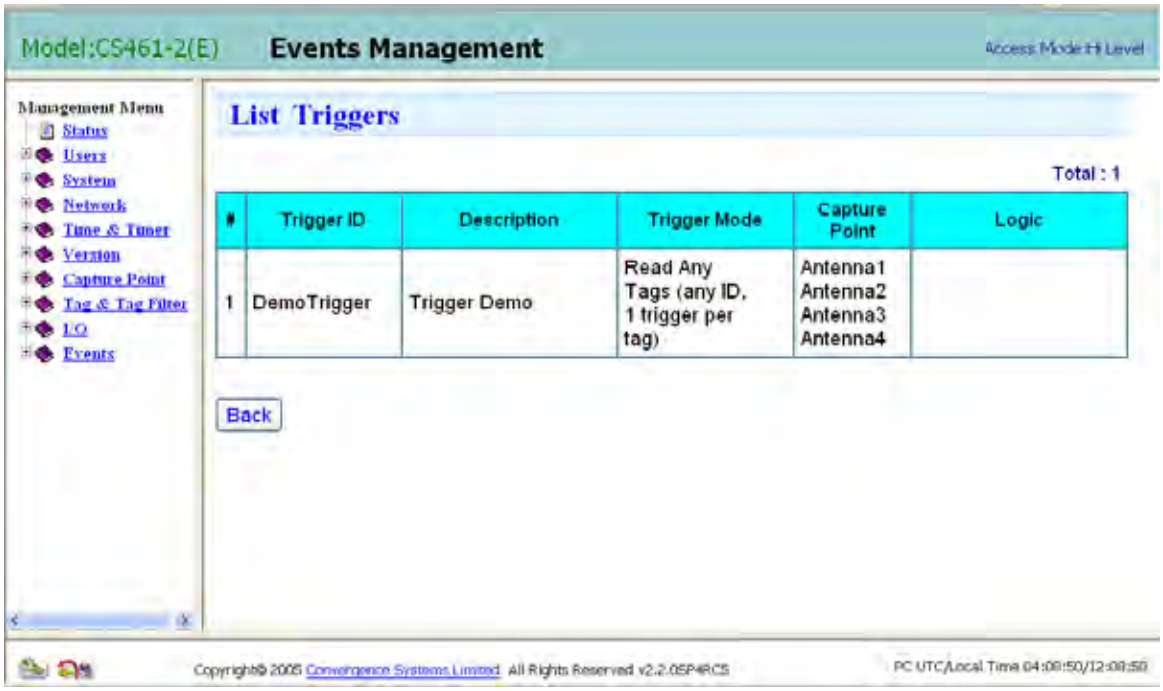


Figure 3-5 List Triggers

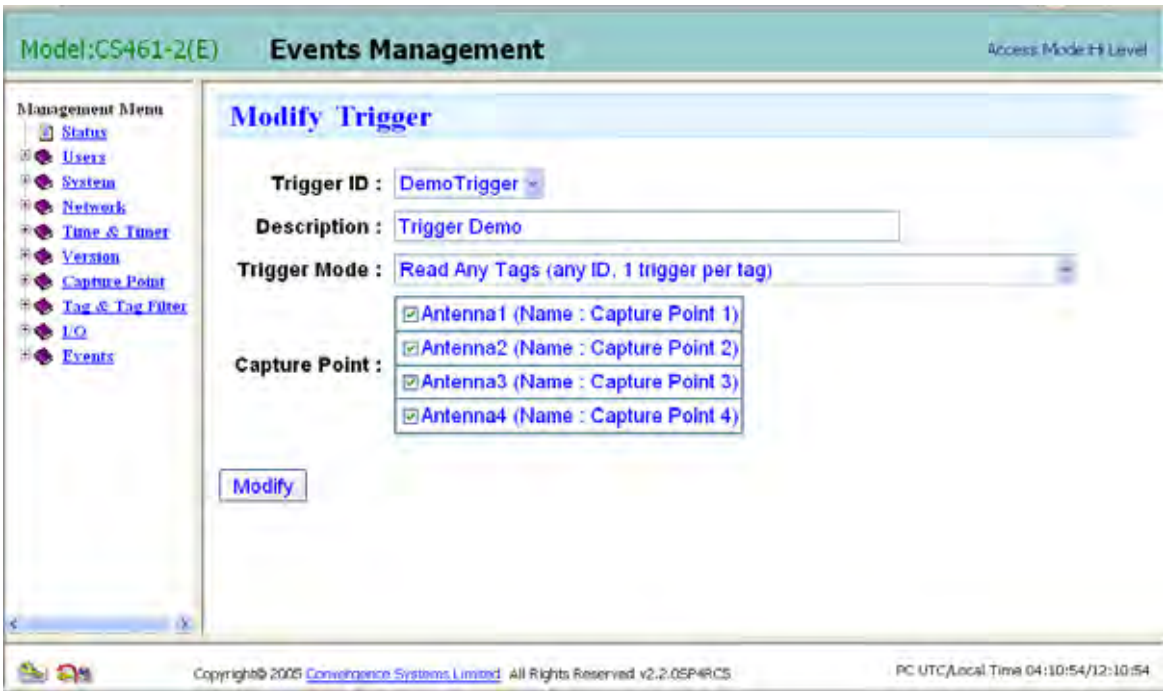


Figure 3-6 Modify Trigger

3.6 Setup Event

Item	Parameter	Value
1	Event ID	DemoEvent
2	Description	Event Demo
3	Inventory Enabling Trigger	Always On
4	Trigger Logic	DemoTrigger
5	Resultant Action	DemoAction
6	Inventory Disabling Trigger	Never Stop
7	Enable	Checked

Please verify the settings of DemoEvent are identical to Figure 3-7. Otherwise, please modify the settings of DemoEvent.

- Please open page “Modify Event” as shown in Figure 3-8. You can reach the page by clicking “Events -> Event Management -> Modify Event”
- Fill in correct values and click “Confirm Modification” button.

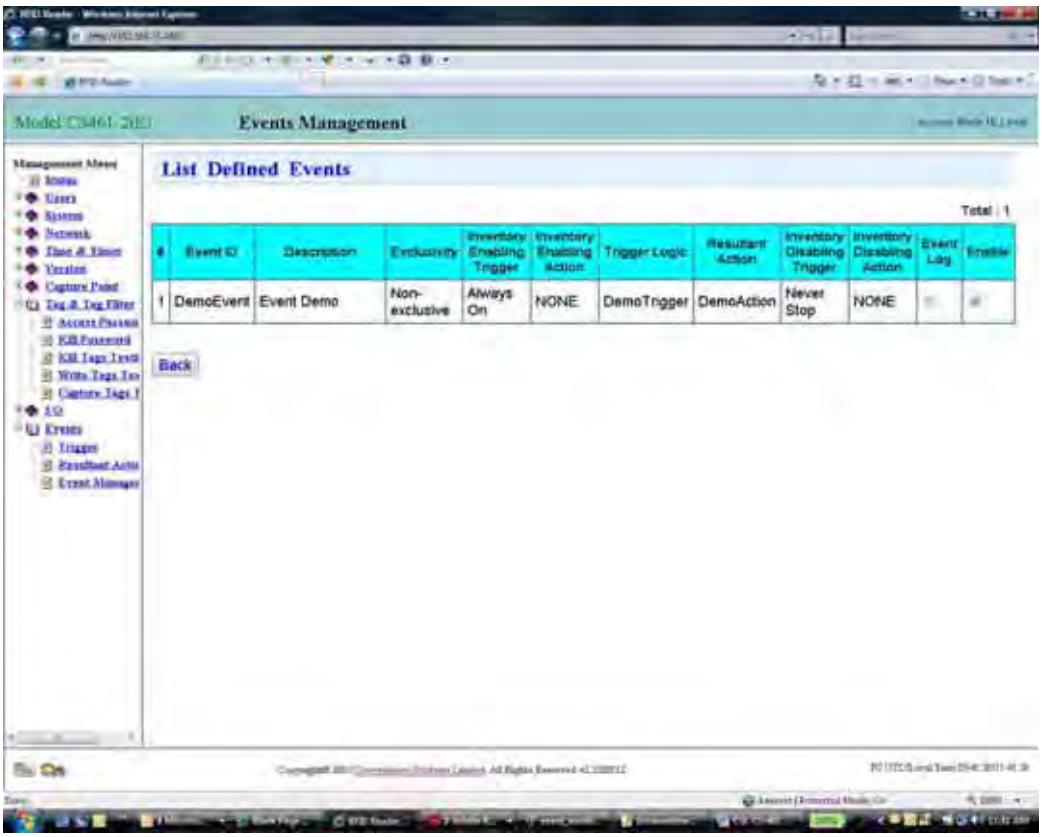


Figure 3-7 List Defined Events

Copyright © Convergence Systems Limited, All Rights Reserved

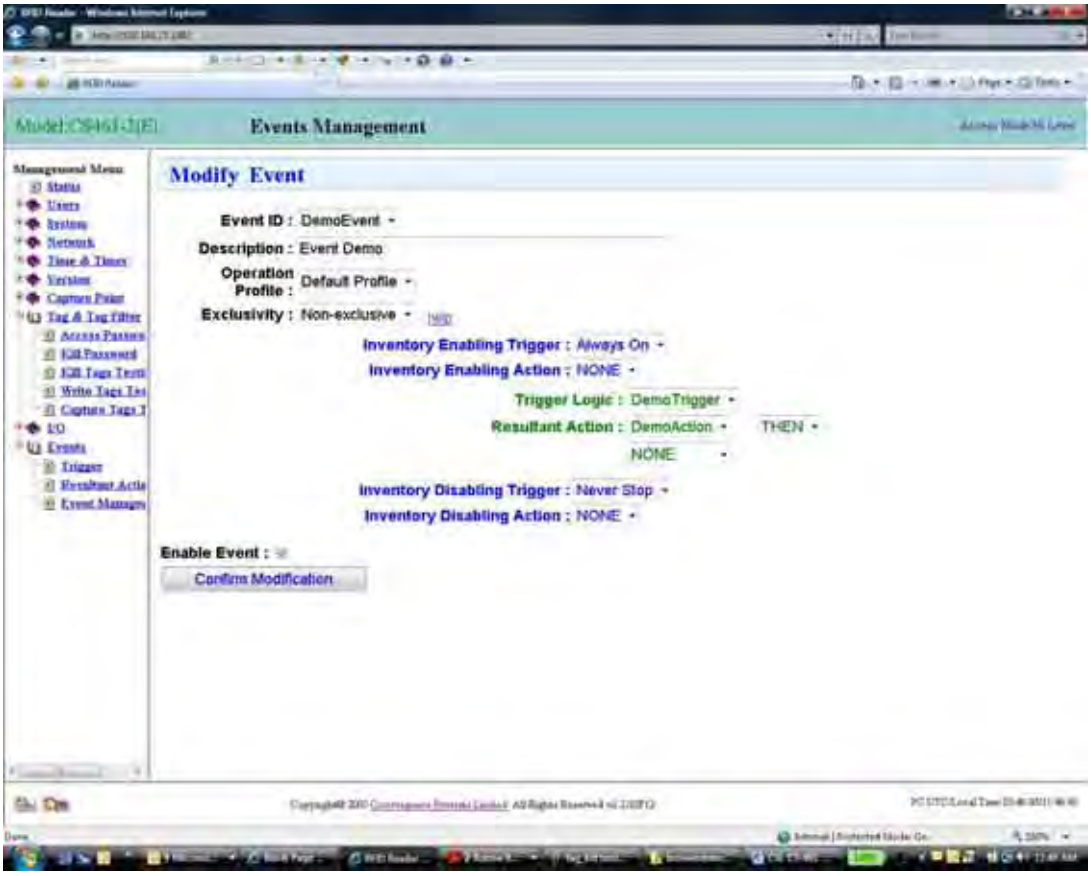


Figure 3-8 Modify Event

3.7 Read Tags

- Please connect antennas to reader. Then open page “Capture Tags (Time Window Mode, Event Driven) - EPC”. You can reach the page by clicking “Tag & Tag Filter -> Capture Tags Testing -> Capture Tags (Time Window Mode, Event Driven) - EPC”.
- Received tags information will be shown in a table as shown in Figure 3-9. The background of row will change to red color gradually if the corresponding tag has left the field.

Figure 3-9 Capture Tags (Time Window Mode, Event Driven) - EPC

Copyright © Convergence Systems Limited, All Rights Reserved

4 Web Browser Interface

4.1 Home Page

The home page of the web-based administration interface can be entered by just entering the IP address of the reader (default IP address is printed on the label) on the web browser (Internet Explorer is required).

For example, if the IP address of the reader is 192.168.25.173, you should enter:
http://192.168.25.173

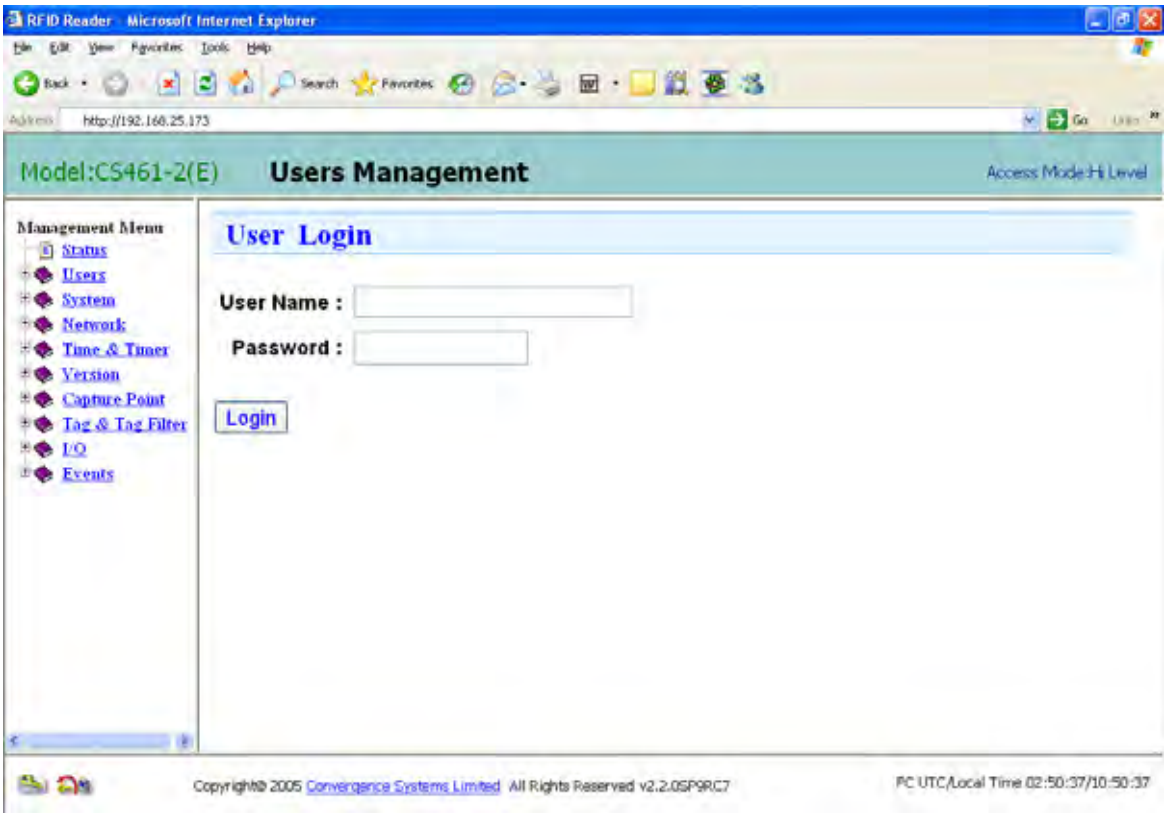


Figure 4-1

Caution:

If you see a blank page after entering the web interface, please refer to section 2.2 step 2. to configure the Internet Explorer settings and install the Microsoft XML Core Services (MSXML 4.0 Service Pack 2).

4.2 Status

The “Status” page gives a quick overview of the current status of the reader.

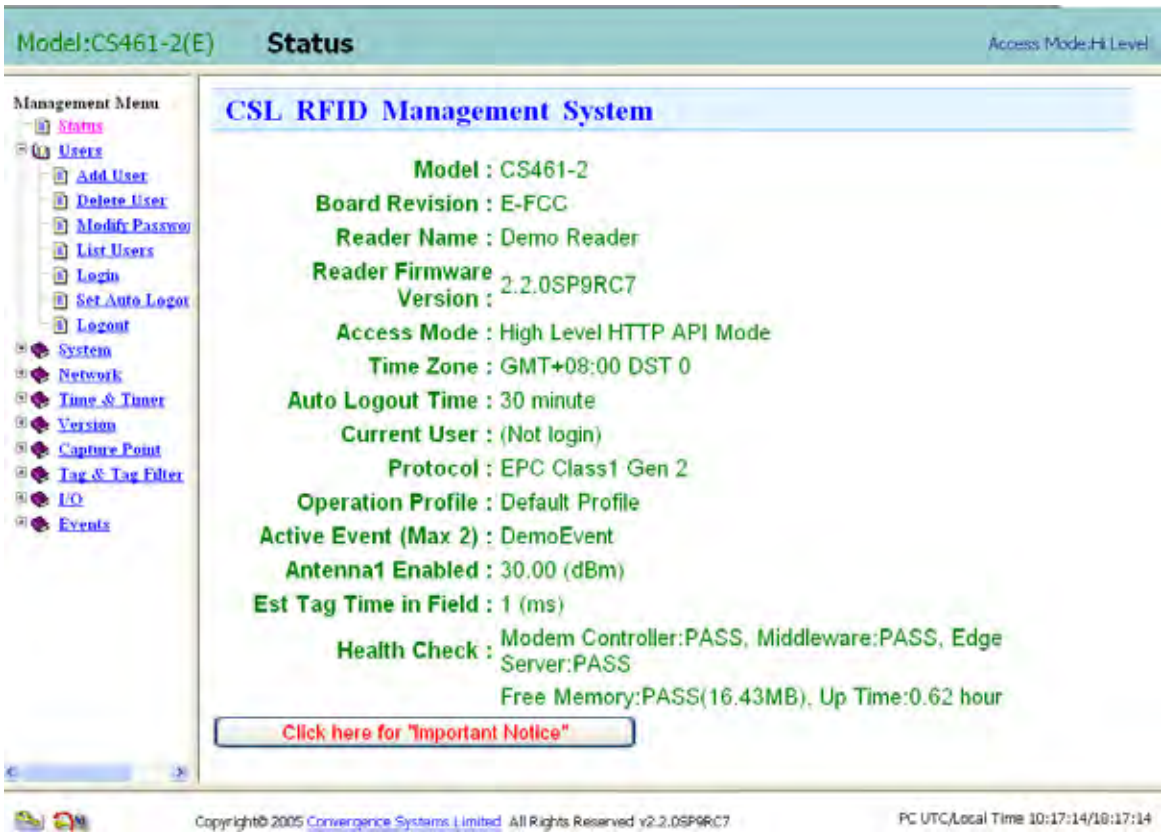


Figure 4-2 Status

4.3 Users Management

The "Users Management" page contains sub-menu for adding, deleting and modifying password, set auto-logout time and login/logout.

Figure 4-3 User Management

4.3.1 Add User

To add user, input the user name, password, authorization level and description. Then click "Add".

Figure 4-4 Add User

4.3.2 Delete User

To delete user, select the user name and then click “Delete”.

Figure 4-5 Delete User

4.3.3 Modify Password

To modify password, input the current password, new password and retype new password. Then click “Modify”.

Figure 4-6 Modify Password

4.3.4 List Users

The “List Users” page lists all the users and his/her authority.

Figure 4-7 List User

4.3.5 Set Auto Logout Time

The “Set Auto Logout Time” page allows ones to set the time for automatic logout if the user is idle.

Figure 4-8 Set Auto Logout Time

4.3.6 Login/Logout

The “Login/Logout” page is for users to login or logout the web browser interface.

4.4 System Management

The "System Management" page contains many sub-menus to configure the reader for operation. Users are recommended to access these pages to determine the required settings before any operation.

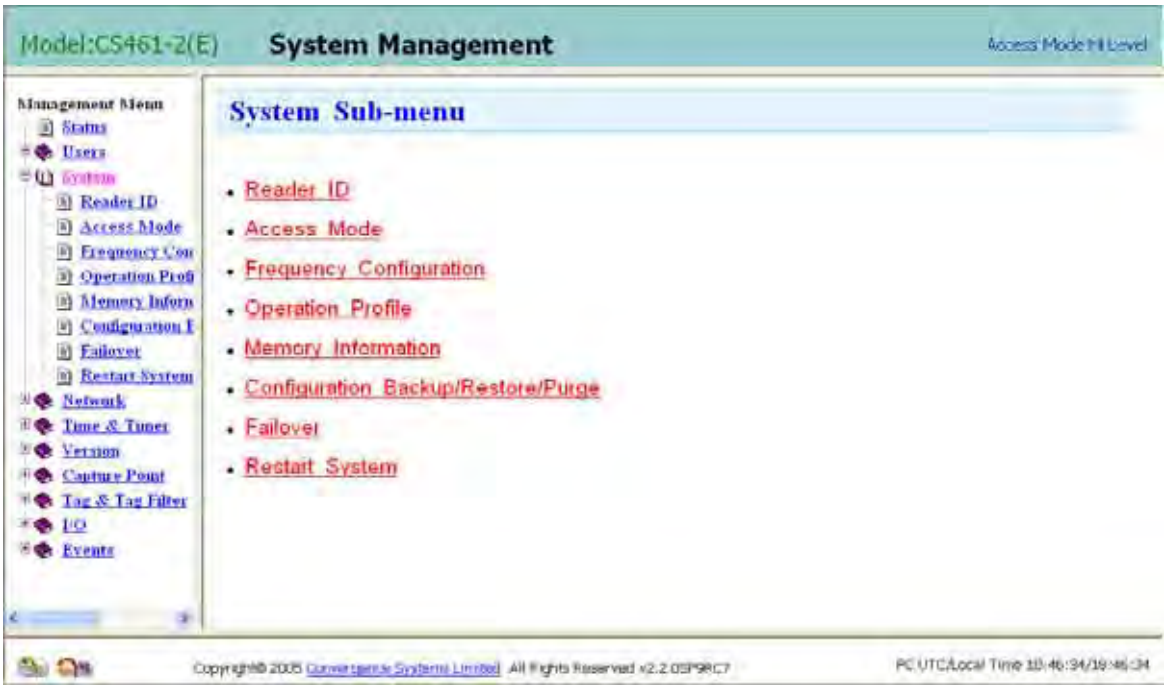


Figure 4-9 System Management

4.4.1 Reader ID

Here is the “Reader ID” submenu:

Figure 4-10 Reader ID

Set Reader ID

One can then set the unique ID for the reader. This is needed for easy future reference and programming access:

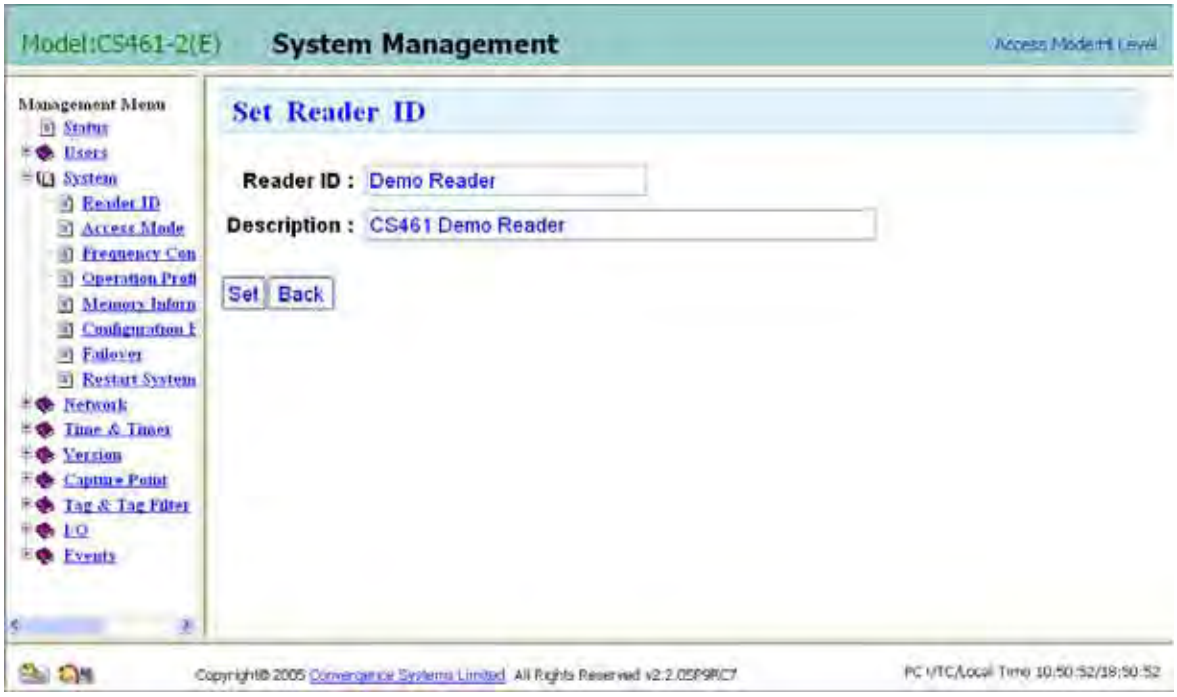


Figure 4-11 Set Reader ID

Get Reader ID

The reader ID can be retrieved by the “Get Reader ID” page:

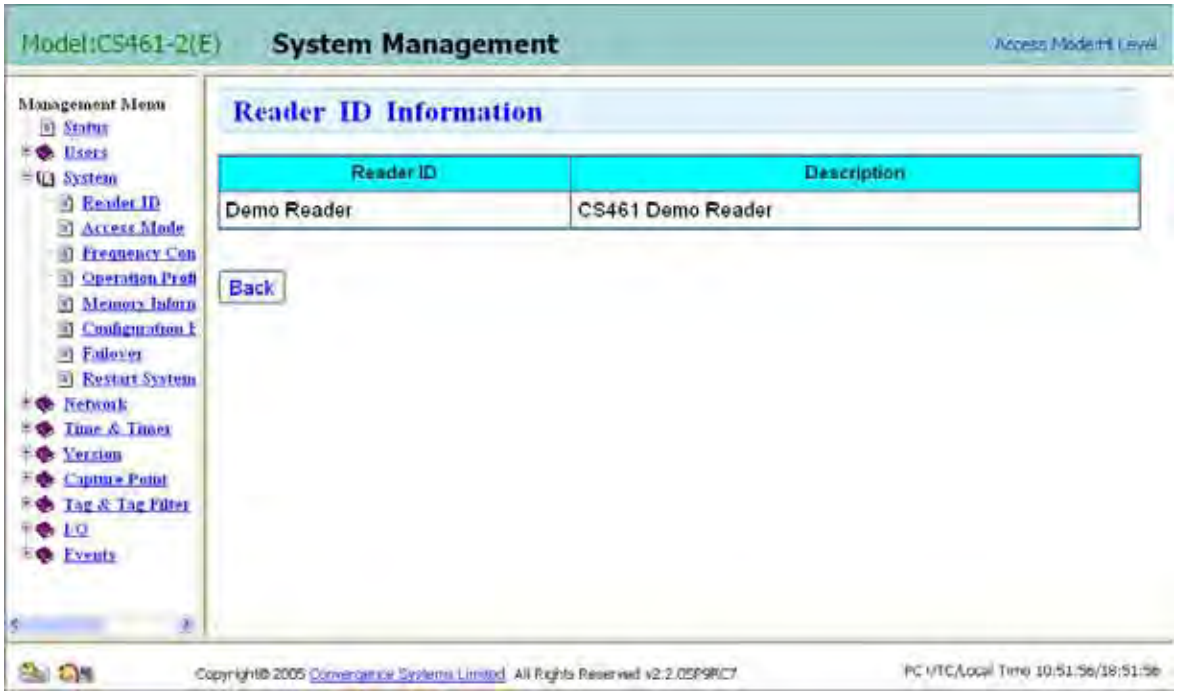


Figure 4-12 Get Reader ID

4.4.2 Access Mode

Here is the “Access Mode” submenu:

Figure 4-13 Access Mode

Set Access Mode

For configuring the reader to be controlled by High Level API, please remember to set the Access Mode to “High Level HTTP API Mode”

For configuring the reader to be controlled by MACH1 API (Low-Level), please remember to set the Access Mode to “Low Level MACH1 API Mode”.

If you require to configure the reader on the web interface (e.g. Network setting, Time settings etc.), you must set the Access Mode to “High Level HTTP API Mode”.

Copyright © Convergence Systems Limited, All Rights Reserved

Figure 4-14 Set Access Mode

Get Access Mode

The access mode can be retrieved by the "Get Access Mode" page:

Figure 4-15 Get Access Mode

4.4.3 Frequency Configuration

The “Frequency Configuration” page allows user to configure the country and frequency to be used by the reader. Please refer to the regulatory law of your region for the allowed frequency to be used. Here is the “Frequency Configuration” submenu:

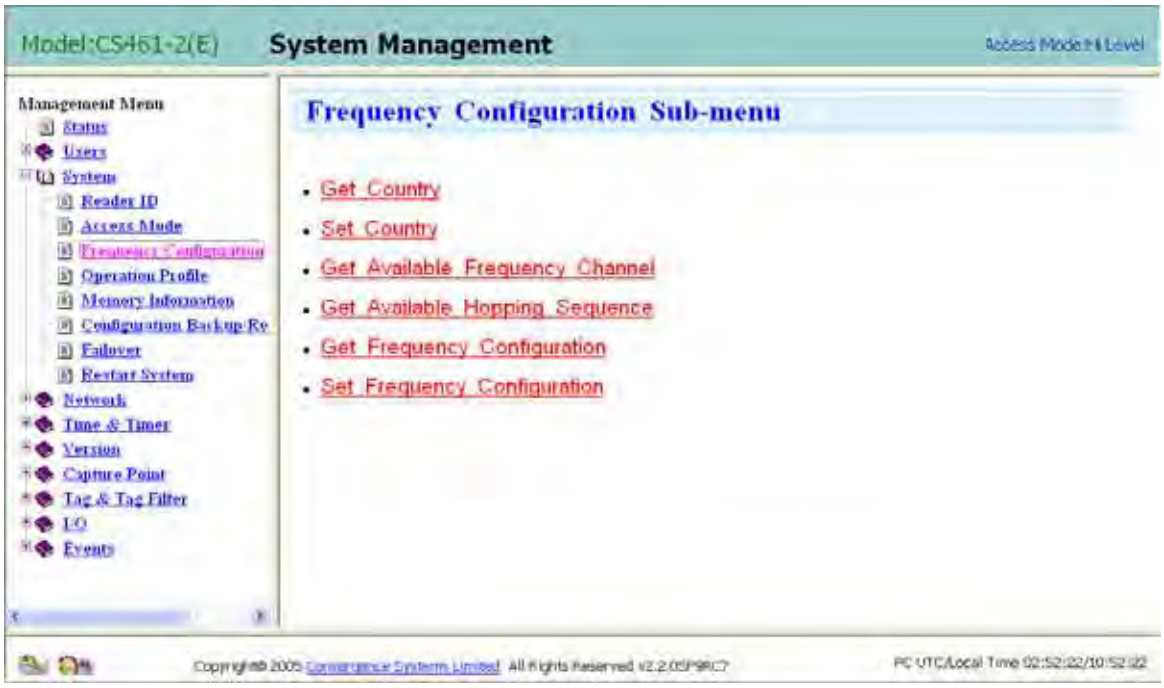


Figure 4-16 Frequency Configuration

Set Country

One can set the country in the “Set Country” page. Select the country and then click “Set”.

Figure 4-17 Set Country

Get Country

The selected country can be retrieved by the "Get Country" page:

Figure 4-18 Get Country

Get Available Frequency Channel

The “Get Available Frequency Channel” page allows one to retrieve the frequency channel available for the selected country.

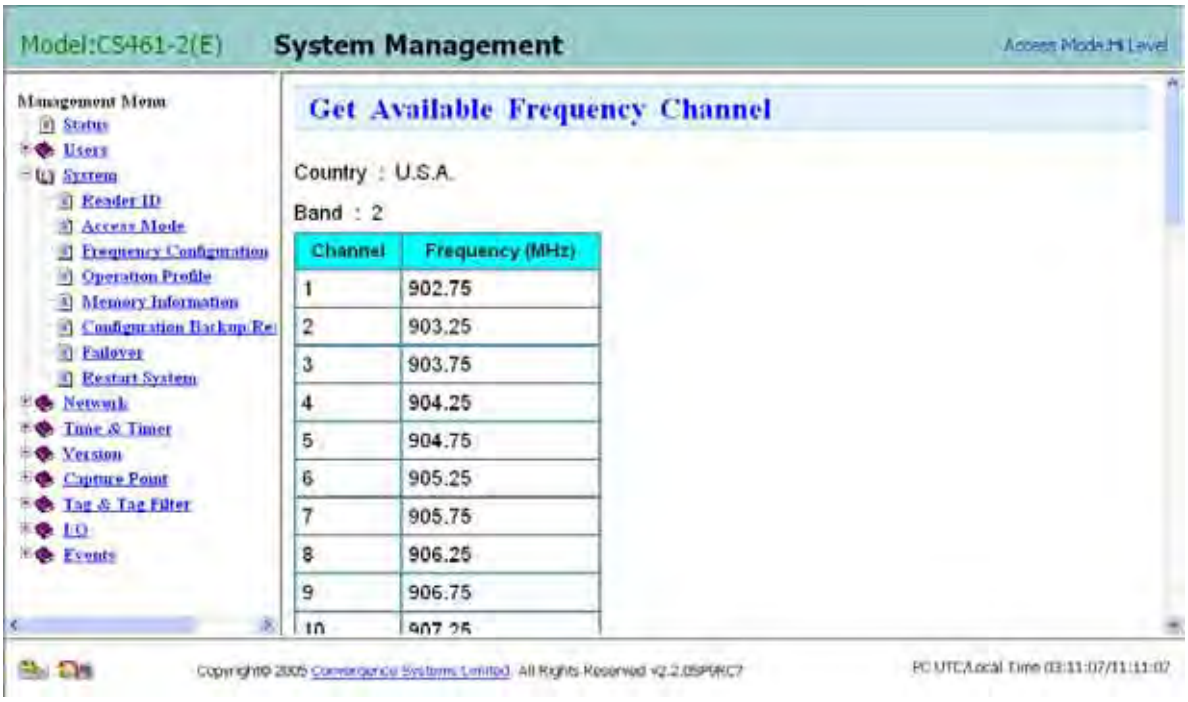


Figure 4-19 Get Available Frequency Channel

Get Available Hopping Sequence

The “Get Available Hopping” sequence page allows one to retrieve the frequency hopping sequence available for the selected country.

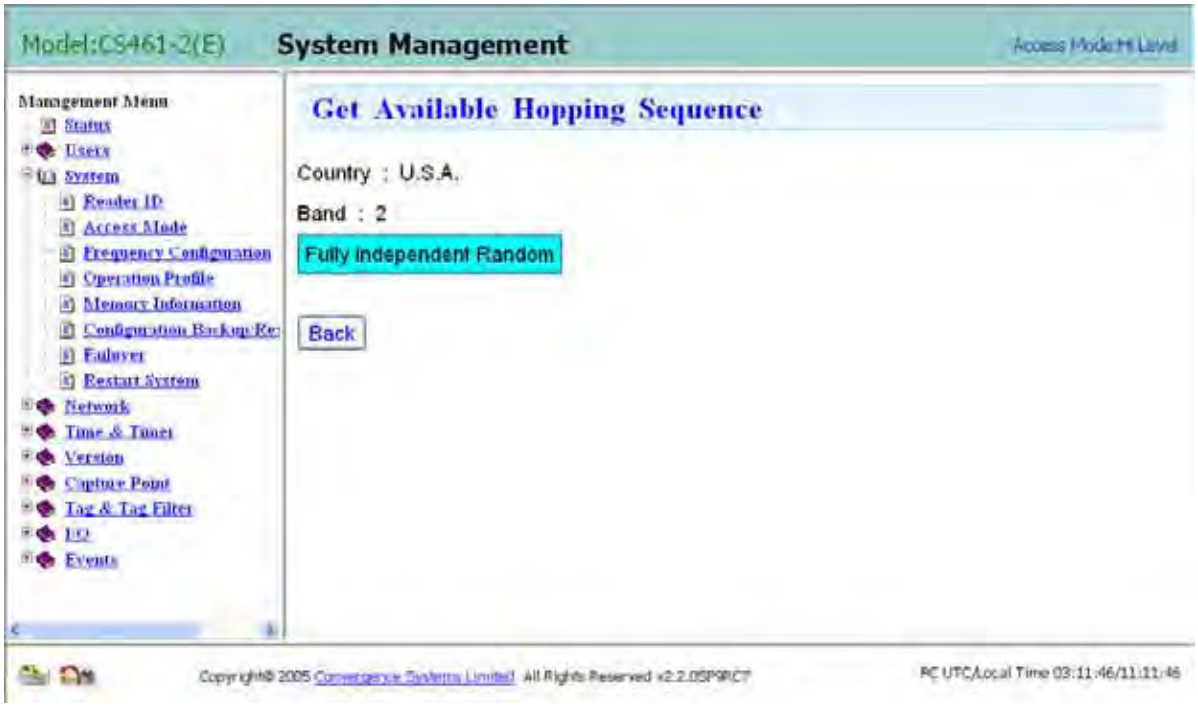


Figure 4-20 Get Available Hopping Sequence

Set Frequency Configuration

The "Set Frequency" page allows one to set the frequency used by the reader. For countries such as U.S.A, Australia, Korea and Taiwan, fixed frequency is not allowed and frequency hopping is used:

Figure 4-21 Set Frequency Configuration

For Europe and Japan, fixed frequency is allowed and set in Operation Profile:

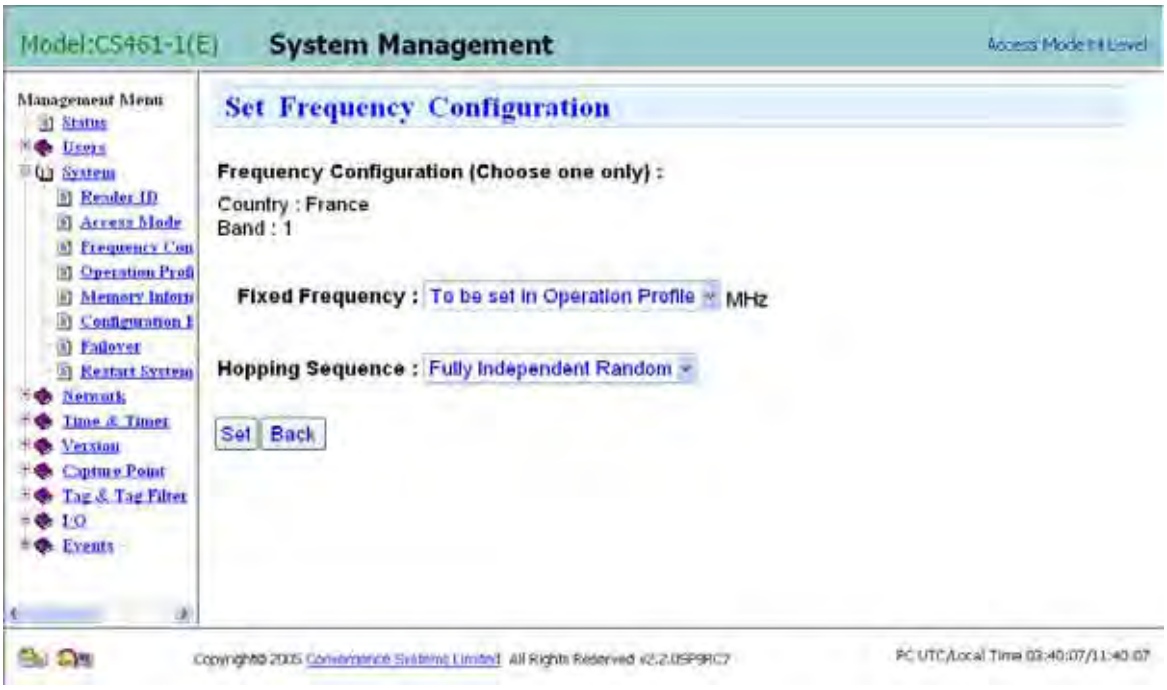


Figure 4-22 Set Frequency Configuration – Fixed Frequency

Get Frequency Configuration

The selected frequency can be retrieved by the “Get Frequency Configuration” page:

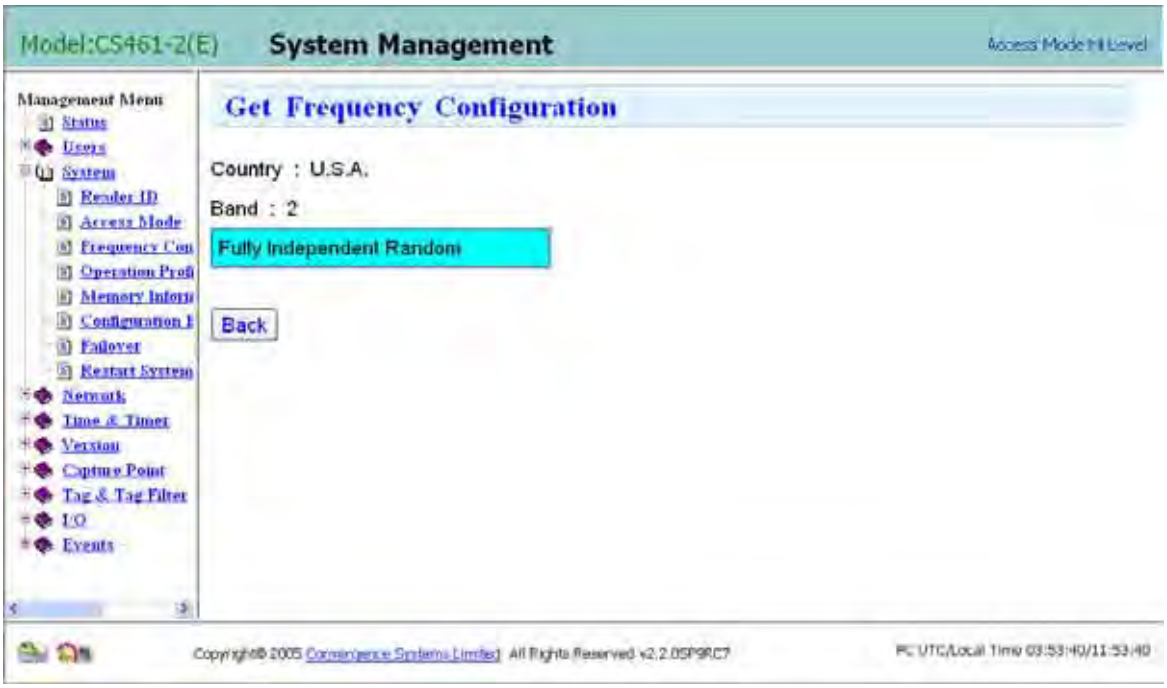


Figure 4-23 Get Frequency Configuration

4.4.4 Operation Profile

The "Operation Profile" page is extremely important as it sets the basic operation profile of the reader.

Parameter	Description
Modulation Profile	For Gen 2 there are different modulation profiles: ***0*** – Highest read rate ***1*** – Lower read rate than 0 but can tolerant higher level of interference ***2*** – Dense reader mode ***3*** – Dense reader mode with lower read rate than 2 but can tolerant higher level of interference ***4*** – Narrower frequency band
Population Est.	Estimated population of tags. It should be set to about 20% more than the maximum number of tags expected to be read at a time.
Session #	Session number must be different from reader to reader if they are pointing into the same zone.
Estimated Tag Time in field	An estimation of how long the tag will remain within the read zone of the antenna. One can calculate by estimating the antenna sweet zone (typically may be 4 feet if traversed across in front of the antenna, and is narrower the further away one is from the antenna) and divide that by the speed of movement of the tag. For fast tag, the reader will radiate all the time to ensure no tag is lost. For slow tag, the reader will selectively radiate less time when no tag is found. In any case, the reader will sample the space a minimum 5 times a second at low duty cycle. Although the default is 1000 ms, one can change that to 1 ms if the tags are moving very fast.

Copyright © Convergence Systems Limited, All Rights Reserved

Parameter	Description
Inventory Search Mode	***Single Target Large Population Inventory***: This mode is for reading a large number of tags at a time accurately. When this mode is used, tags that are read already will not respond to the reader for a short period of time. This can avoid the strong tags from dominating the week ones. This mode should be used with session #1.
Duplicate Elimination Triggering Method	***Autonomous Time Trigger:*** A tag will only be reported once within a duplicate elimination time ***Polling Trigger by Client:*** Tags read are buffered in reader until client application polls the read result. A tag will only be reported once in each polling trigger.
Autonomous Duplicate Elimination Time	If ***Autonomous Time Trigger*** is selected in Duplicate Elimination Triggering Method, this field must be input. It describes the time span of a duplicate elimination cycle, within which duplicate tags will be removed.
Receive Sensitivity	***Maximum Sensitivity*** All tags received by the reader are considered to be read ***Variable Sensitivity*** Different sensitivity can be applied to different antenna. Only tags received reaching the configured sensitivity are considered to be read.
Capture Point	The capture points that one wants to use must be selected. The capture points are unique names (defined in capture point page) corresponding to each of the four antennas, Antenna 1, 2, 3 and 4. For each capture point, the power output, in terms of dBm (dB over a milliwatt) needs to be set. If ***Variable Sensitivity*** is selected in Receive Sensitivity, one should also set the receive sensitivity in terms of dBm for tags to be considered as read.

Copyright © Convergence Systems Limited, All Rights Reserved

Figure 4-24 Operation Profile

For countries such as Japan and Europe, the interface is slightly more complicated where the user needs to select the frequency channels as well because these countries do not use hopping but rather use fixed frequency for operation. Also, for some profiles, it may only be usable in certain frequency channel. For example, profile 0 is a high speed non dense reader mode, and as such, its spectrum is wide and therefore for Japan it can only be run in frequency channel 5. So if you select profile 0, then the only frequency available is channel 5. If you choose profile 1, 2 and 3, then you see all 9 nine channels available for selection. Please see the next two figures that demonstrate the available frequencies for the different profiles are actually different. Furthermore, there is an additional selection where if the frequency channel is noisy thus making Listen Before Talk (LBT) mechanism stops the reader from radiating, then you can elect to have reader automatically search for another fixed channel with less noisy environment.

Copyright © Convergence Systems Limited, All Rights Reserved

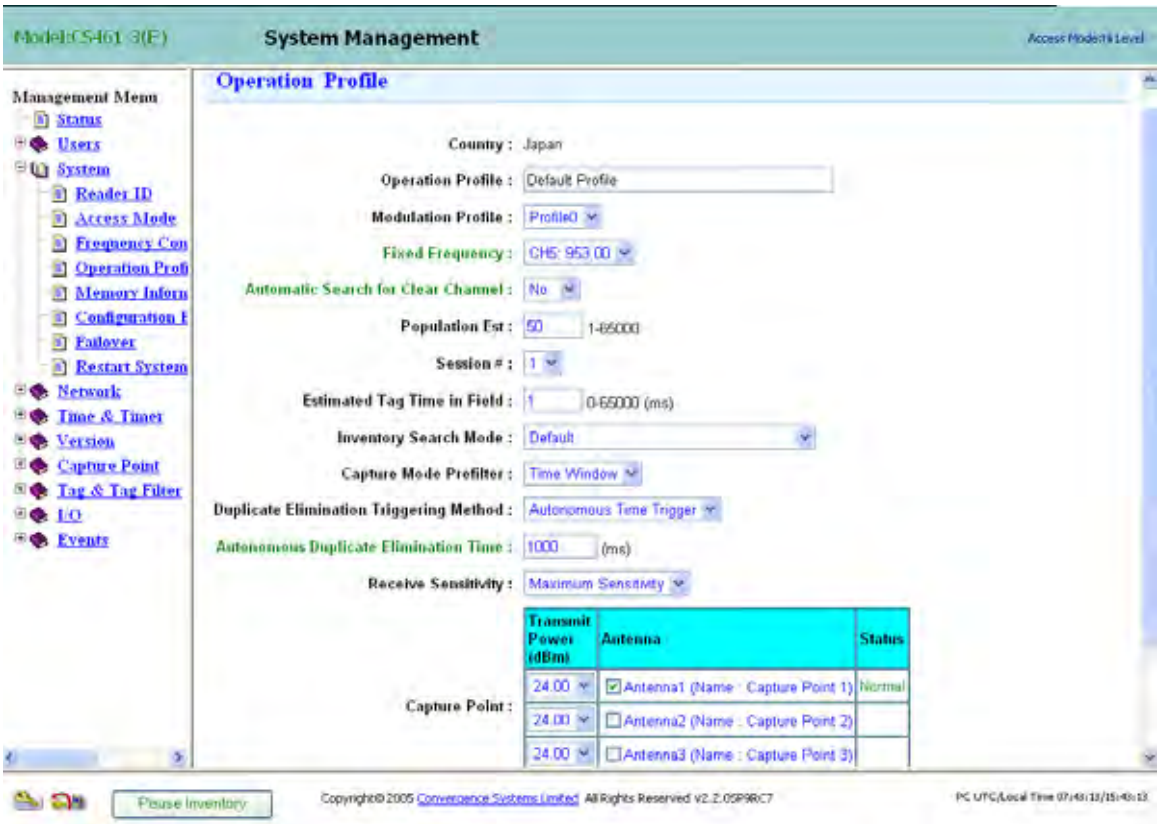

System Management
Management Menu
Status
Users
System
Reader ID
Access Mode
Frequency Con
Operation Profi
Memory Inform
Configuration F
Failover
Restart System
Network
Time & Timer
Version
Capture Point
Tag & Tag Filter
I/O
Events
Operation Profile
Country : Japan
Operation Profile : Default Profile
Modulation Profile : Profile0
Fixed Frequency : CH5: 953.00
Automatic Search for Clear Channel : No
Population Est : 50 1-65000
Session # : 1
Estimated Tag Time in Field : 1 0-65000 (ms)
Inventory Search Mode : Default
Capture Mode Prefilter : Time Window
Duplicate Elimination Triggering Method : Autonomous Time Trigger
Autonomous Duplicate Elimination Time : 1000 (ms)
Receive Sensitivity : Maximum Sensitivity
Capture Point :
Transmit Power (dBm)
Antenna
Status
24.00
Antenna1 (Name : Capture Point 1)
Normal
24.00
Antenna2 (Name : Capture Point 2)
24.00
Antenna3 (Name : Capture Point 3)
Pause Inventory

Figure 4-25

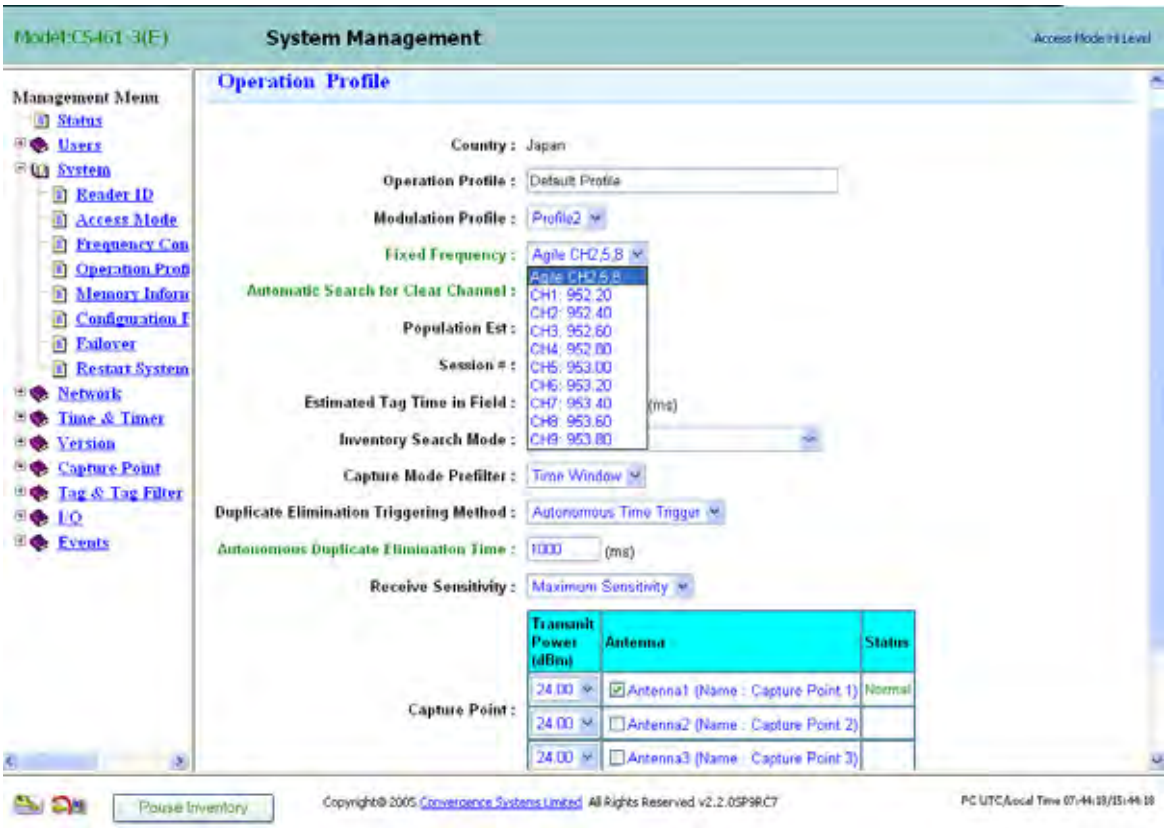

Model:CS461-3(E)
System Management
Access Mode:Hi Level
Management Menu
Status
Users
System
Reader ID
Access Mode
Frequency Con
Operation Prof
Memory Inform
Configuration I
Failover
Restart System
Network
Time & Timer
Version
Capture Point
Tag & Tag Filter
I/O
Events
Operation Profile
Country : Japan
Operation Profile : Default Profile
Modulation Profile : Profile2
Fixed Frequency : Agile CH2,5,8
Automatic Search for Clear Channel :
Population Est :
Session # :
Estimated Tag Time in Field : (ms)
Inventory Search Mode :
Capture Mode Prefilter : Time Window
Duplicate Elimination Triggering Method : Autonomous Time Trigger
Autonomous Duplicate Elimination Time : 1000 (ms)
Receive Sensitivity : Maximum Sensitivity
Capture Point :
Transmit Power (dBm)
Antenna
Status
24.00
Antenna1 (Name : Capture Point 1)
Normal
24.00
Antenna2 (Name : Capture Point 2)
24.00
Antenna3 (Name : Capture Point 3)
CH1: 952.20
CH2: 952.40
CH3: 952.60
CH4: 952.80
CH5: 953.00
CH6: 953.20
CH7: 953.40
CH8: 953.60
CH9: 953.80
Pause Inventory
Copyright© 2005 Convergence Systems Limited All Rights Reserved v2.2.0SP9RC7

Figure 4-26

4.4.5 Memory Information

The “Memory Information” page shows the RAM and Flash memory available.

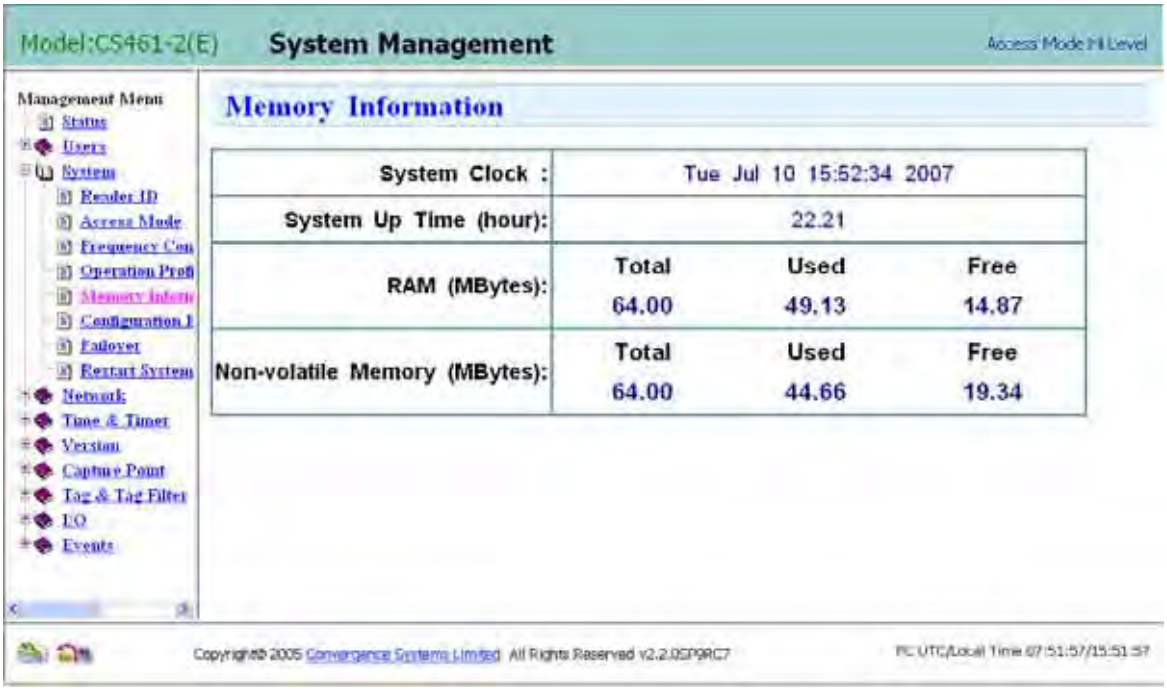


Figure 4-27 Memory Management

4.4.6 Configuration Backup/Restore/Purge

The “Configuration Backup/Restore/Purge” page allows saving of configurations, restoring of configurations and restoring the factory default

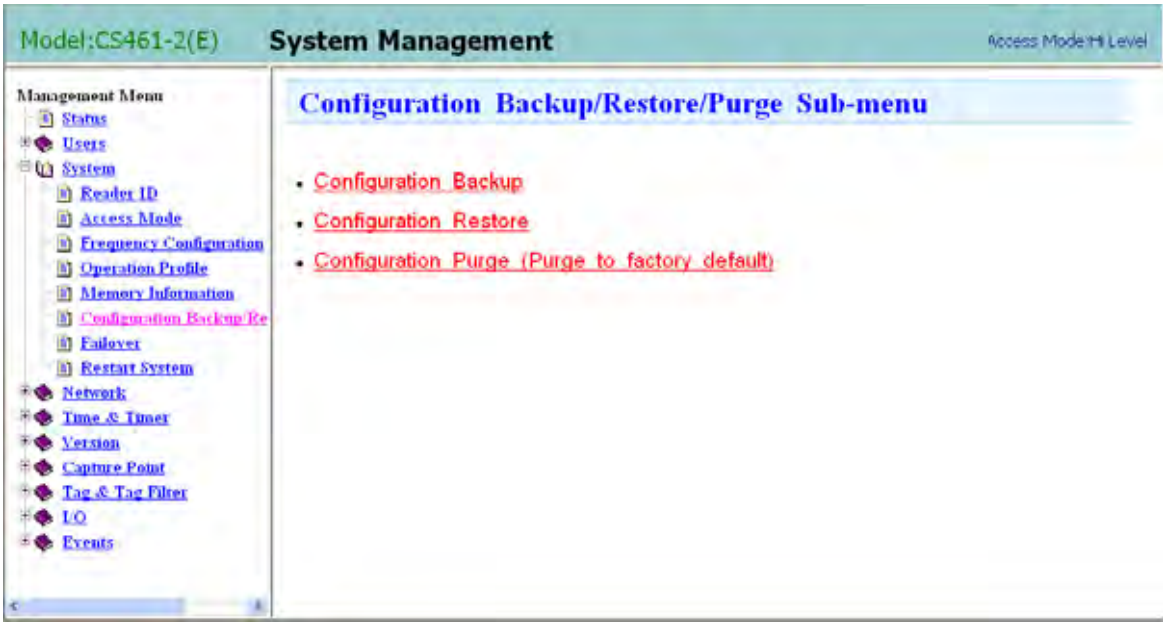


Figure 4-28 Configuration Backup/Restore/Purge

Configuration Backup

To backup configuration, click “Proceed” in “Configuration Backup” page. The system would start to backup the configuration to file.

Copyright © Convergence Systems Limited, All Rights Reserved

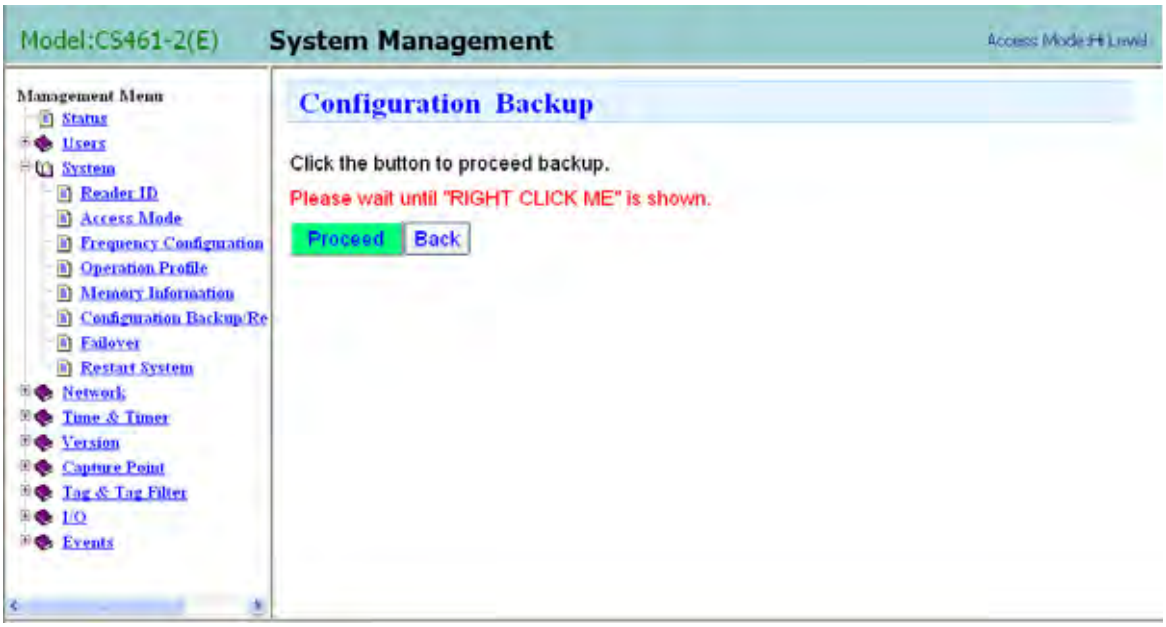


Figure 4-29 Configuration Backup

Once the backup is finish, a link “RIGHT CLICK ME” is displayed. One can then save the configuration backup by right clicking the link and select “Save Target As…”.

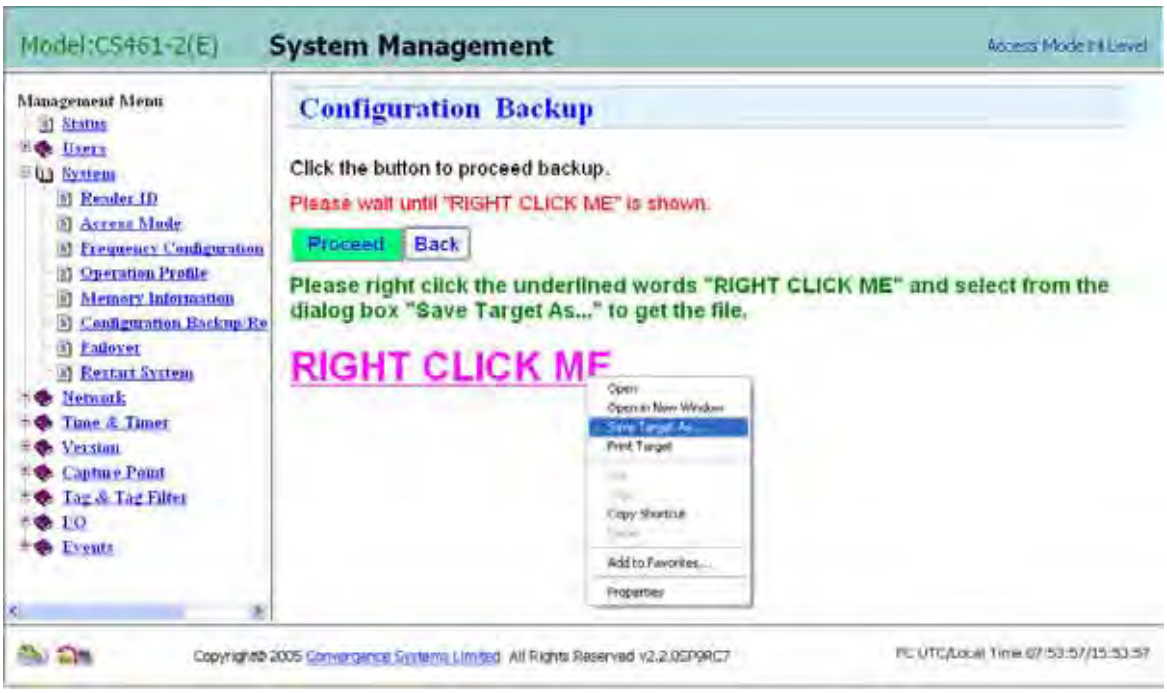


Figure 4-30 Configuration Backup (Cont’d)

Configuration Restore

To restore backup configuration, click “Browse…”. Then select the backup configuration file and click “Proceed”.

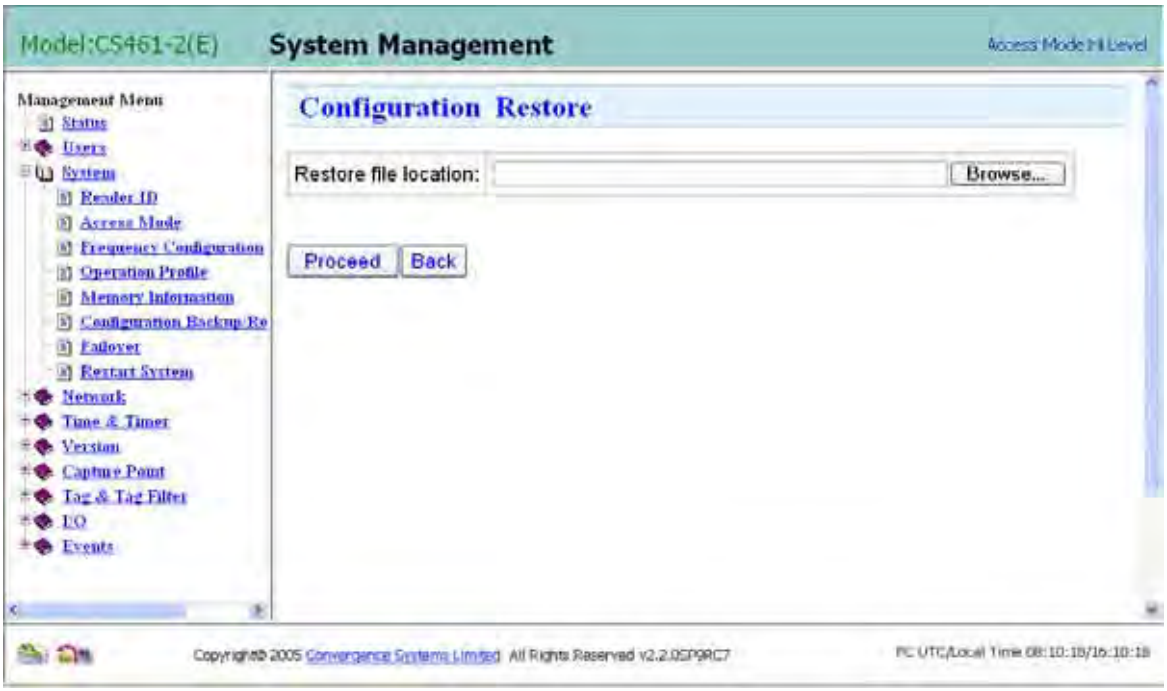


Figure 4-31 Configuration Restore

Configuration Purge

To purge the configuration to factory default, input “Y” and click “Proceed”. Then, reboot the reader. Note that the reader IP will become 192.168.25.248.

Figure 4-32 Configuration Purge

4.4.7 Failover Configuration

One can enable network failure data backlog in the “Failover Configuration” page. It allows the reader to buffer the tags read during network failure in memory. Buffered tags are sent to trusted server when network is restored.

A further selection is necessary in the way the data is backlogged: stop when full or overwrite when full. If one does not check that box, it means the backup will be stopped when buffer is full. If one checks that box, the backup will overwrite itself when buffer is full.

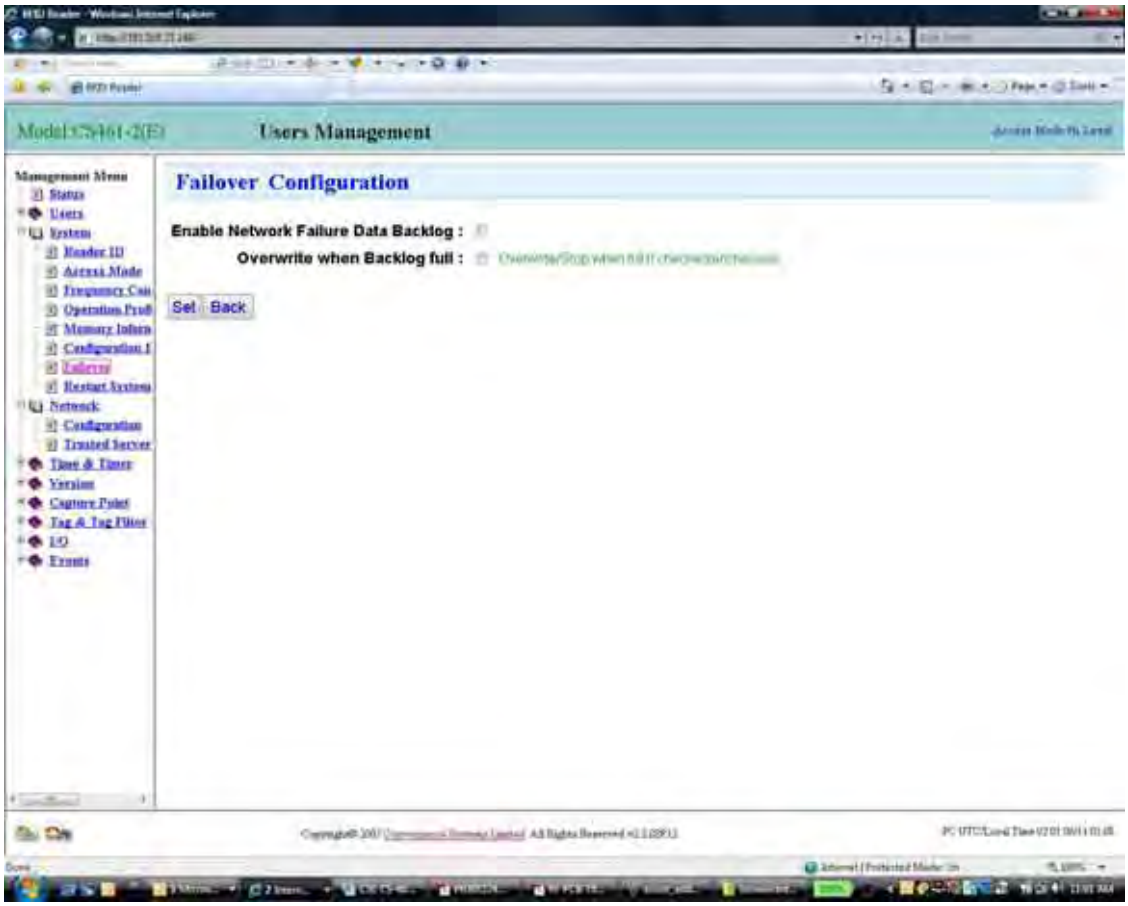


Figure 4-33 Failover Configuration

4.4.8 Restart System

To restart the system, input “Y” and click “Process”.

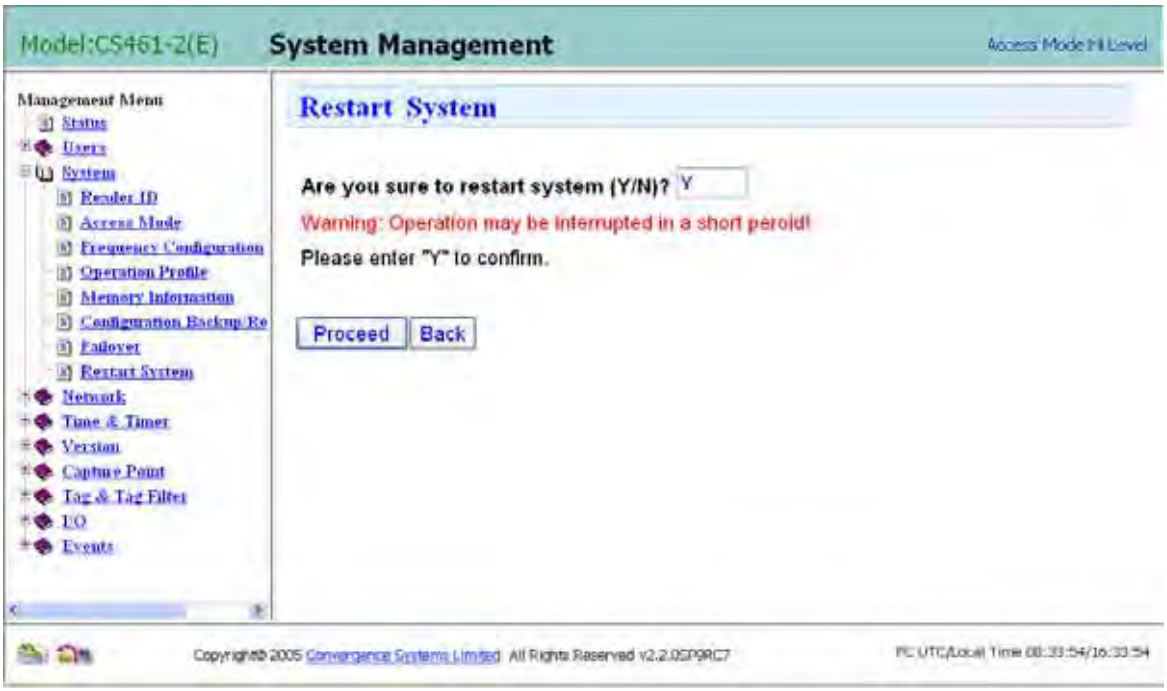


Figure 4-34 Restart System

4.5 Network Management

"Network Management" page allows the user to set the network parameters. Here is the network management sub-menu:

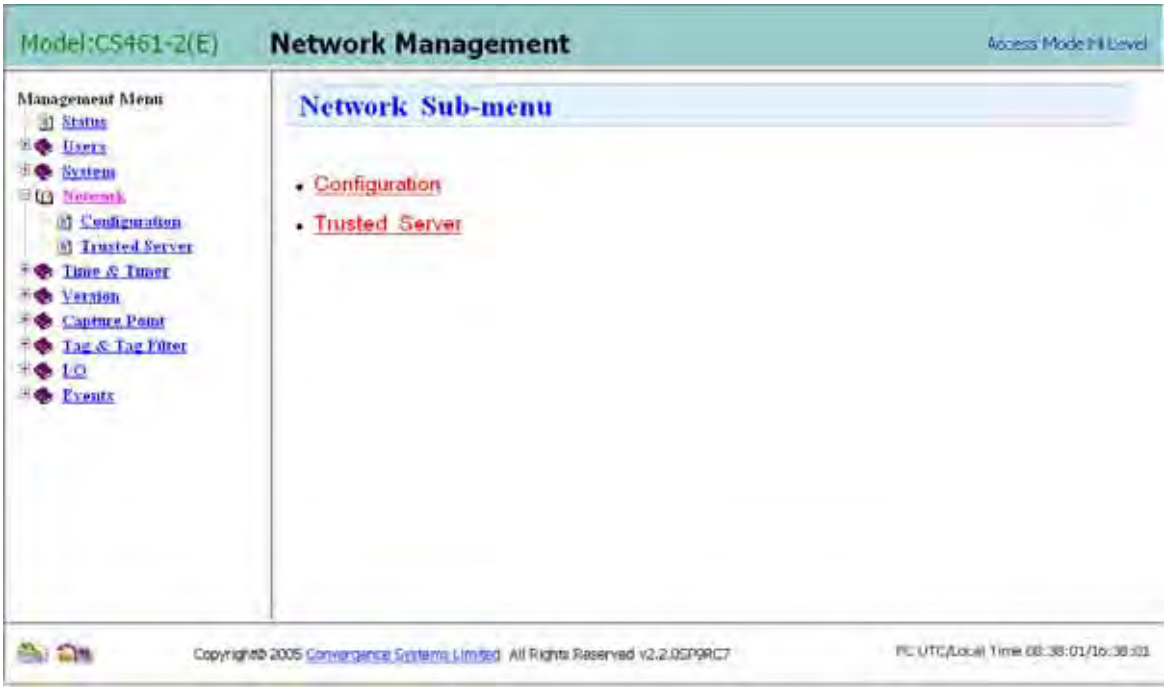


Figure 4-35 Network Management

4.5.1 Network Configuration

In “Network Configuration” page, one can input the network parameters such as the reader IP and port number, network mask, gateway and DNS server. However, if one changes the port number, make sure next time you access the web server you add the port number at the end of the IP address. For example: http://192.168.25.233:1238 (if not specified, port = 80 is assumed by Internet Explorer).

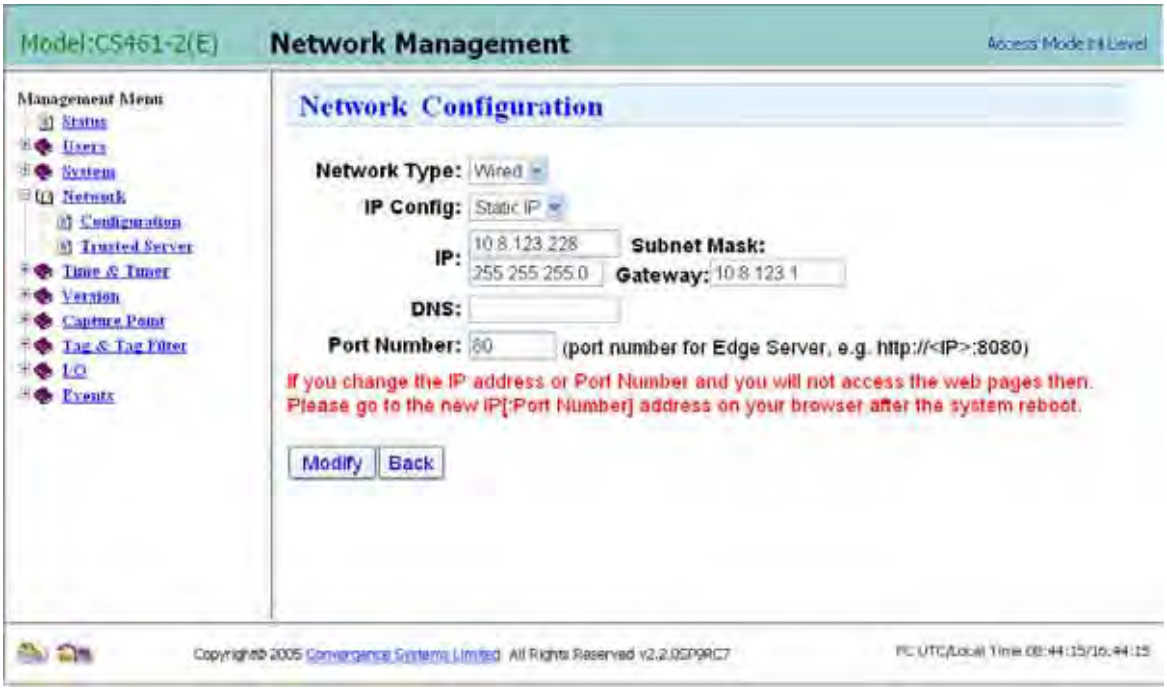


Figure 4-36 Network Configuration

4.5.2 Trusted Server

Trusted server for automatic data submission by the reader using the event engine is set in the "Trusted Server" page. Here is the "Trusted Server" submenu:

Figure 4-37 Trusted Server

Add Trusted Server

To add a trusted server for receiving tag events, input the IP address and port number. Also, there is a distinction between whether the listening port is on the PC side or on the reader side.

Figure 4-38 Add Trusted Server

Modify Trusted Server

To modify trusted server, select the server ID, modify any fields and click “Modify”.

Figure 4-39 Modify Trusted Server

Delete Trusted Server

To delete trusted server, select the server ID and click “Delete”.

Figure 4-40 Delete Trusted Server

List Trusted Server

Information of trusted server can be retrieved by the “List Trusted Server” page.

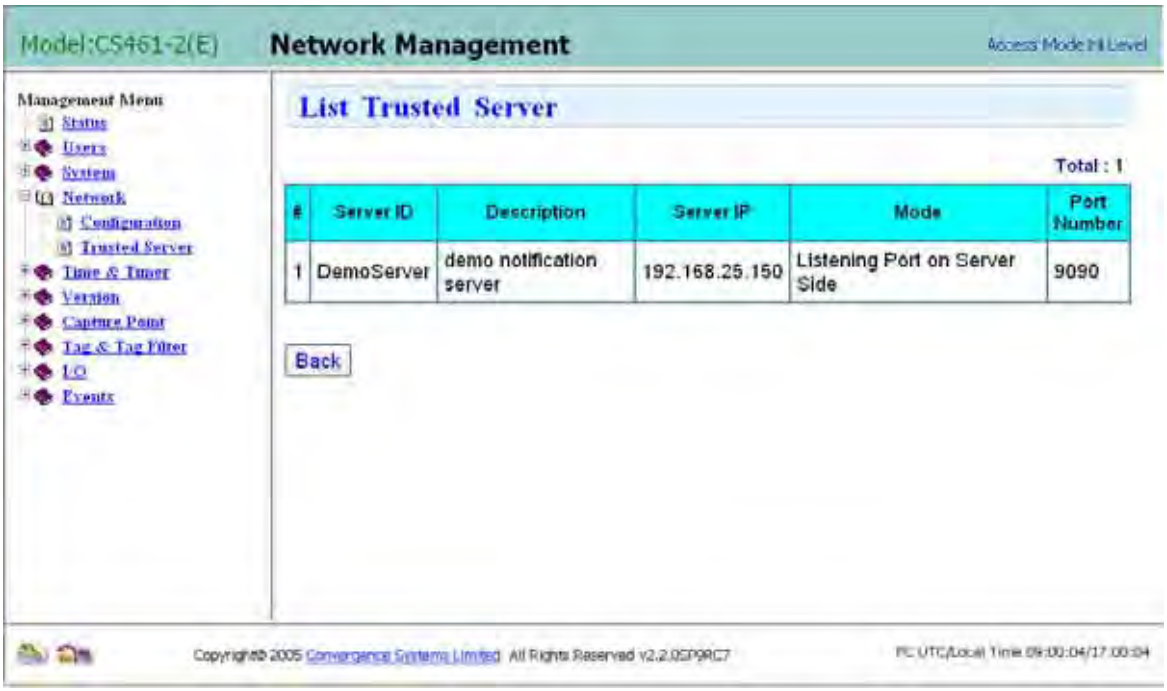


Figure 4-41 List Trusted Server

4.6 Time and Timer Setting

Here is the “Time and Timer Setting” submenu:

Figure 4-42 Time & Timer Setting

4.6.1 Date/Time

The "Date/Time" page allows the user to set the real time clock inside the reader. Please note that you have to configure the UTC (GMT) time on the reader. The local time will then be calculated based on the time zone you set. Note that for some country they also practice Daylight Savings Time.

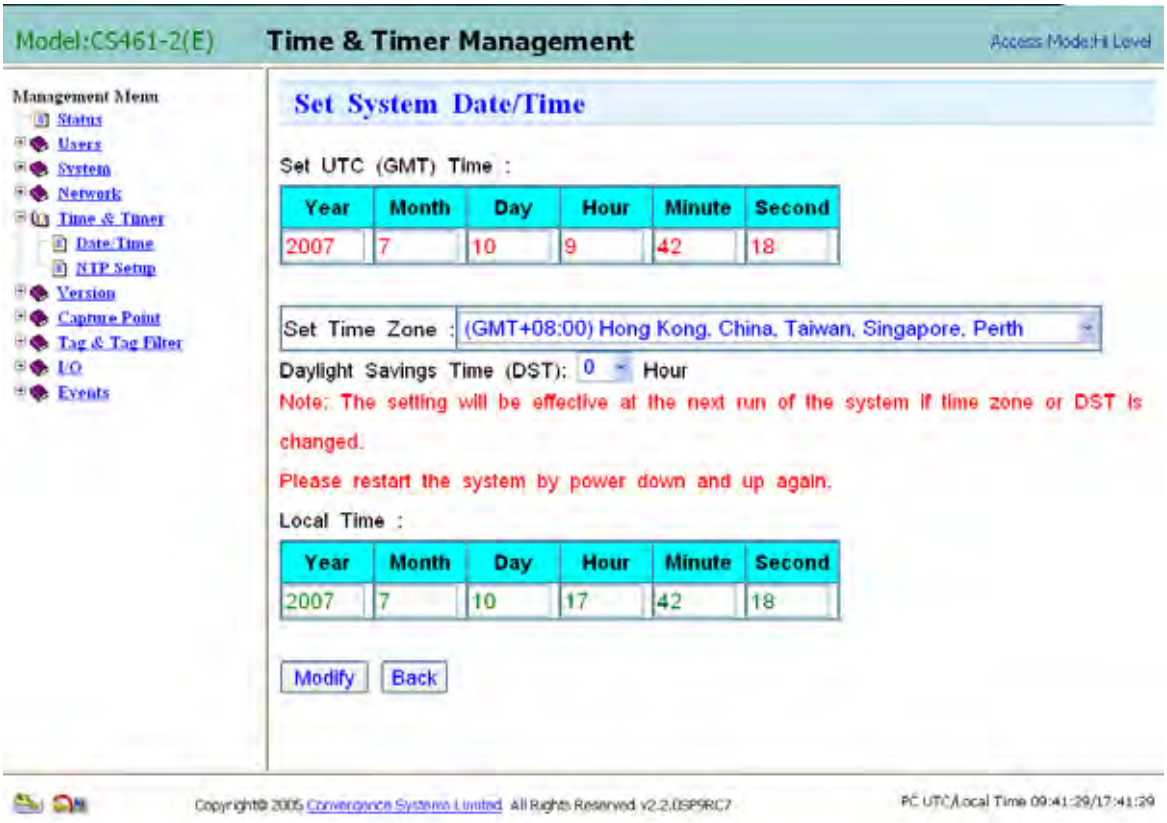


Figure 4-43 Set System Date & Time

Warning: After changing the date and time, the reader may pause reading 30-60 seconds for re-initiation. It is recommended not to open the "Capture Tags Testing" page in "Tag & Tag Filter" for viewing tags in this period.

4.6.2 NTP Setup

This page allows one to setup the NTP server. Be sure to enter the gateway and DNS server in the network configuration page in order for the NTP server be reacheable by the reader.

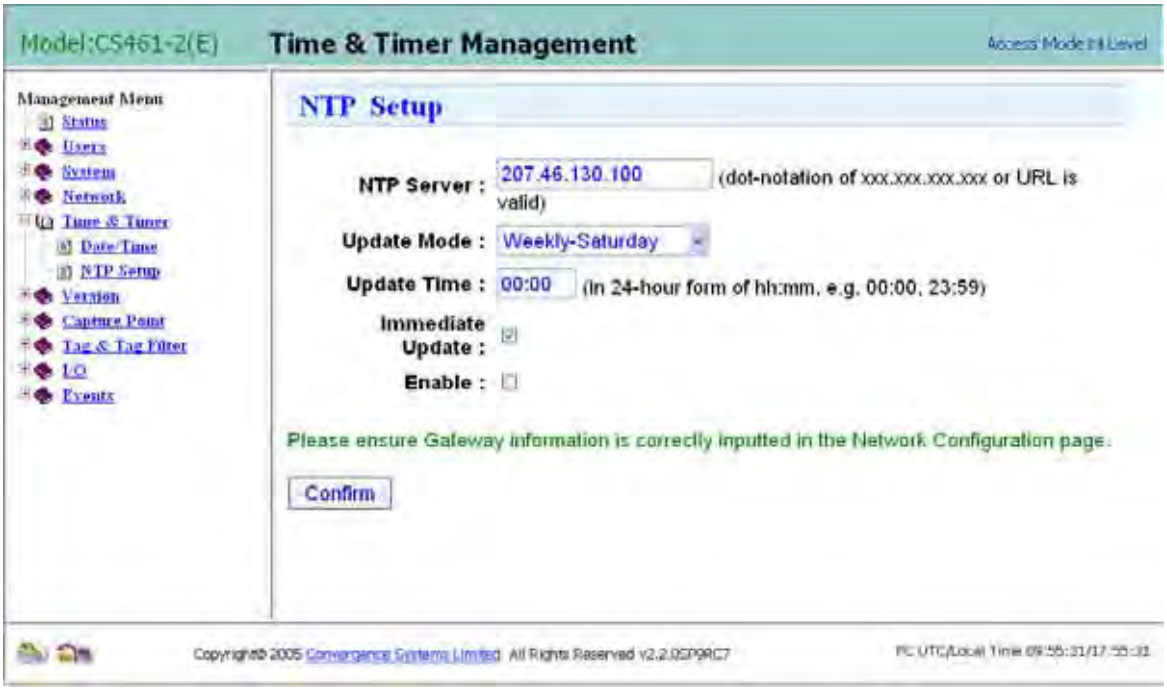


Figure 4-44 NTP Setup

4.7 Version Management

The “Version Management” page allows you to review the version upgrade history (in the Version Control Submenu), and to do firmware upgrade (in the Firmware Upgrade Submenu).

Figure 4-45 Version Management

Version Control

In the “Version Control” sub-menu page, one can see the version number of the software. This is an important page to check if the versions are correct, especially after a firmware upgrade. It also shows the upgrade history of the reader.

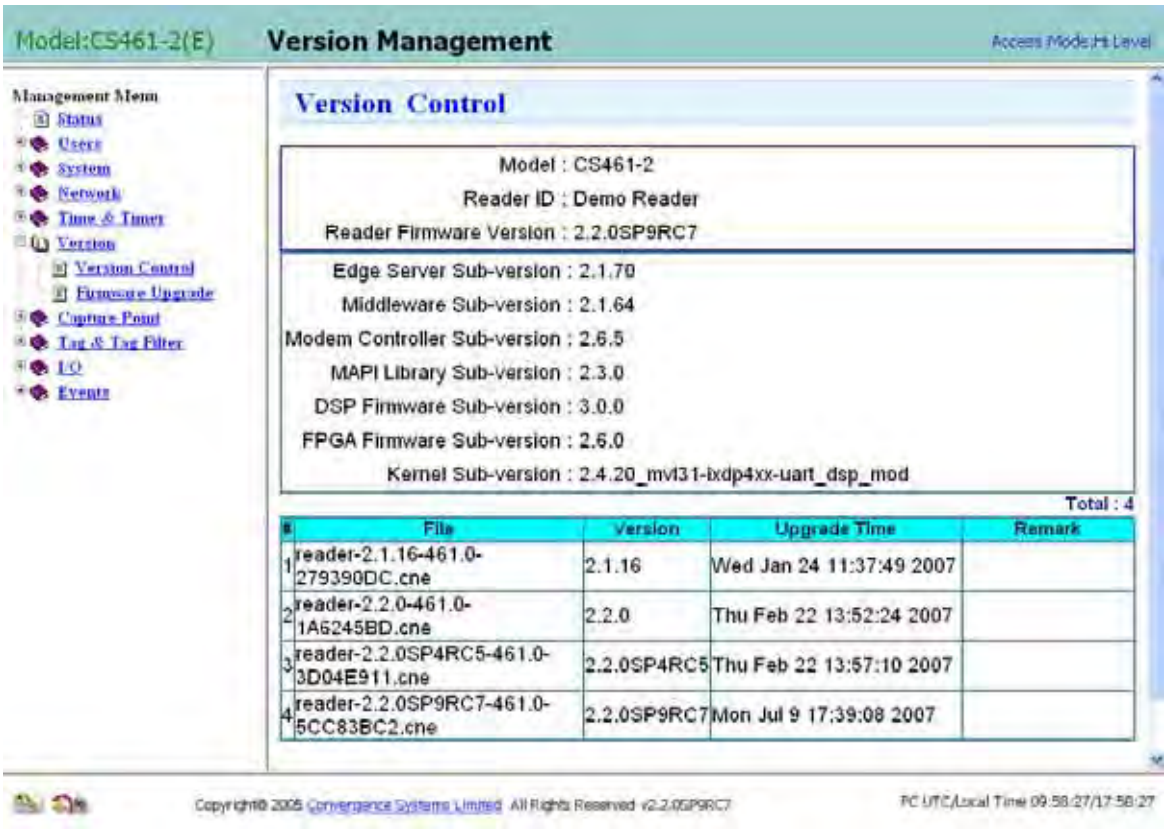


Figure 4-46 Version Control

Firmware Upgrade

In the “Firmware Upgrade” submenu, just press the “Browse” button and find the upgrade file (which the user has already downloaded from CSL website before). Just select the upgrade file with the .cne extension, without doing anything to it, and press open. Then press the “Firmware Upgrade” button. The upgrade takes a few minutes, depending on the size of that particular upgrade. Please wait until you see the success message. The reader needs to be rebooted afterward, either physically unplugging the power supply, or by the web interface (Restart page in System menu).

Figure 4-47 Firmware Upgrade

4.8 Capture Point

The name of each antenna port is the capture point name (Some readers call it read point name). This name can be configured. In other words, each antenna port (or capture point, or read point) can be uniquely identified and accessed or referred to. Note that the word capture and read are interchangeably used in the context of this reader. A capture point is the same as a read point.

Figure 4-48 Capture Point

4.9 Tag & Tag Filter

The “Tag & Tag Filter” page allows you to read (capture) tags, write tags and set access password. Read tags include the reading of Bank 01 (Bank 1) that contains the PC bits and the EPC ID bits (or whatever other information defined for use by customers), the reading of Bank 10 (Bank 2) that contains the TID or UID data (read only), and the reading of Bank 11 (Bank 3) that contains the User Memory bits. Write tags include the writing of Bank 1 and Bank 3. Please note that in this reader capture tag is the same as read tag, the words capture and read are interchangeably used. Below is the “Tag & Tag Filter” submenu:

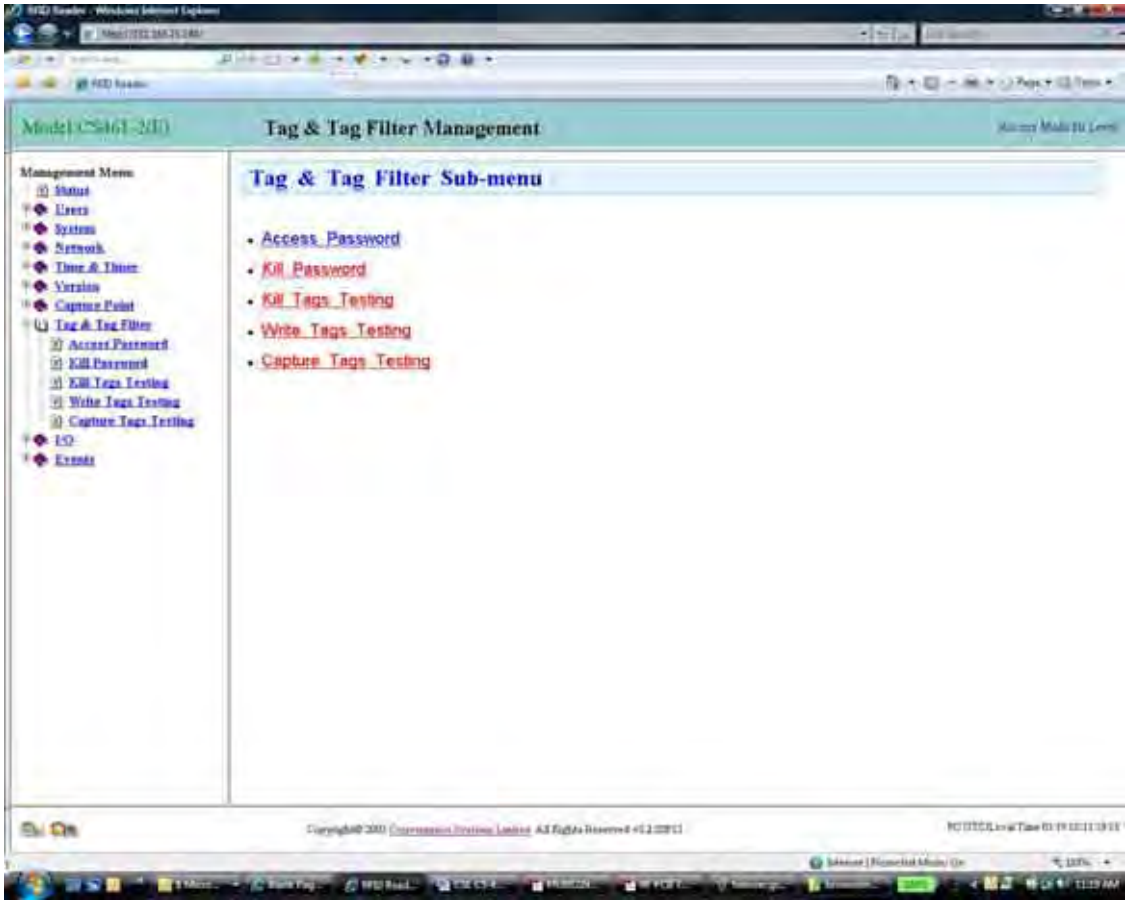


Figure 4-49 Tag & Tag Filter

Copyright © Convergence Systems Limited, All Rights Reserved

4.9.1 Access Password

In “Access Password” page, you can write access password or modify access password:

Figure 4-50 Access Password

Write Access Password

In “Write Access Password” page, you can use halt filter to select those tags that you want to write access password to, and then also choose to lock the access password (write lock AND read lock) afterward. Note that the normal practice is to write and read lock the access password. Only in rare cases should the end user not lock it. Moreover, you can select the “Write Fixed Number” mode such that the writing operation is stopped after writing a fixed number of tags.

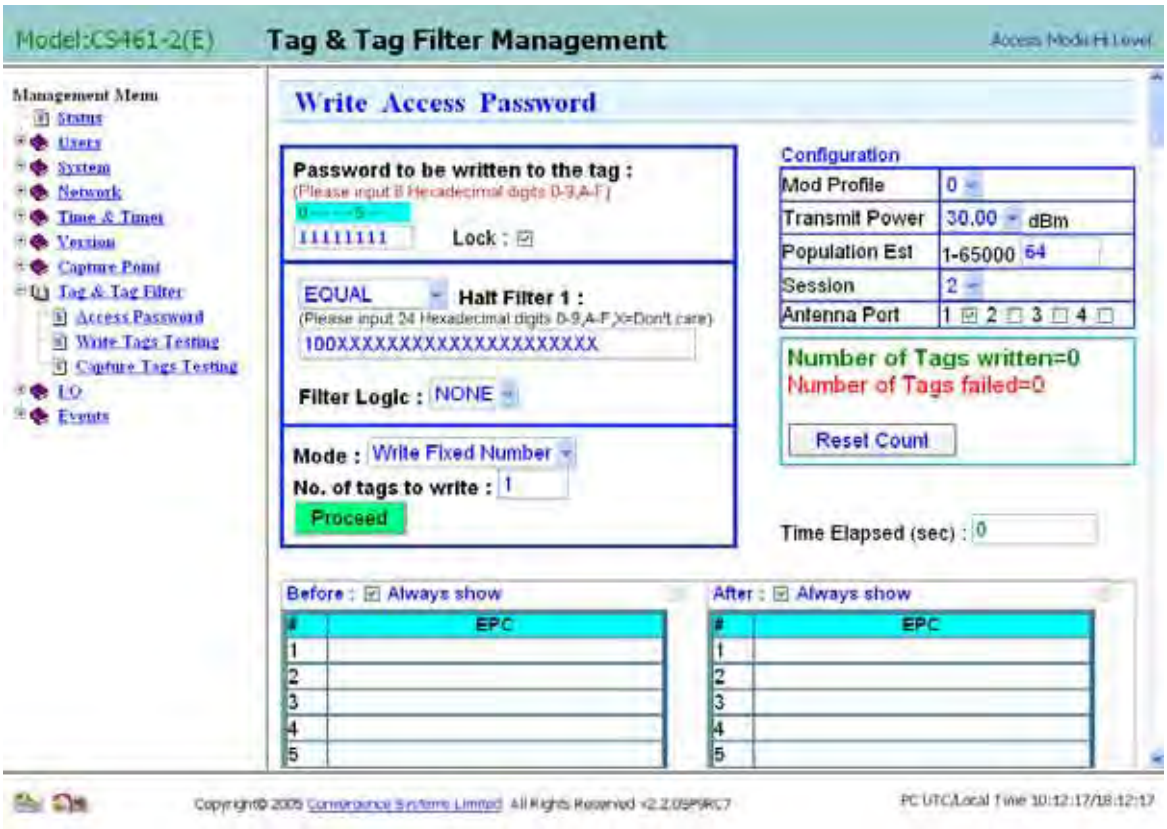


Figure 4-51 Write Access Password

Modify Access Password

In “Modify Access Password” page, you can use halt filter to select those tags that you want to modify access password to, and then also choose to lock the access password (write lock AND read lock) afterward. Note that the normal practice is to write and read lock the access password. Only in rare cases should the end user not lock it. Since for modify access password it implies the tag already has a password inside, then the user must input the current password in order to unlock it in the first place, thus there is an additional first line “Current Password for unlock”.

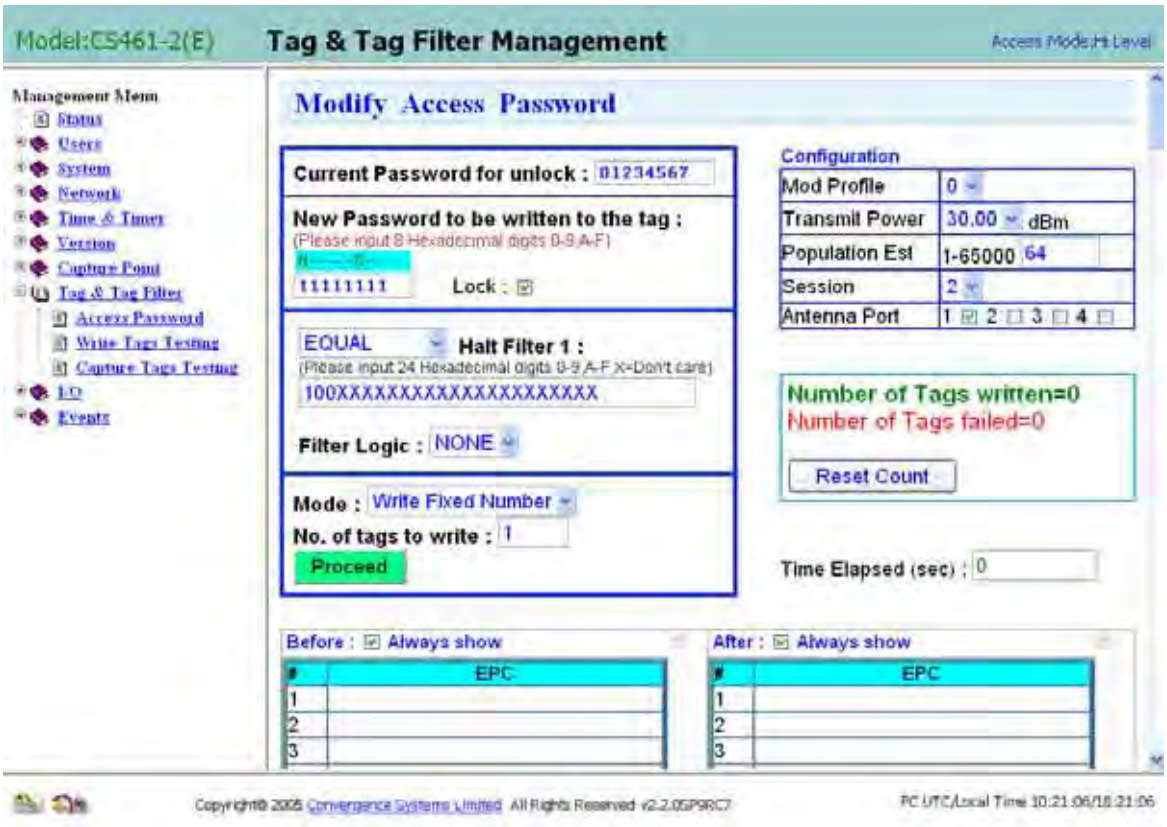
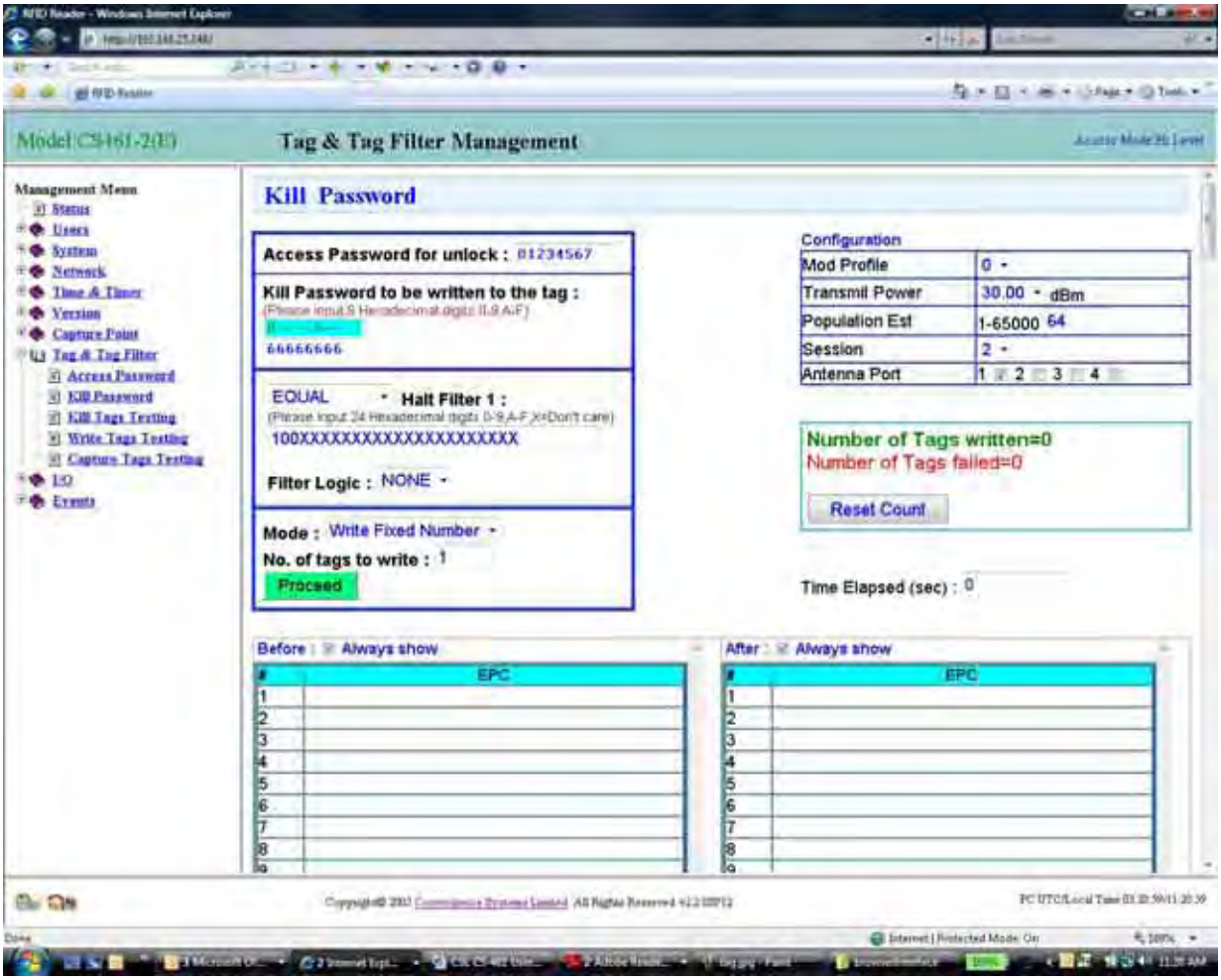


Figure 4-52 Modify Access Password

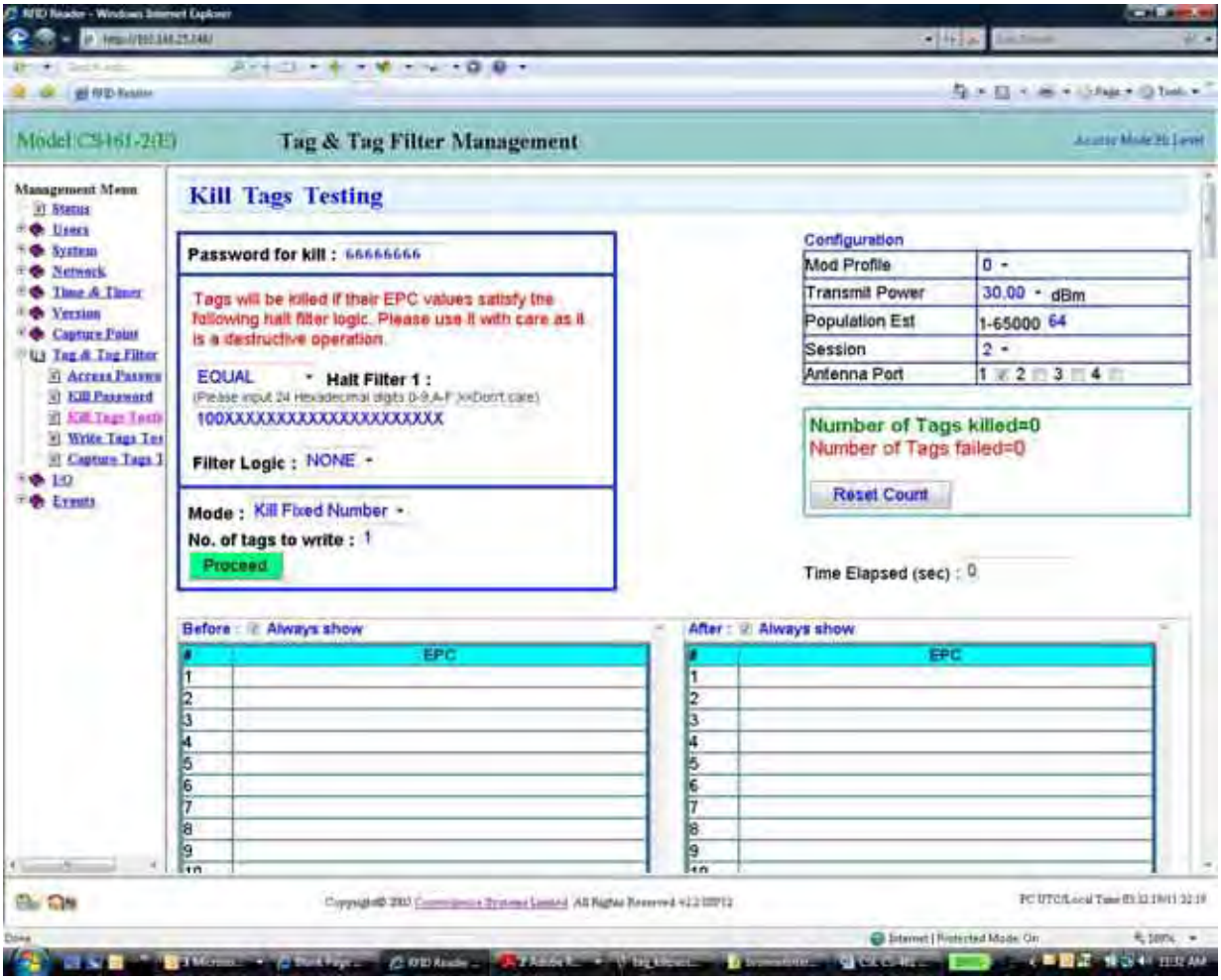
4.9.2 Kill Password

In Kill Password page you can write a kill password onto a tag based on some filtering logic to select some particular tags. In case the kill password has been locked, you can also specify the access password to unlock the tag for a new kill password.



4.9.3 Kill Tags Testing

In Kill Tag testing page you can test the ability of the reader to kill a batch of tags. One can select



4.9.4 Write Tags Testing

In “Write Tags Testing” page, there are a number of different pages for user to write tags:

Figure 4-53 Write Tags Testing

Write Tags Testing (with dual halt filters)

In this page, you can write EPC ID (within Bank 01) on any tag with up to 2 halt filters.

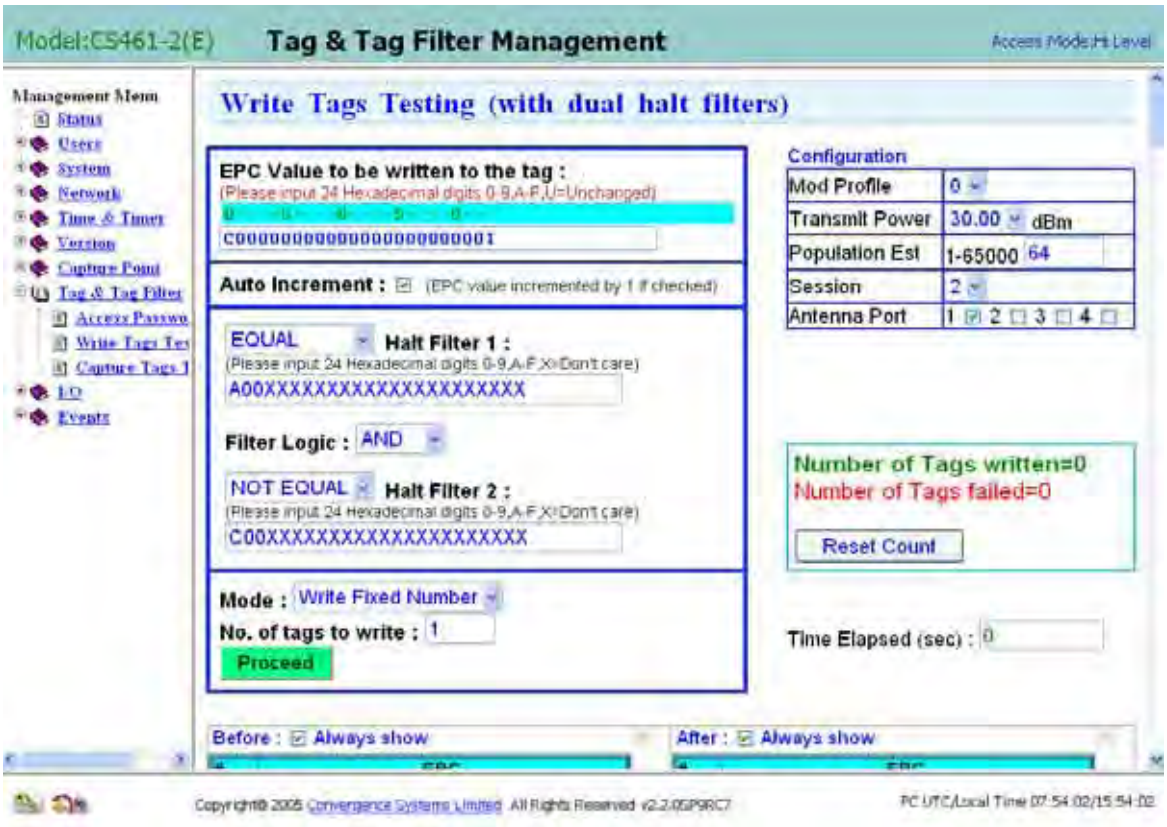


Figure 4-54 Write Tags Testing (with dual halt filters)

You can rewrite any tag's EPC ID regardless of its original EPC ID. Configure the reader settings, enter the new ID in "EPC Value" and then click the "Proceed" button to start writing. If you would like the EPC value to be incremented automatically for next write, please tick the "Auto Increment" box.

You can define which tag(s) to be written with new EPC ID. The steps for writing tags are:

1) Enter the original EPC ID of the tag(s) in the first "Halt Filter" field (note that you can enter a "X" character as a wildcard or don't care)
2) Enter the other ID on the second halt filter field. Note that this second halt filter is often used to prevent "cyclic write". **If you set this second halt filter to be the destination EPC ID field, then the write operation will not write tags that just have been written to. Please look carefully at the example above, and it is demonstrating a way that a tag will not be written more than once.**
3) Configure the reader settings on the upper right hand side column, making sure the port you are connecting the antennas to are ticked, and make sure the population estimation is within reasonable range.
4) Enter the new ID in the "EPC Value" field

5) Tick “Auto Increment” box if you want the EPC Value to be increased for each write
6) Select “Write Fixed Number” in “Mode” if you want to write a fixed number of tags; or click “Write Many” button if you want to write unlimited number of tags
7) Down below, one can check the original ID on the left column and the corresponding written ID on the right column. If the write is unsuccessful the right hand side will be “Error”. Note that this same tag that was unsuccessfully written can be found again later and successfully written. So on the left column the same ID may be repeated. If during the second time the tag is successfully written then the “Error” message would not appear on the corresponding row on the right hand side column.

The total number of successful writes and failed writes are shown on the box in the middle part of right hand side. Note that if there is no mechanism to stop cyclic write than these number will keep on accumulating as the tag is written over and over again.

Write Tags Testing (all banks) – Hex based

This is a more complicated page allowing the user to write information to Bank 1 and Bank 3. This write page allows writing based on a hex based unit, i.e. with 4 bits as the unit. So if the user only writes at boundaries definable by 4 bits or 8 bits, or any multiple of 4 bits, then the user can use this page. The user can also choose to lock the memory bank afterward, or unlock it afterward. Whether locking or unlocking afterward, the Access Password must be enabled and the correct password punched in. Just like other write tag pages, the user can choose to add halt filters (up to two) to select certain tags to write to.

Copyright © Convergence Systems Limited, All Rights Reserved

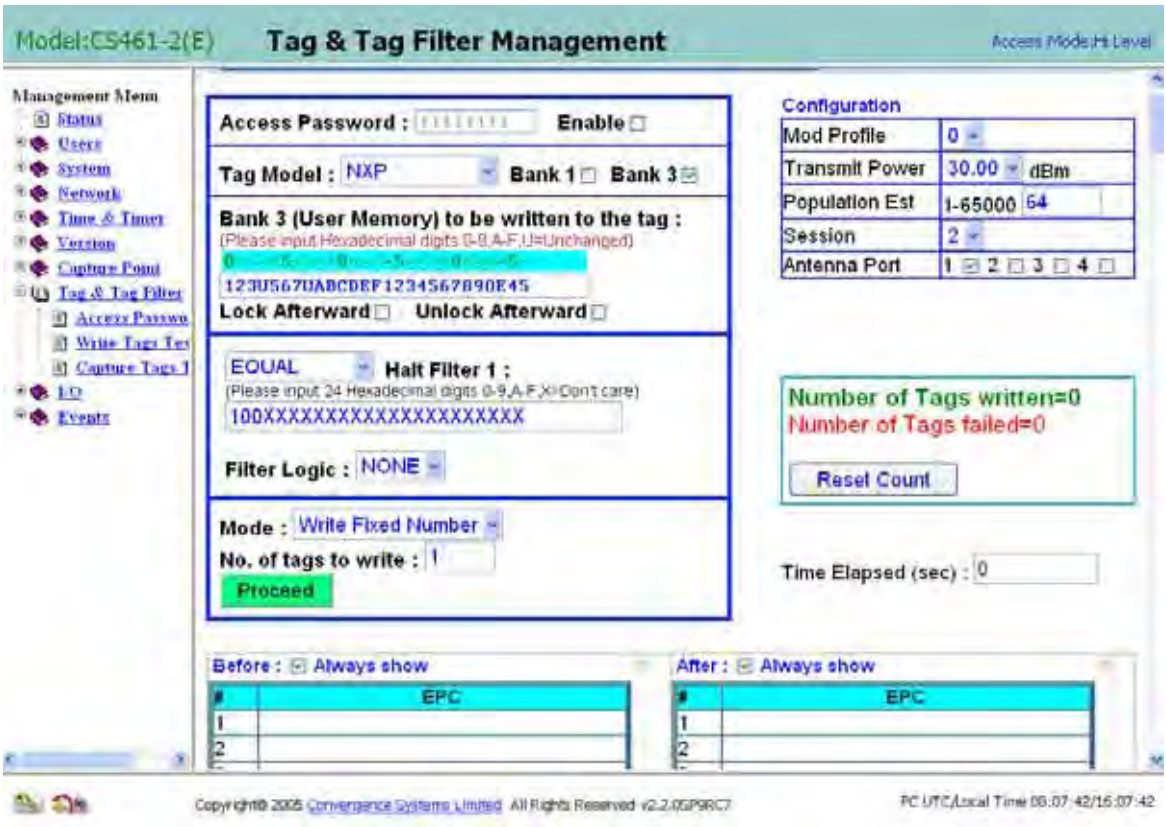


Figure 4-55 Write Tags Testing (all banks) – Hex based

4.9.5 Capture Tags Testing

In “Capture Tags Testing” page, there are a number of different pages for user to do tag capturing (reading):

Figure 4-56 Capture Tags Testing

Capture Tags (Time Window Mode, Event Driven) - EPC

One can monitor the ID of the tags being captured due to operation of an event (autonomous event). Click the “Show Tag” button to start capturing tags. Note that the tag that was seen before but no longer seen is highlighted in red. So the rows with transparent background are the tags still being successfully read by the reader.

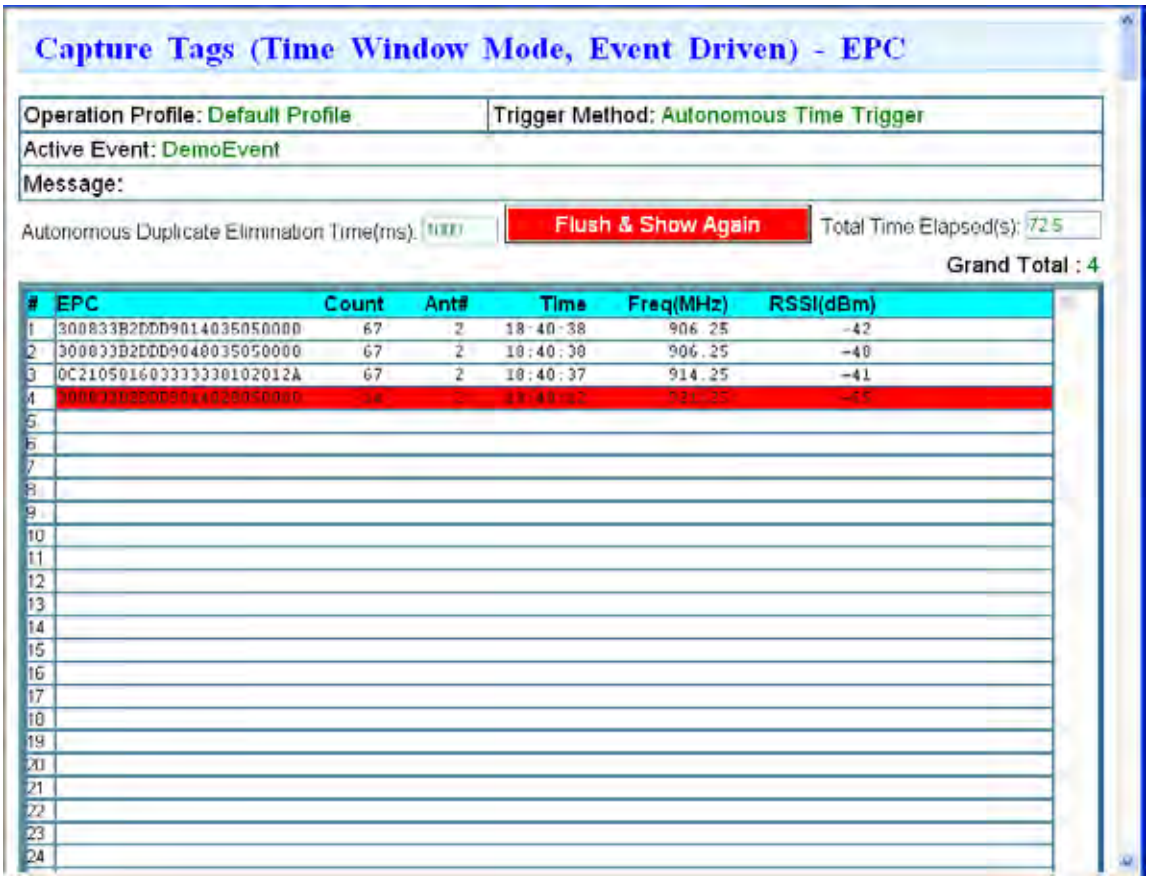


Figure 4-57 Capture Tags (Time Window Mode, Event Driver) - EPC

Capture Tags (Time Window Mode) – 2 Banks

This page allows you to read information from Bank 01, 10 and 11 (Bank 1, 2 and 3). Bank 01 contains CRC bits (4 hex, 16 bits), Protocol Control (PC) bits (4 hex, 16 bits), EPC ID (24 hex characters, 96 bits). Bank 10 contains TID or UID bits, for EPC format, it is 8 hex characters, 32 bits; for ISO format, it is 16 hex characters, 64 bits. Bank 11 contains User Memory bits, and in this particular tag, it is 56 hex characters, 224 bits. Note that you have to select the manufacturer of the tag on the second line. Also, which bank you want to read must also be selected. For Bank 1, it is always shown, and the 4 green hexadecimals in the beginning are the protocol control bits.

Copyright © Convergence Systems Limited, All Rights Reserved

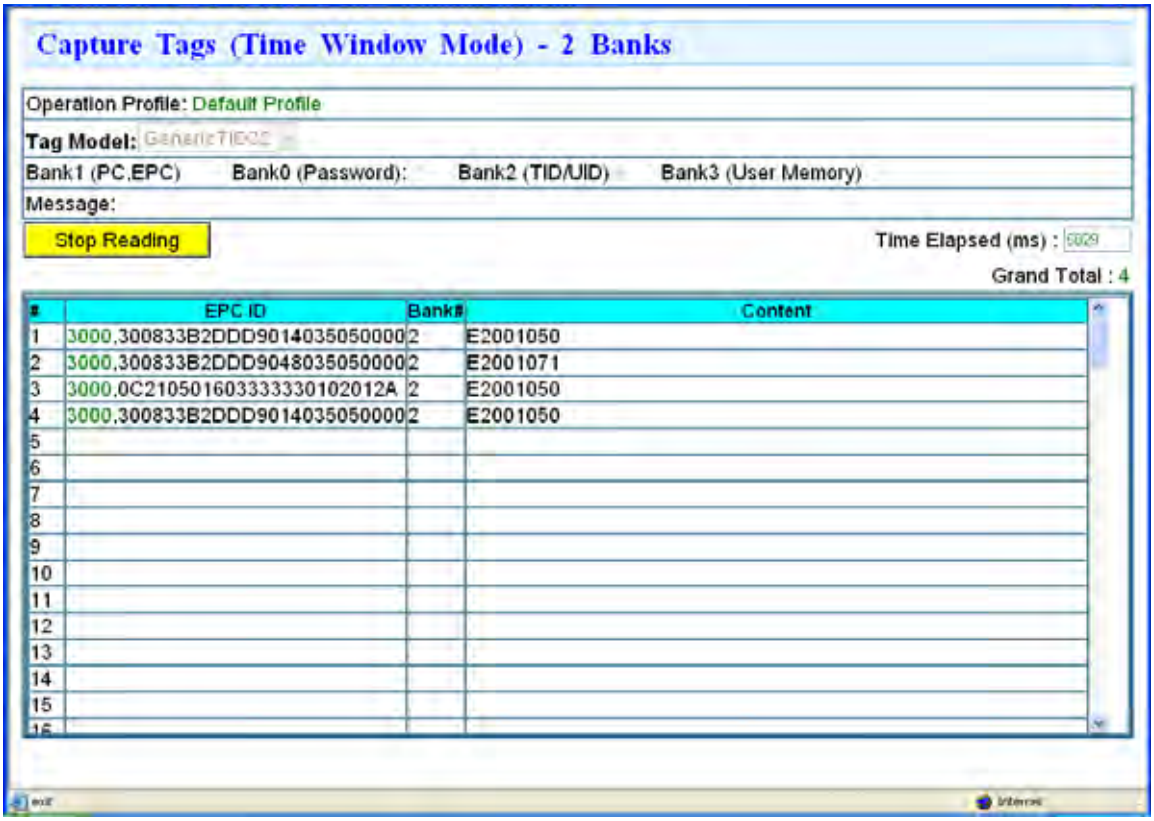


Figure 4-58 Capture Tags (Time Window Mode) – 2 Banks

Copyright © Convergence Systems Limited, All Rights Reserved

4.10 I/O Management

The “IO Management” page allows users to define various ports of input and output, giving them logical name for subsequent uses in event and other management. There are four inputs and eight outputs.

For the details of controlling IO port by High-level HTTP API command, please refer to the Programmer’s Manual.

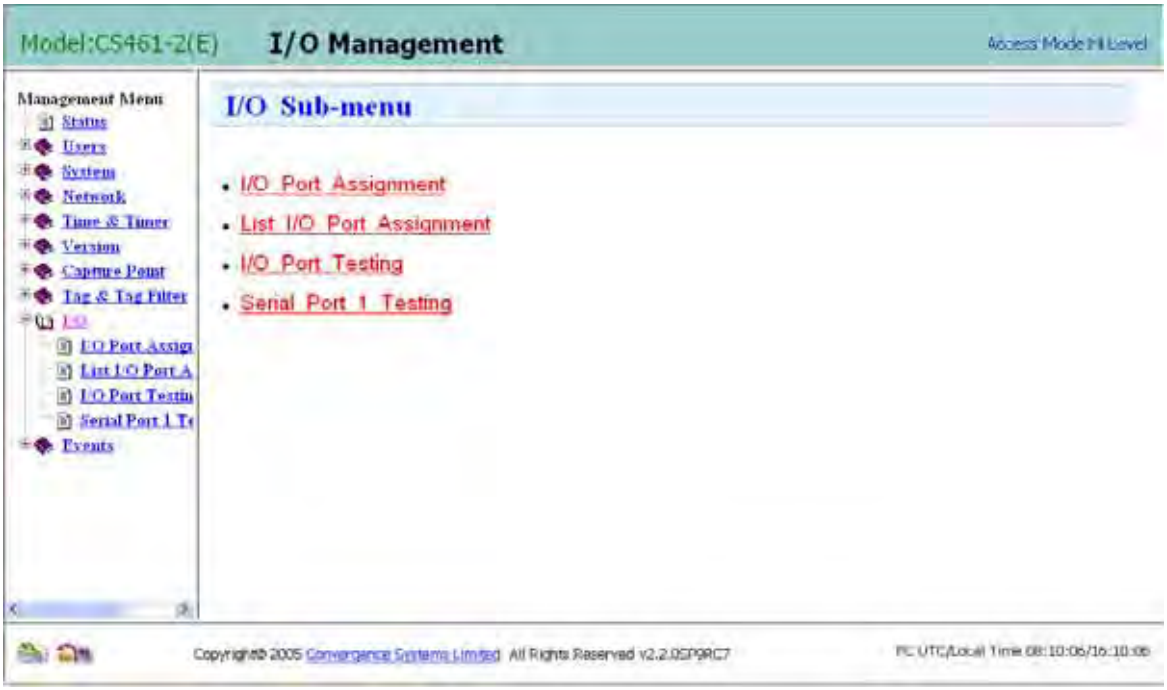


Figure 4-59 I/O Management

4.10.1 I/O Port Assignment

This page allows one to assign parameter to each I/O port:

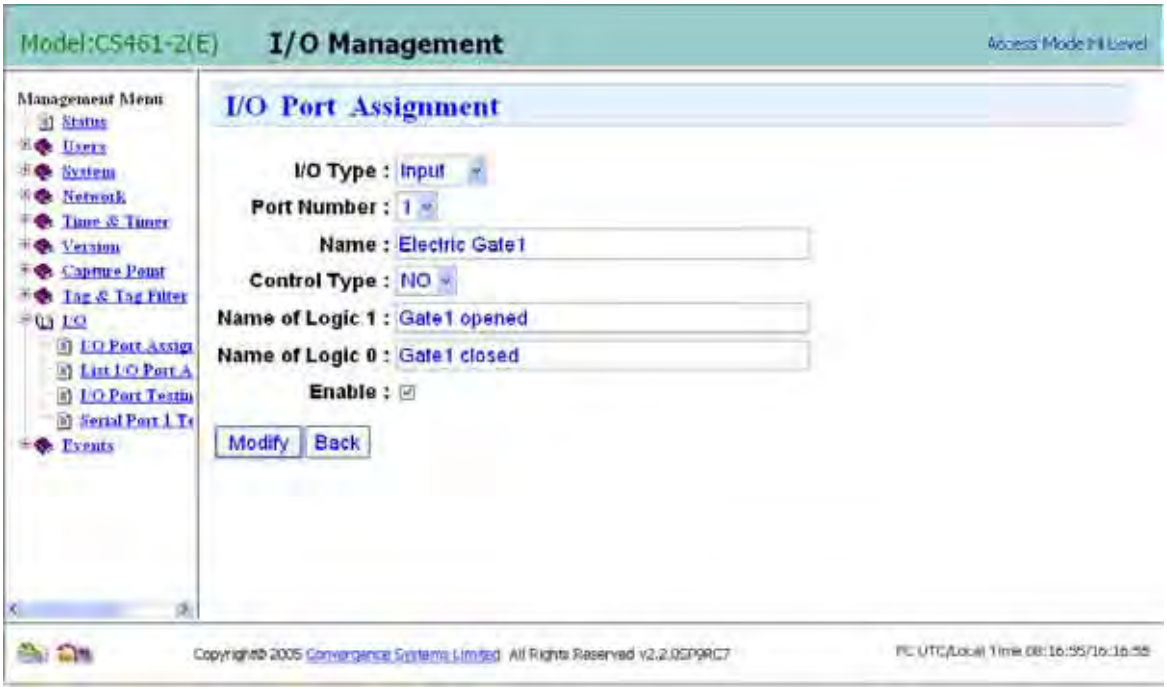


Figure 4-60 I/O Port Assignment

4.10.2 List I/O Port Assignment

Here is the “IO Port Assignment List” page:

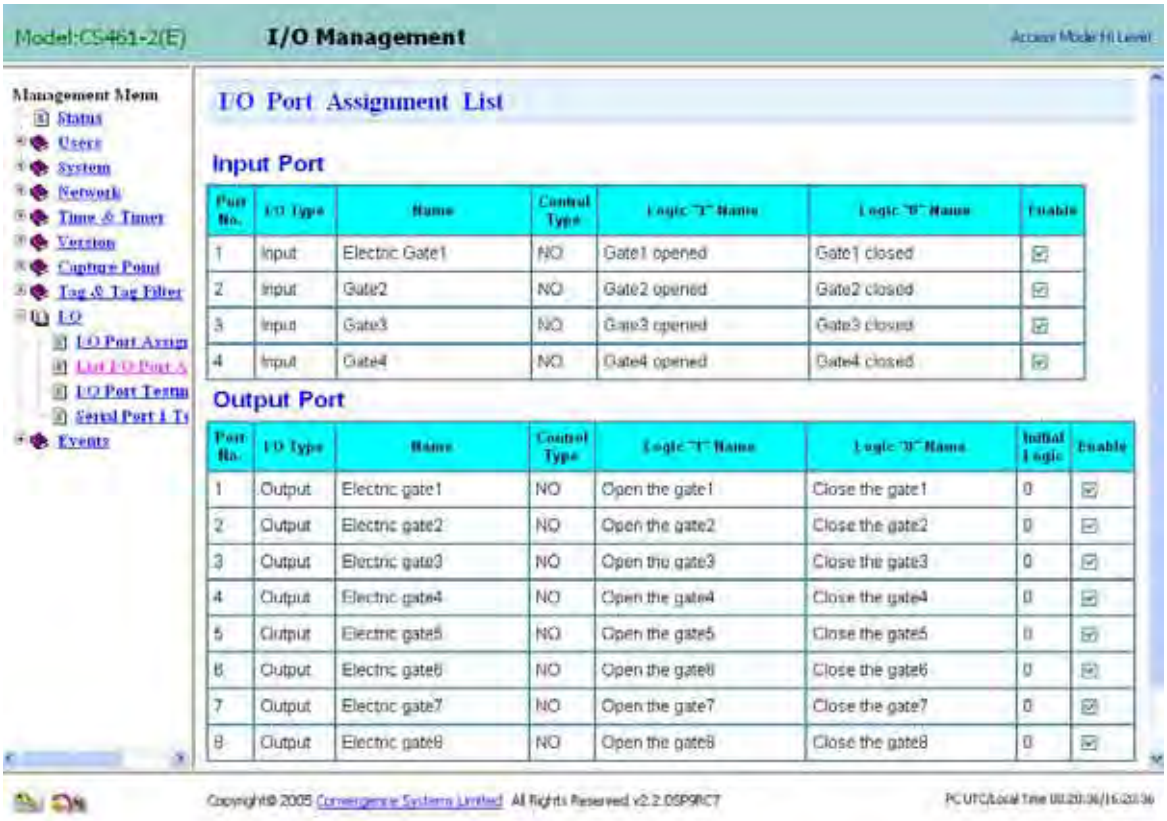


Figure 4-61 List I/O Port Assignment

4.10.3 I/O Port Testing

To test the I/O port, login is required:

Login: test engineer
Password: cnernd

Figure 4-62 I/O Port Testing - Login

Below is the “IO Port Testing” page, it allows one to look at sensor input (by pressing the “Update” button), and control the outputs for system testing.

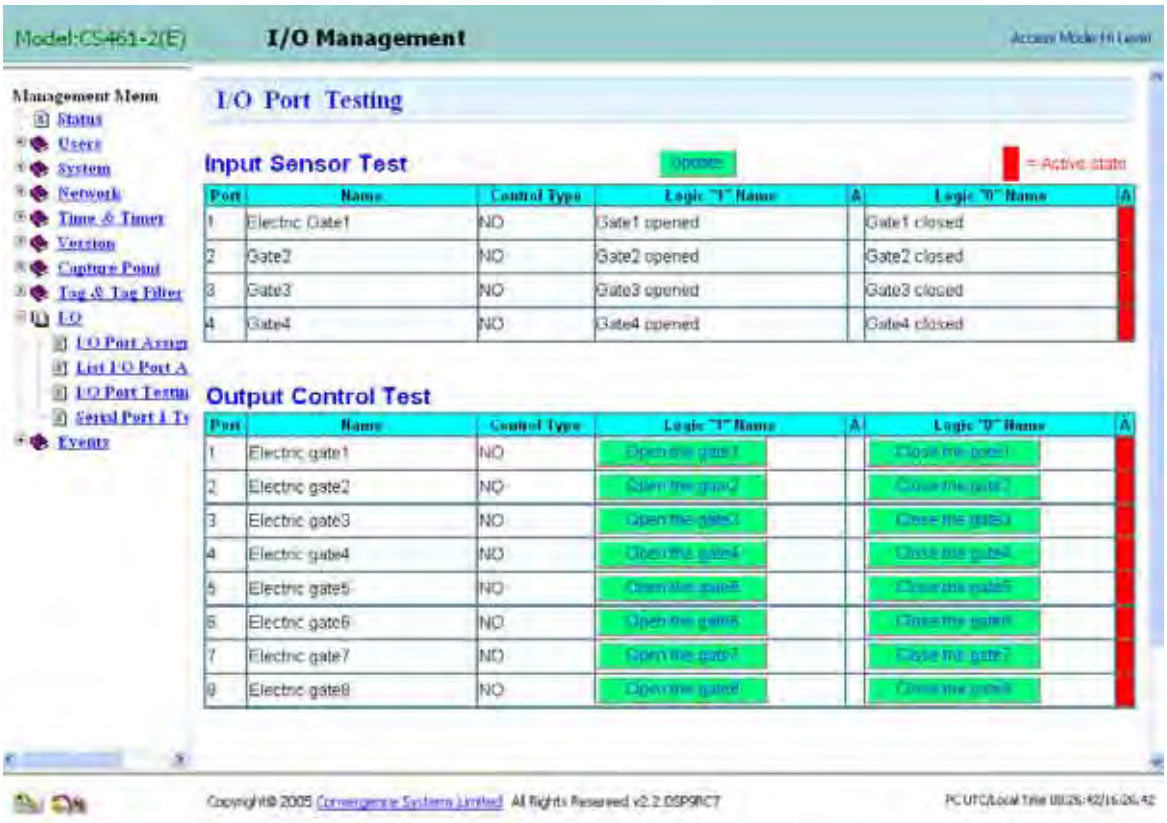


Figure 4-63 I/O Port Testing

4.10.4 Serial Port 1 Testing

In “Serial Port 1 Testing” page, you can test the serial communication between the reader IO port and the external device.

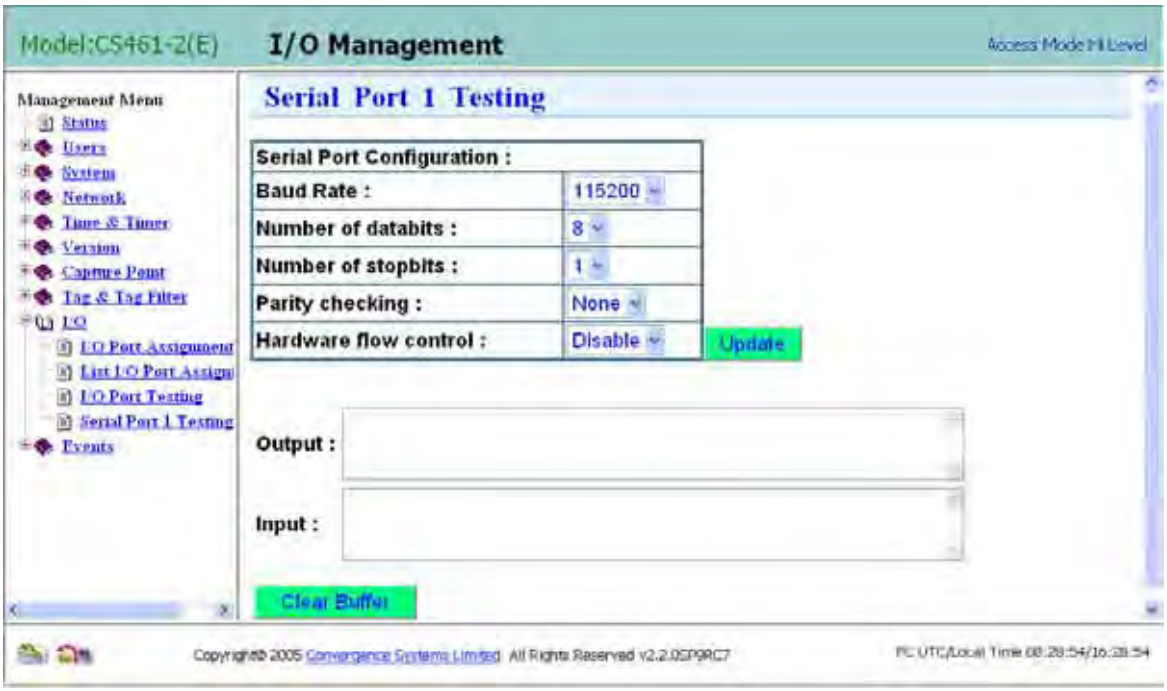


Figure 4-64 Serial Port 1 Testing

4.11 Event Management

Event is the most important part of the reader configuration. By setting it intelligently, one can handle many business applications autonomously with no interactive computation requirements needed from remote server. To create and enable an event, one needs to set up triggers, resultant actions, and then use that to assemble events. Once an event is created, the reader would run according to it continuously, and will continue even if the reader is powered down and up (rebooted). In other words, the reader is running autonomously.

Figure 4-65 Event Management

4.11.1 Event

Here is the “Event” submenu:

Figure 4-66 Event Management

Add Event

Below is the “Add Event” page:

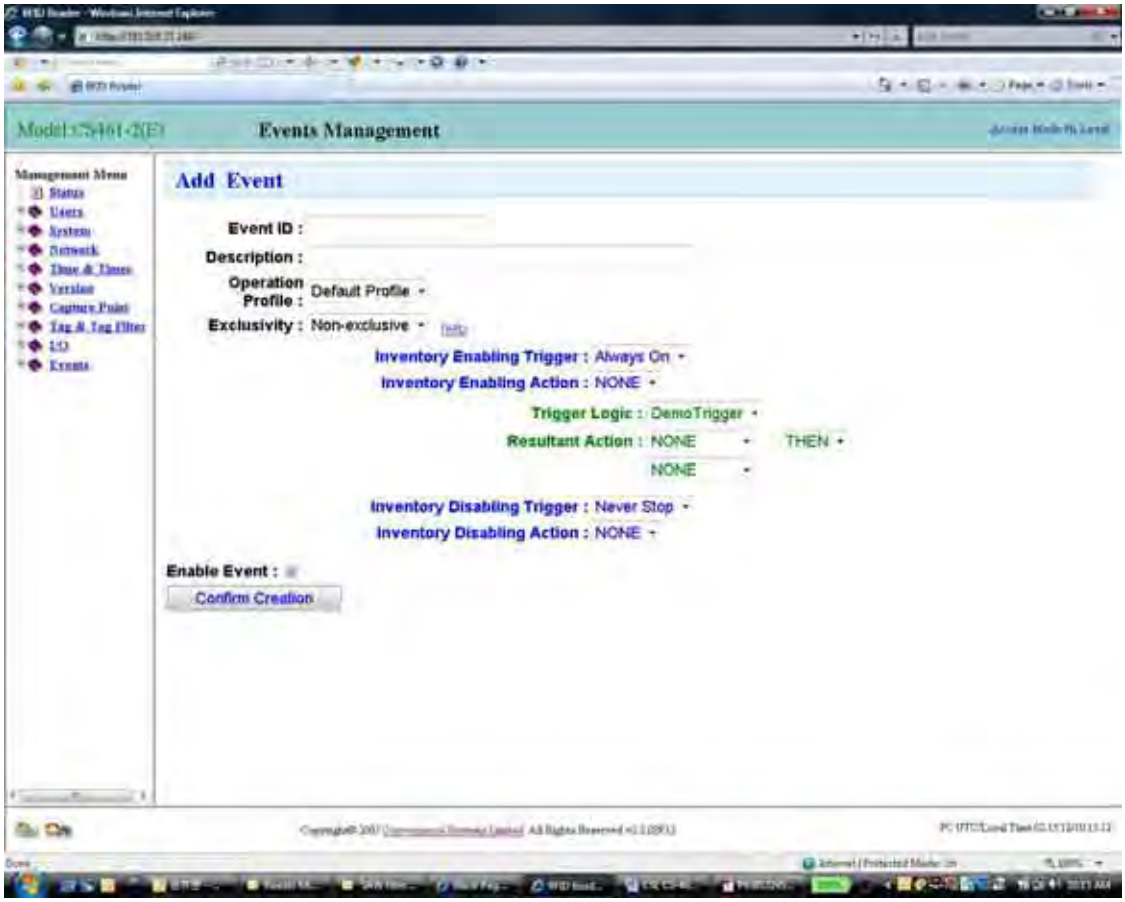


Figure 4-67 Add Event

One has to input a name for each event.

One should select the operation profile for the event. This operation profile is defined in the "System" page "Operation Profile" submenu.

The Inventory Enabling Trigger is the initial trigger that turns on the RF power of the reader to start doing inventory. This trigger can be set to "Always On" and then the reader will do inventory the moment the reader is powered on. Note that this trigger has to be defined in the Trigger page. Note that if one wants the reader to be always on, simply choose "Always On" in the Inventory Enabling Trigger entry.

The Inventory Enabling Action is the action that accompanies an inventory enabling trigger. For example, one may want to turn on a signal light when the inventory has started.

Once the inventory enabling cycle is entered, then the event engine would look for actual event triggers, and these triggers can be boolean operated together as defined in the entry "Trigger Logic". The Trigger Logic is a boolean combination of triggers that are defined in the

Copyright © Convergence Systems Limited, All Rights Reserved

“Trigger” page which will be described later.

When the Trigger Logic is satisfied, the event is established, and the resultant actions are defined in “Resultant Action” section. Again it can be a combination, sequential (THEN), of actions.

The overall inventory enabling cycle is ended based on the triggers defined in “Inventory Disabling Trigger” section. Sometimes this can be another Infrared at the exit of the reader read zone, or it can be defined as a period of time of no tag reads. If the user wants the reader to be always reading tags, then the selection “Never Stop” should be chosen here.

The Inventory Disabling Action is the action that accompanies the inventory disabling trigger. For example, one may want to turn off a signal light (that was turned on due to an inventory enabling action as described before) when the inventory is stopped.

Modify Event

To modify event, select the “Event ID”, modify the event and click “Confirm Modification”.

Copyright © Convergence Systems Limited, All Rights Reserved

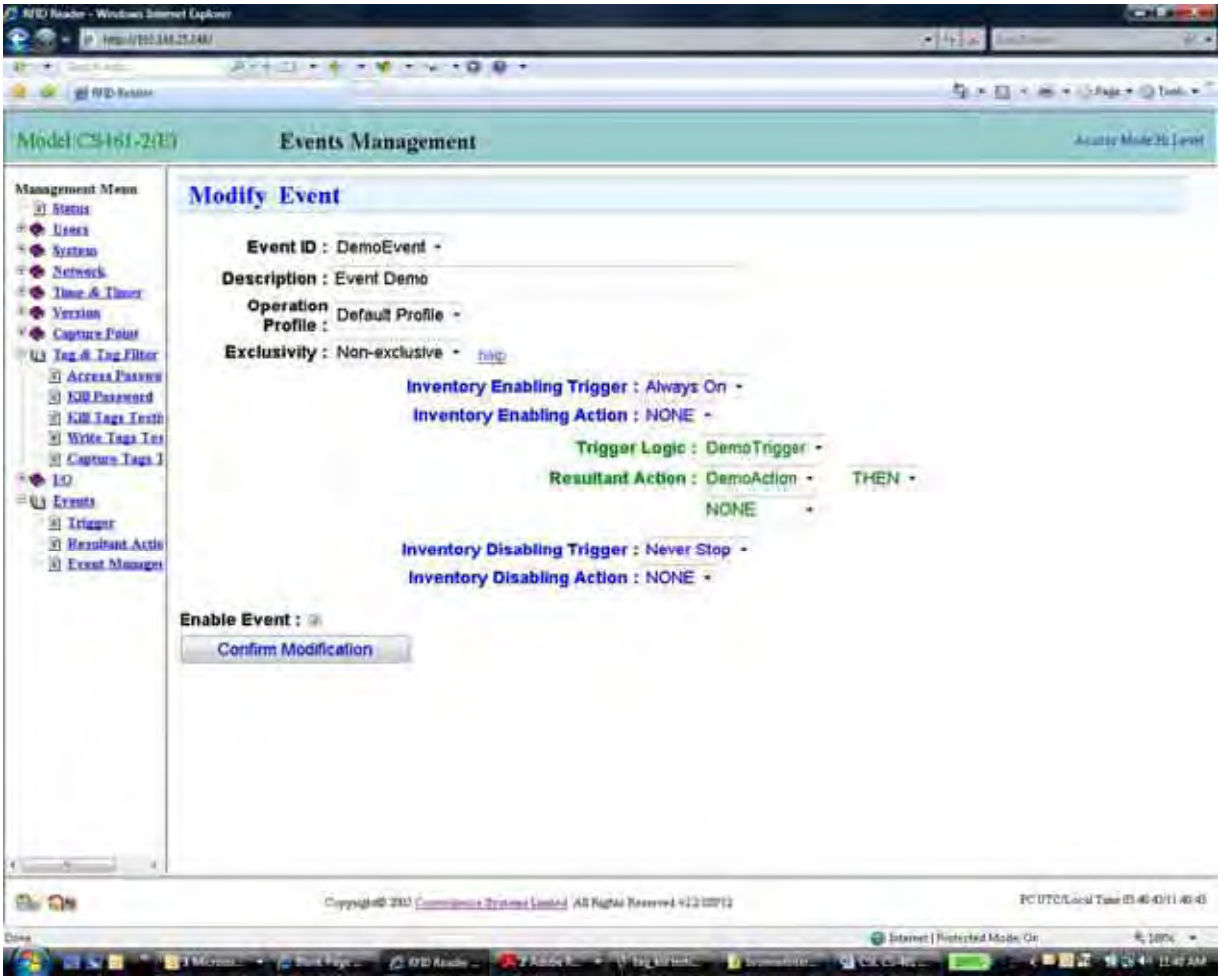


Figure 4-68 Modify Event

Enable/Disable Event

To enable/disable event, select/de-select the checkbox “Enable Event” and click “Modify”.

Figure 4-69 Enable/Disable Event

Delete Event

To delete event, select the “Event ID” and click “Delete”.

Figure 4-70 Delete Event

List Event

Below is the “List Event” page:

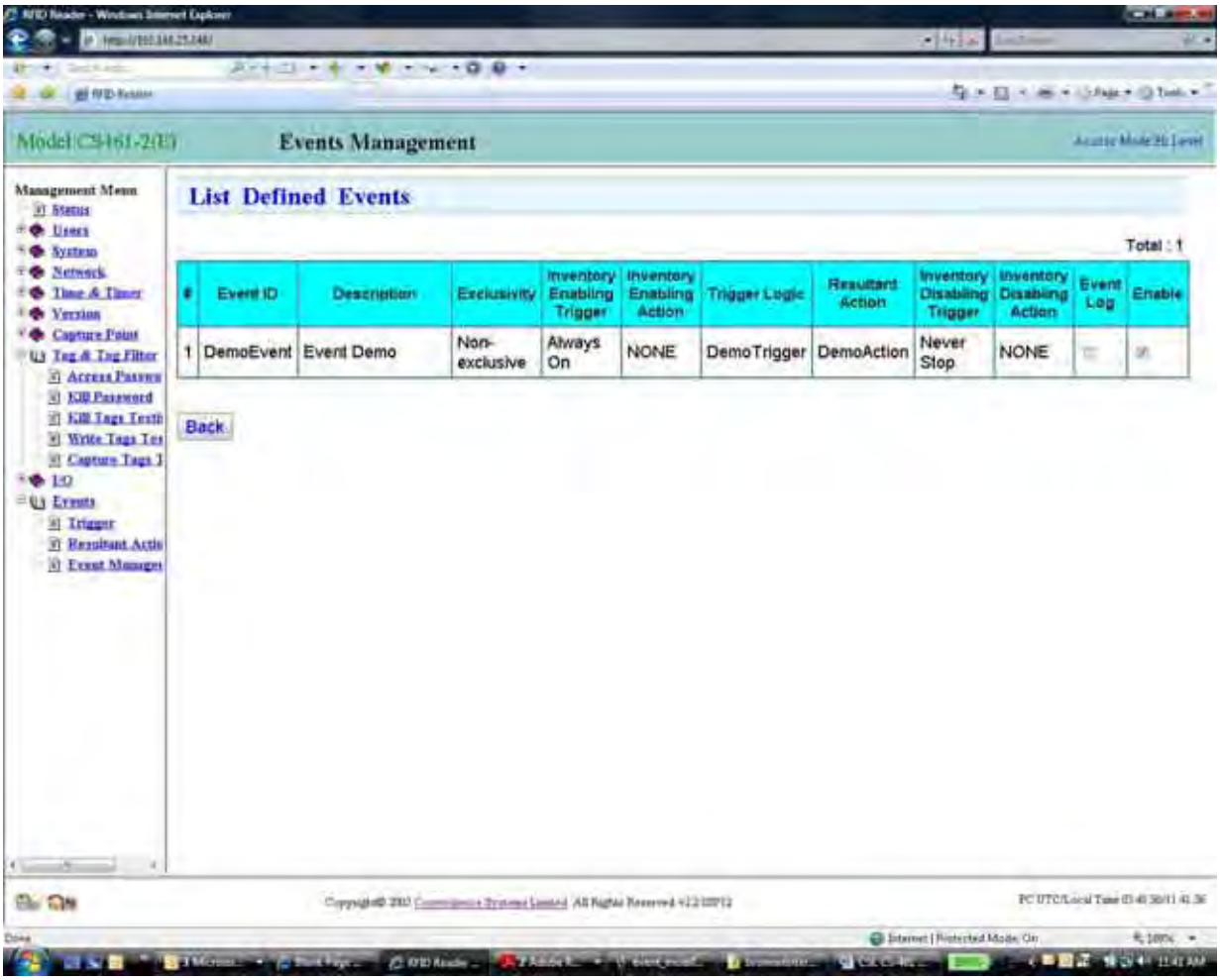


Figure 4-71 List Event

4.11.2 Trigger

A trigger is a stimulus that causes the reader to recognize it and do something about it.
The trigger is used in Inventory Enabling, Inventory Disabling, and of course inside the actual Event Triggering Logic section. Below is the “Trigger” submenu:

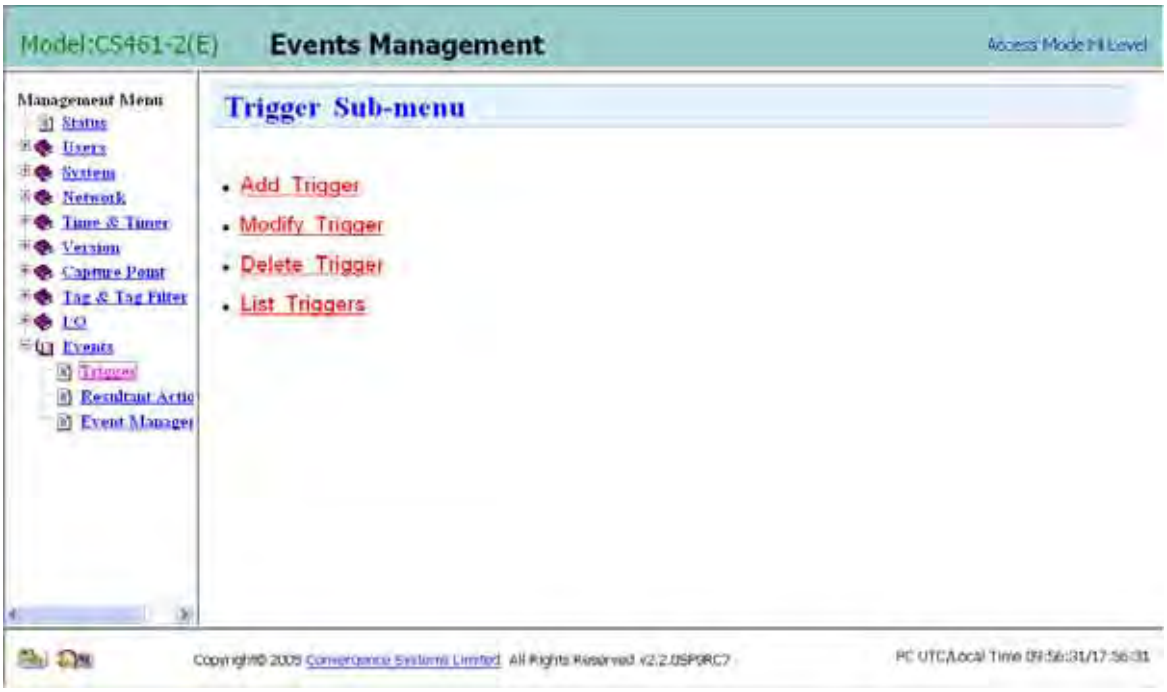


Figure 4-72 Trigger

Add Trigger

Below is the “Add Trigger” page:

Figure 4-73 Add Trigger

There are many different types of trigger which are describer as follows:

1. "Read Any Tags (any ID, 1 trigger per tag)" would look at tags coming in to the four antenna ports (or capture points), the ones being ticked here would be selected, and will generate 1 trigger per tag (different ID) notification. Note that in Time Windowed Mode there is a duplicate elimination action within each time window, and for the same ID within that window, it will only be recorded once into the buffer. Hence for each different ID within that duplicate elimination time it will generate an event.
2. "Input Sensor State" would look at the state (high or low) of one of the general purpose IO input.
3. "No Tag Read in Specified Time Span" would check if for a specified time read, no tag passes through the reader read zone.
4. "No Tag Read during Inventory Enabling Cycle" would check if within an inventory enabling cycle, no tag passes through the reader read zone.
5. "Read Any Tags (any ID, 1 trigger on first tag of inventory enabling cycle)" would look at any tag coming in during the inventory enabling cycle, and would trigger the onset of an event on the first such tag. Any subsequently incoming tag of other ID would not cause additional event triggering.
6. "Read Any Tags (any ID, 1 trigger at the end of inventory enabling cycle)" does the same thing as 5 but only send out the event at the end of the inventory enabling cycle.

Copyright © Convergence Systems Limited, All Rights Reserved

For “read any tags” trigger, the user also has to specify which antenna port or capture point it is collecting the tags from. To choose it, just tick the box on the left of each entry.

Figure 4-74 Add Trigger

Modify Trigger

To modify trigger, select the “Trigger ID”, modify the trigger and click “Modify”.

Figure 4-75 Modify Trigger

Delete Trigger

To delete trigger, select the “Trigger ID” and click “Delete”.

Figure 4-76 Delete Trigger

List Trigger

Below is the “List Trigger” page.

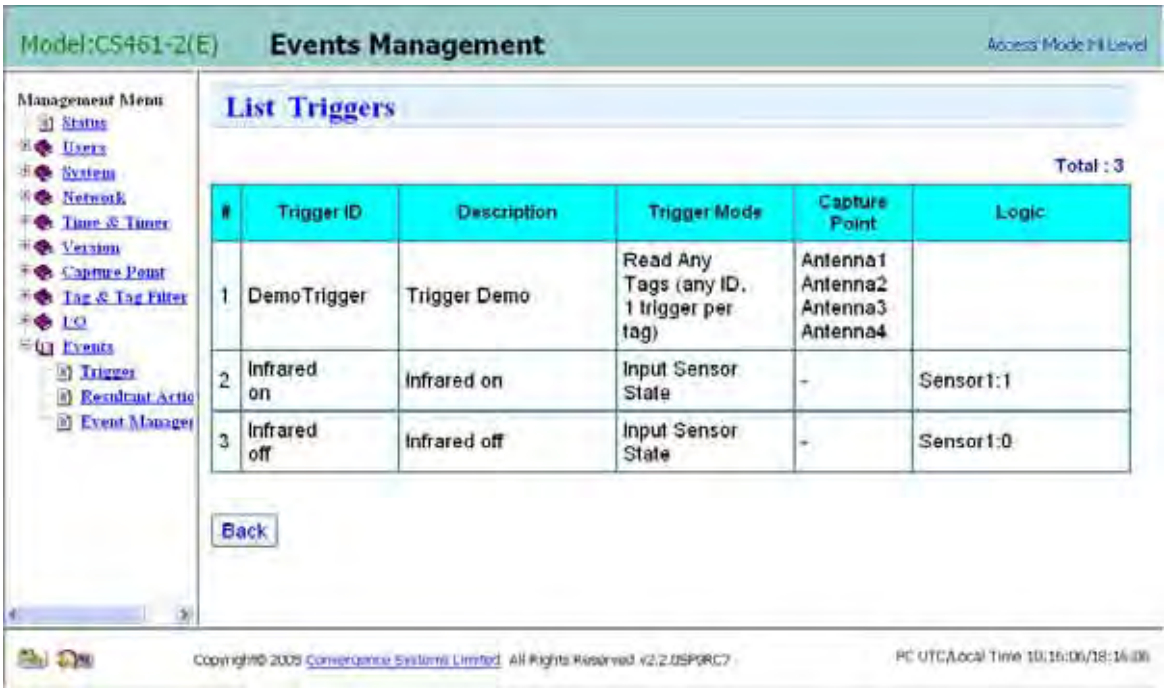


Figure 4-77 List Trigger

4.11.3 Resultant Action

The “Resultant Action” pages define the resultant action that will be enforced when an event logic is established. Below is the “Resultant Action” submenu:

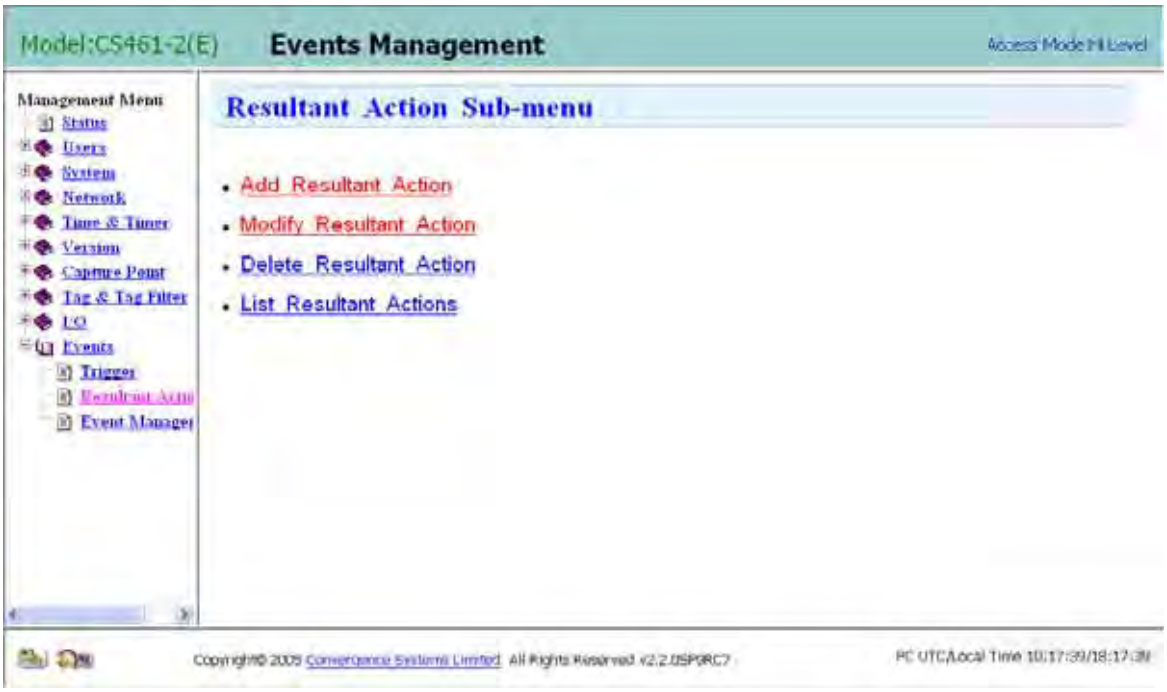


Figure 4-78 Resultant Action

Add Resultant Action

There are 4 types of possible action:

1. Do Nothing (Only Show on Screen) – here nothing is affected, except the tags collected can be shown on browser screen. Note that there are APIs that can collect the tag IDs or information on demand from the remote server. So this is actually a polling mode in terms of collecting tag information.
2. Batch Alert to Server – here the collected tag information are sent to Server at the end of each duplicate elimination cycle (Time Window)
3. Instant Alert to Server – here the collected tag information are sent to Server immediately as it is read.
4. Output Port – here the General Purpose IO output port would be controlled to have certain level change or pulse or even pulse train.

Figure 4-79 Add Resultant Action

If one selects "Batch Alert to Server" or the "Instant Alert to Server", then one has to also select the Server ID, which is defined in the Trusted Server page of the Network page. The user also has to select the Report ID that describes the format of the report. The user can also input the idle time (in second) to close the socket.

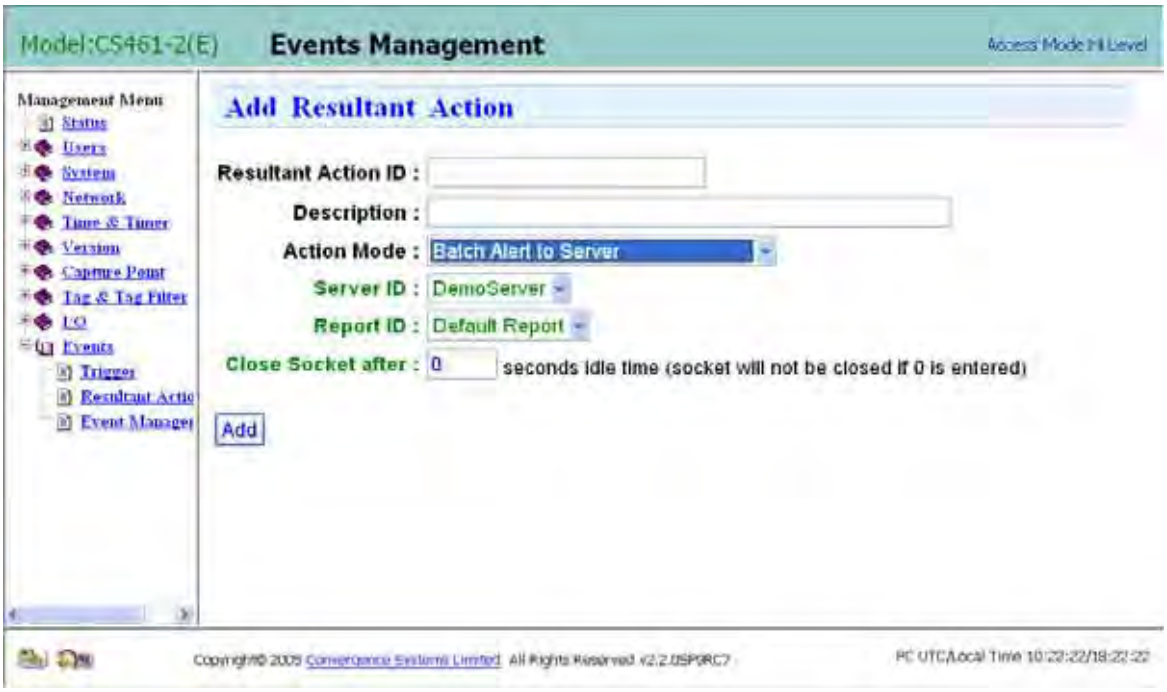


Figure 4-80 Add Resultant Action (Cont'd)

If one selects "Output Port" then one has to input few more fields. The user has to select the Port Number, 1 to 8. The Output Logic has to be selected, which can be either Low, High, or Pulse.

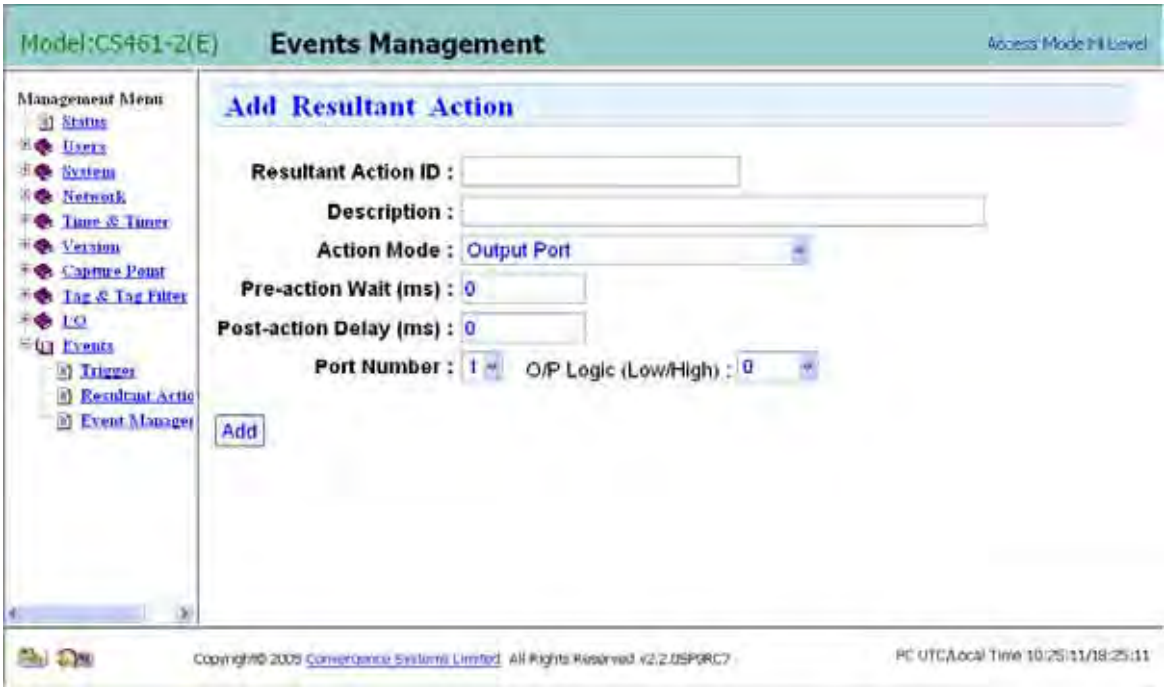


Figure 4-81 Add Resultant Action (Cont'd)

After that, the user has to select the output, which can either be a level, or a pulse. If it is a

pulse that the user has chosen, there are additional information needed, including Pulse Logic, Pulse Mode, Pulse Width, Duty Cycle, Duration, etc. The Pulse Logic can either be Low High Low or High Low High. The Pulse Mode can either be One Shot Pulse, Impulse or Pulse Train.

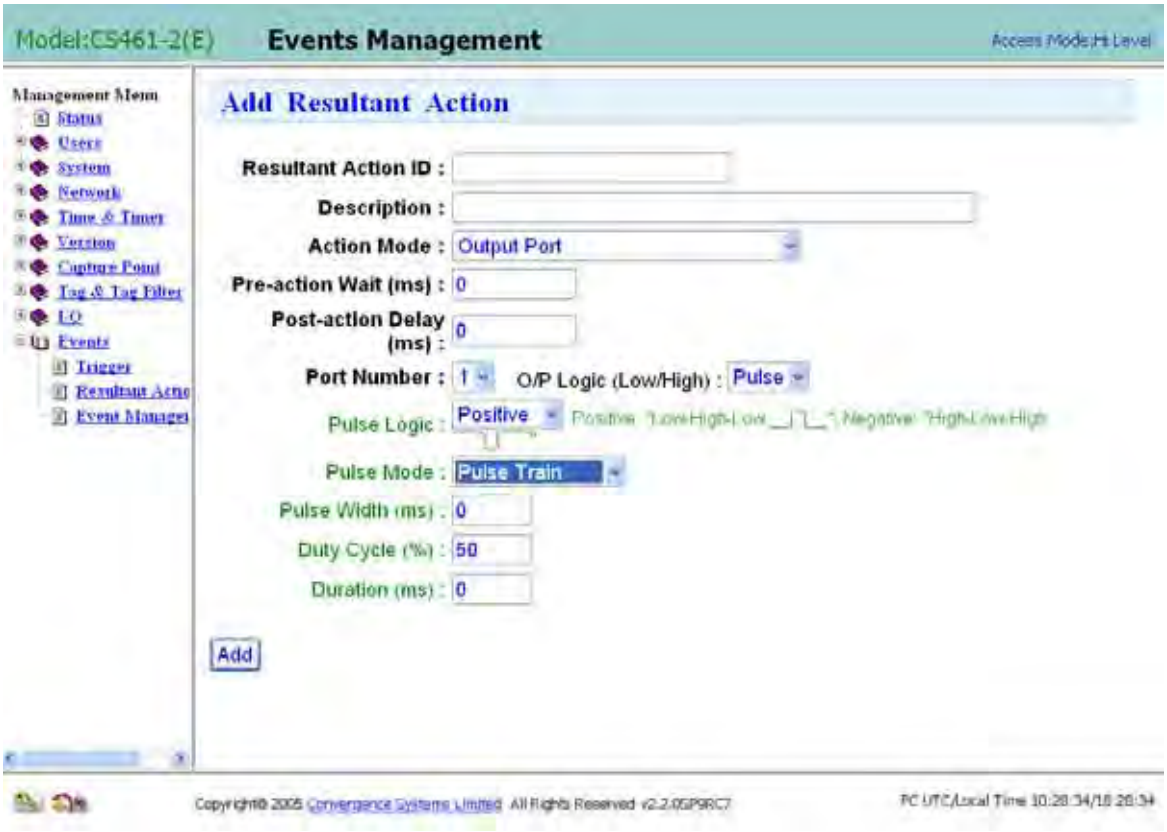


Figure 4-82 Add Resultant Action (Cont'd)

Modify Resultant Action

To modify resultant action, select the "Resultant Action ID", modify it and then click "Modify".

Figure 4-83 Modify Resultant Action

Delete Resultant Action

To delete resultant action, select the “Resultant Action ID” and click “Delete”.

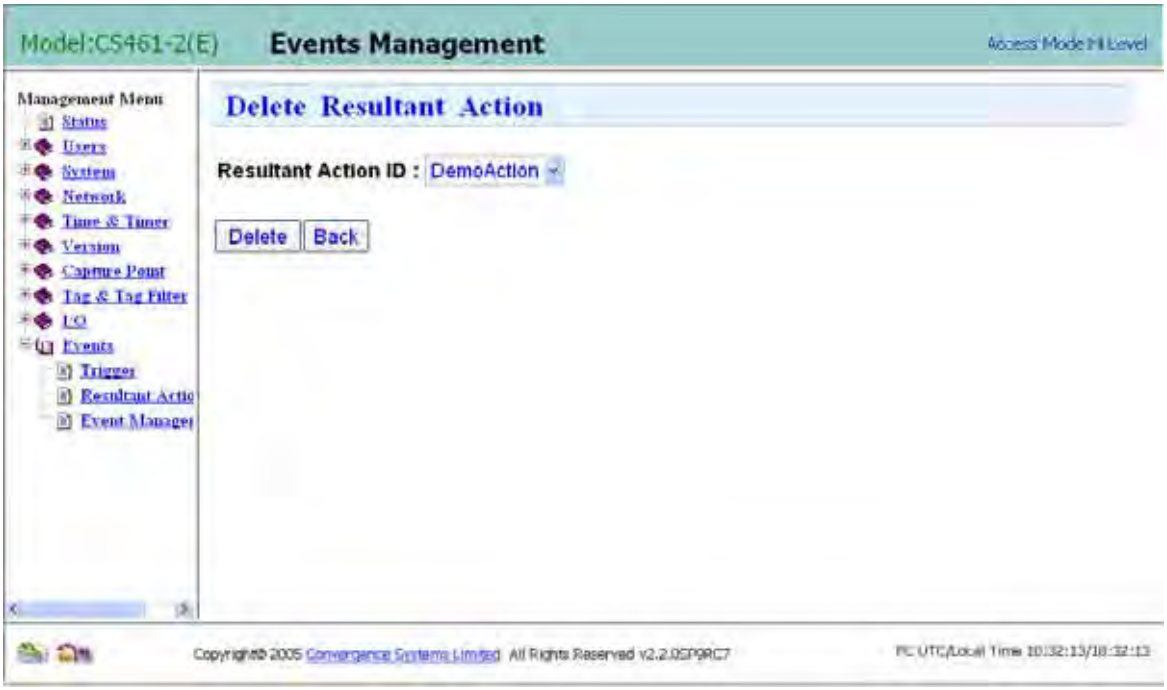


Figure 4-84 Delete Resultant Action

List Resultant Action

Below is the “List Resultant Action” action page.

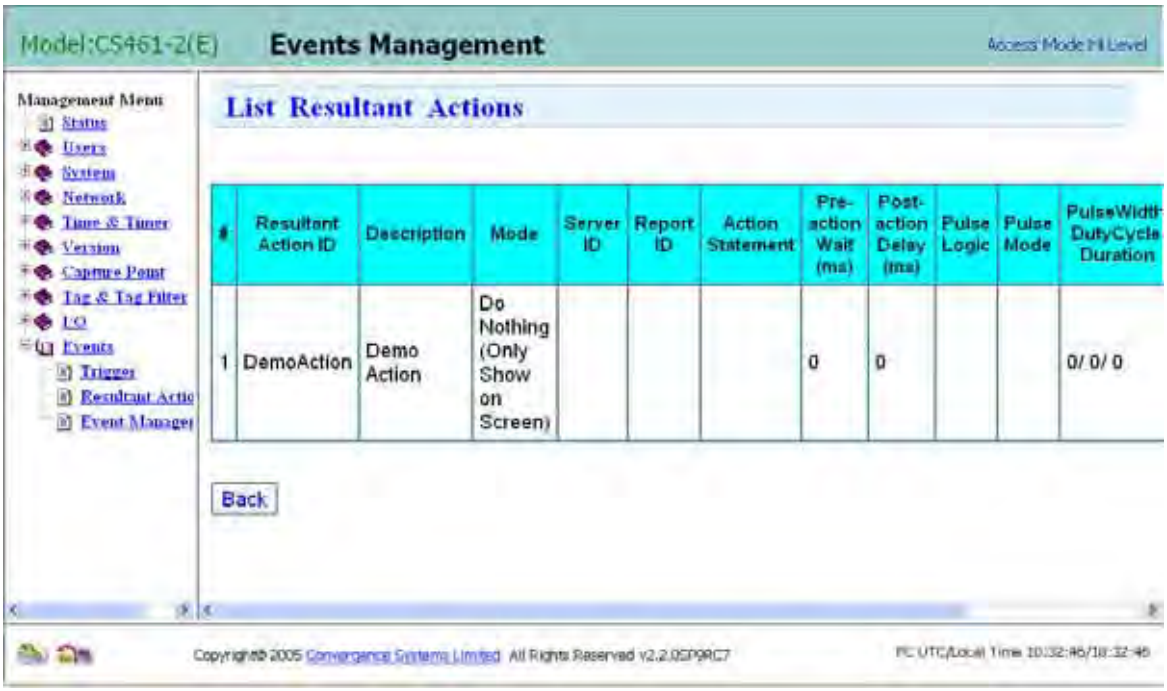


Figure 4-85 List Resultant Action

5 Programming Interface

The CSL CS-461 reader has two sets of Application Programming Interfaces (API):

1) CSL High Level API Manual
   - Make sure the reader is configured as “High Level API Mode” in Access Mode

2) CSL Low Level API Manual
   - Make sure the reader is configured as “Low Level API Mode” in Access Mode

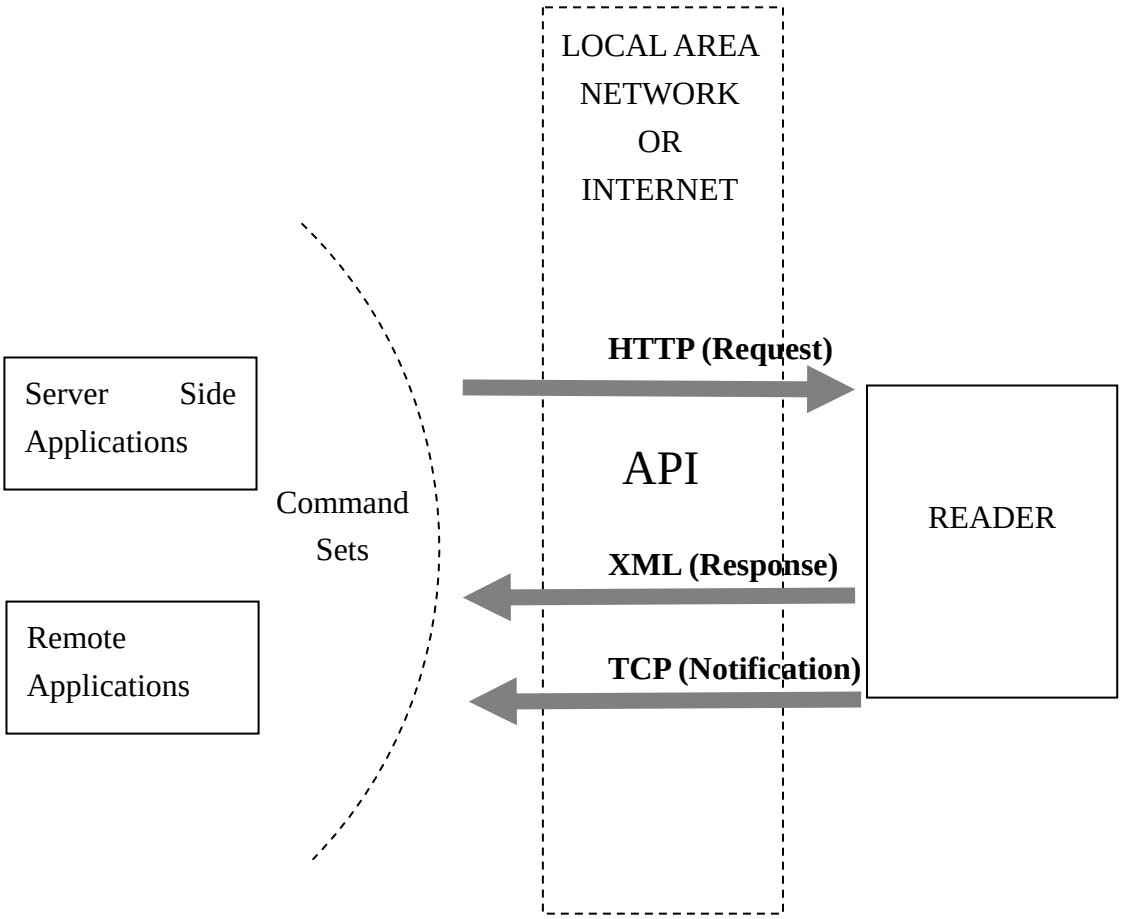
This section will cover a brief introduction as well as sample usage scenarios of the API. For details of programming methods, please refer to the series of Command Set Manuals.

Copyright © Convergence Systems Limited, All Rights Reserved

5.1 High Level API

The High Level API utilizes HTTP protocol for Requests and Responses. Server application sends HTTP requests to reader and then the reader responds to the requests in XML format. Operations such as parameter settings, parameter queries and read tags from current buffer are performed using such synchronous model.

In addition to HTTP, the High Level API also utilizes TCP protocol for Notifications of tag events and errors. These Notifications are sent from reader to trusted server asynchronously using TCP connection.



With the High Level API, operations such as read tags, write tags, duplication elimination in selected time window, event-driven output and input-trigger event are allowed.

Copyright © Convergence Systems Limited, All Rights Reserved

5.1.1 HTTP Request Query

The format of High-level HTTP API query from server to reader is as follows:

http://<*IP_address_of_Reader*>/API?session_id=<*session_id*>&command=<*command*>&

<*param1*>=<*param1_value*>

where:

Variable	Description
<IP_address_of_Reader>	IP address of the CS461 reader
<session_id>	The session ID obtained in the XML response message from reader after user login (not necessary for some commands, e.g. login)
<command>	High-level API command
<param1>	Setting parameter for the corresponding command. It can be optional or more than one parameter
<param1_value>	Value for the corresponding parameter setting

5.1.2 XML Response

The response of API is an XML object embedded in the HTTP response body with the format as follows:

```
<?xml version="1.0" ?>
<CSL>
    <Command>command1</Command>
    <Ack>ack_value</Ack>
    <Param1="param1_value" Param2="param2_value" />
</CSL>
```

Copyright © Convergence Systems Limited, All Rights Reserved

5.1.3 TCP Notification

If an event has a trusted server defined, tag data will be sent to the trusted server by TCP protocol.

Example 1: Read Tag

```
cmd=evtNtf&evt_id=POS Event&src_ip=192.168.25.248
&ant=Ant1&cp_id=Capture Point 2&idx=A1&tag_id=
300833B2DDD903C035055A92&rssi=-35&time=1159526240\n
```

Example 2: Read Tag Failover

```
cmd=blogNtf&evt_id=POS Event&src_ip=192.168.25.248
&ant=Ant1&cp_id=Capture Point 2 &idx=C1&tag_id=
300833B2DDD903C035055A92&rssi=-35&time=1159526240\n
```

Please note that the <\n> means new line character. It indicates the end of every notification packet.

5.1.4 Typical Program Flow

A typical program interfaces with the reader using High Level API has the following flow:

Copyright © Convergence Systems Limited, All Rights Reserved

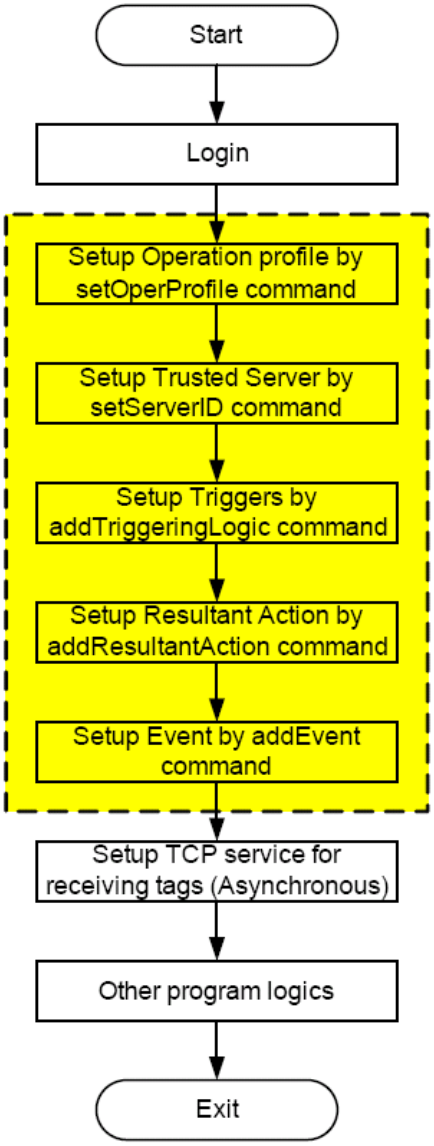


Figure 5-1 Typical Program Flow Using High Level API

Before accessing the reader, users have to login.

Example: Login

1. HTTP query string

```
http://192.168.25.208/API?command=login&username=root&password=csl2006
```

2. XML object in HTTP response

```
<?xml version="1.0"?>
<CSL>
    <Command>login</Command>
    <Ack>OK: session_id=324ab688</Ack>
</CSL>
```

Copyright © Convergence Systems Limited, All Rights Reserved

A session ID is returned which is required for all following commands to set up Operation Profile, Trusted Server, Trigger, Resultant Action and Event. Finally, a TCP service should be set up to receive asynchronous tag events from reader.

Copyright © Convergence Systems Limited, All Rights Reserved

5.1.5 Sample Usage Scenario – Access Control

Development Platform

The demo program is developed in Microsoft Visual Studio 2005 Professional Edition. It is written in Visual C# 2005 and utilizes Microsoft .Net Framework 2.0.

File List of Source Code

Filename	Type	Description
CS461_HL_API.cs	Source code	Class for High Level API. It implements API using .Net framework. It could be modified to become a class library and used in other projects.
CS461 Access Control.csproj	Project file	Project file used by VS2005
frmAccessLog.cs	Source code	Code for Access Log dialog
frmForm1.cs	Source code	Code for Main Screen
frmSettings.cs	Source code	Code for Settings dialogue
frmWelcome.cs	Source code	Code for Welcome dialogue
Program.cs	Source code	Code for application startup
Properties/AssemblyInfo.cs	Source code	Assembly information

All files with filename ended with “.Designer.cs” and “.resx” are generated by VS2005.

Demo Description

This is a simple demo illustrating how to read tags using Autonomous Time Trigger. The demo simulates an access control system to read a tag ID and then matches it with the data in the file “db.csv”. If the tag ID is found, information corresponding to the user is displayed. Otherwise, unauthorized access screen is shown.

1. **Reader connection**: Once the application starts, it connects to the reader device. In `frmForm1.cs`, an instance of `CS461_HL_API` is created. This instance is for connecting to the reader using High Level API:

```
CS461_HL_API reader = new CS461_HL_API();
```

Then, a method `loadUserSettings()` is invoked. This method retrieves reader information such as URI, login name and password from user settings:

Copyright © Convergence Systems Limited, All Rights Reserved

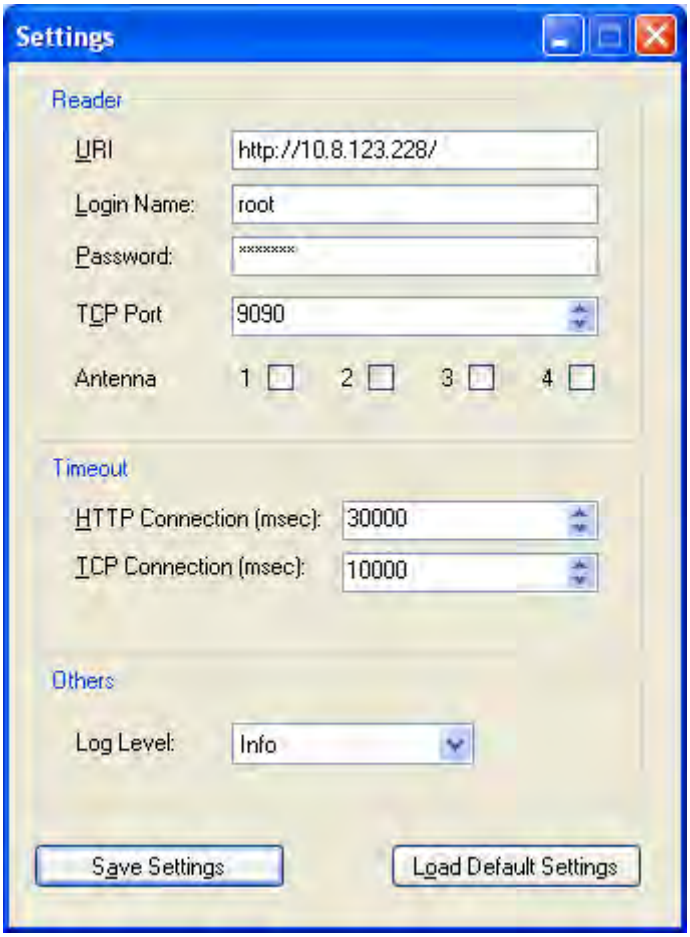


Figure 5-2

The information is set to the `CS461_HL_API` object:

```
reader.login_name =
(string)Application.UserAppDataRegistry.GetValue("LoginName", "root");
reader.login_password =
(string)Application.UserAppDataRegistry.GetValue("LoginPassword", "csl2006");
reader.http_timeout =
(int)Application.UserAppDataRegistry.GetValue("HttpTimeout", 30000);
reader.api_log_level =
reader.LogLevel((string)Application.UserAppDataRegistry.GetValue("LogLevel",
"Info"));
reader.setURI((string)Application.UserAppDataRegistry.GetValue("URI",
"http://192.168.25.208/"));
```

The reader is then connected by invoking the `connect()` method of `CS461_HL_API`:

```
reader.connect();
```

In the `connect()` method, `login()` method is invoked which in turn calls the “login” command of High Level API using HTTP by supplying the username and password as parameters:

```
string cmd = "login";
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
...
StringBuilder sbReq = new StringBuilder();
sbReq.Append(String.Format("{0}API?command={1}&username={2}&password={3}",
httpUri.AbsoluteUri, cmd,   LoginName, LoginPassword));
string resp = sendHTTPRequest(sbReq.ToString());
```

If login successful, the reader will return an ACK message as follows:

```
<?xml version="1.0" ?>
<CSL>
   <Command>login</Command>
   <Ack>OK: session_id=4c531266</Ack>
</CSL>
```

The `login()` method then retrieves the `session_id` as all commands afterward must contain this id to maintain the login session.

2. **Setup Trigger, Action and Event**: Once the reader is connected in `frmForm1.cs`, the method `setupReader()` is invoked. This method set up the Trigger, Action and Trigger required for the application.

   i) Disable all Events that are currently running on reader:

```
//Disable all events
System.Collections.ArrayList eventList;
eventList = reader.listEvent();
if (eventList != null)
{
   foreach (EVENT_INFO e in eventList)
   {
      reader.enableEvent(e.id, false);
   }
}
```

The above code segment first retrieves the Event list by invoking the `listEvent()` method of `CS461_HL_API`. This method in turn calls the "listEvent" command of the High Level API using HTTP:

```
string cmd = "listEvent";
...
StringBuilder sbReq = new StringBuilder();
sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));

string resp = sendHTTPRequest(sbReq.ToString());
```

Response of the "listEvent" command will be an XML containing the information of all Event settings current on the reader:

```
<?xml version="1.0" ?>
<CSL>
   <Command>listEvent</Command>
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
    <EventMode mode="0" />
    <AutomaticConfigure desc="automatic configure CSLEvent for DSPI" enable="true" />
    <EventList>
      <event desc="Event Demo" enable="false" event_id="DemoEvent" event_log="false"
inventoryDisablingTrigger="Never Stop" inventoryEnablingTrigger="Always On"
operProfile_id="Default Profile" resultant_action="DemoAction" triggering_logic="DemoTrigger"
/>
    </EventList>
</CSL>
```

The `listEvent()` method then parses the XML and return an ArrayList of `EVENT_INFO`. For each `EVENT_INFO`, the method `enableEvent()` of `CS461_HL_API` is invoked. This method will call the "enableEvent" command of High Level API using HTTP to disable the event:

```
public bool enableEvent(string id, bool enable)
{
  string cmd = "enableEvent";
  ...
  StringBuilder sbReq = new StringBuilder();

  sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
  sbReq.Append(String.Format("&event_id={0}&enable={1}",   id,   (enable)   ?
"true" : "false"));

  string resp = sendHTTPRequest(sbReq.ToString());
  ...
}
```

The reader will return an ACK message of the command as follows:

```
<?xml version="1.0" ?>
<CSL>
    <Command>enableEvent</Command>
    <Ack>OK: </Ack>
</CSL>
```

ii) Setup Operation Profile. The operation profile controls the behavior of the reader such as antenna used and the RF power. In this example, "Autonomous Time Trigger" is used. It allows duplicate elimination which prevents the same tag being sent more than once within the same time window (in this example, the time window is set to 1000 ms). Note that the parameters for operation profile are case sensitive:

```
//Setup Operation Profile
OPERATION_PROFILE profile = new OPERATION_PROFILE();

profile.profile_id = "Default Profile";
profile.profile_enable = true;
profile.modulation_profile = "Profile0";
profile.population = 5;
profile.session_no = 3;
profile.transmit_power = "20.00";
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
profile.window_time = 1000;
profile.capture_mode = "Time Window";
profile.ant1_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant1",
0) == 1) ? true : false;
profile.ant2_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant2",
0) == 1) ? true : false;
profile.ant3_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant3",
0) == 1) ? true : false;
profile.ant4_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant4",
0) == 1) ? true : false;
profile.trigger = "Autonomous Time Trigger";

if (reader.setOperProfile(profile) == false)
{
   tsslStatus.Text = "Fail to set operation profile";
   return false;
}
```

In the above code segment, the method setOperProfile() of CS461_HL_API is invoked which in turn calls the “setOperProfile” command using HTTP:

```
string cmd = "setOperProfile";
...
sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
         httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&profile_id={0}&captureMode={1}&duplicateElimina
         tionTime={2}", profile.profile_id, profile.capture_mode,
         profile.window_time));
sbReq.Append(String.Format("&modulationProfile={0}&populationEst={1}&session
         No={2}", profile.modulation_profile, profile.population,
         profile.session_no));
sbReq.Append(String.Format("&transmitPower={0}&antennaPort={1}&enable={2}",
         profile.transmit_power, antennaPort, enable));
sbReq.Append(String.Format("&triggerMethod={0}", profile.trigger));

string resp = sendHTTPRequest(sbReq.ToString());
```

iii) Setup Trusted Server. In order to receive event notification, the machine running the application must be set as the trusted server of the reader. This application will first try to create a trusted server. If it fails, properly means that a server with the same id already exists. It will try to modify the existing server before returning fail. In this example, the trusted server mode is set to “Listening Port on Server Side”. It means that the reader will try to connect to the IP and port provided when event occurs. Again, this value is case sensitive.

```
//Setup Trusted Server
SERVER_INFO svr = new SERVER_INFO();
svr.id = "Access Control Server";
svr.desc = "Access Control Server";
IPHostEntry he = Dns.GetHostEntry(System.Environment.MachineName);
svr.ip = he.AddressList[0].ToString();
svr.server_port = server.tcp_port.ToString();
svr.mode = "Listening Port on Server Side";
svr.enable = true;
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
if (reader.setServerID(svr) == false)
{
    if (reader.modServerID(svr) == false)
    {
        tsslStatus.Text = "Fail to set trusted server";
        return false;
    }
}
```

The method `setServerID()` is invoked which calls the "setServerID" command of High Level API using HTTP to create the trusted server:

```
string cmd = "setServerID";
...
StringBuilder sbReq = new StringBuilder();

string enable = "false";
if (svr.enable)
    enable = "true";

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&server_id={0}&desc={1}", svr.id, svr.desc));
sbReq.Append(String.Format("&server_ip={0}&server_port={1}&enable={2}",
svr.ip, svr.server_port, enable));
sbReq.Append(String.Format("&reader_ip={0}&mode={1}", svr.reader_port,
svr.mode));

string resp = sendHTTPRequest(sbReq.ToString());
```

iv) Setup Resultant Action. In this example, the action mode is set to "Batch Alert to Server" which means the reader will send the tag event report to the trusted servers in a batch at the end of the time window.

```
//Setup Resultant Action
reader.delResultantAction("Access Control Action");

RESULTANT_ACTION_INFO action = new RESULTANT_ACTION_INFO();
action.id = "Access Control Action";
action.desc = "Access Control Demo";
action.mode = "Batch Alert to Server";
action.server_id = svr.id;
action.report_id = "Default Report";

if (reader.addResultantAction(action) == false)
{
    tsslStatus.Text = "Fail to set resultant action";
    return false;
}
```

The method `addResultantAction()` is invoked which calls the "addResultantAction" command of High Level API using HTTP:

Copyright © Convergence Systems Limited, All Rights Reserved

```
string cmd = "addResultantAction";
...
StringBuilder sbReq = new StringBuilder();

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&action_id={0}&desc={1}&action_mode={2}",
info.id, info.desc, info.mode));
sbReq.Append(String.Format("&server_id={0}&report_id={1}", info.server_id,
info.report_id));

string resp = sendHTTPRequest(sbReq.ToString());
```

v) Setup Event. Add an event with "DemoTrigger" as the trigger logic and the action created in the previous step as the resultant action. The "DemoTrigger" used is pre-set which triggers event when any tag is read in any antenna.

```
//Setup Event
reader.delEvent("Access Control Event");

EVENT_INFO eventInfo = new EVENT_INFO();
eventInfo.id = "Access Control Event";
eventInfo.desc = "Access Control Demo";
eventInfo.profile = profile.profile_id;
eventInfo.trigger = "DemoTrigger";
eventInfo.action = action.id;
eventInfo.log = false;
eventInfo.enable = true;
eventInfo.enabling = "Always On";
eventInfo.disabling = "Never Stop";

if (reader.addEvent(eventInfo) == false)
{
    tsslStatus.Text = "Fail to set event";
    return false;
}
```

The method `addEvent()` is invoked which calls the "addEvent" command of High Level API using HTTP:

```
string cmd = "addEvent";
...
StringBuilder sbReq = new StringBuilder();
string eventEnable = "false";
if (info.enable)
    eventEnable = "true";
string eventLog = "false";
if (info.log)
    eventLog = "true";

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
          httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&event_id={0}&desc={1}&triggering_logic={2}&oper
          Profile_id={3}", info.id, info.desc, info.trigger, info.profile));
sbReq.Append(String.Format("&resultant_action={0}&event_log={1}&enable={2}",
          info.action, eventLog, eventEnable));
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
sbReq.Append(String.Format("&inventoryEnablingTrigger={0}&inventoryDisabling
         Trigger={1}", info.enabling, info.disabling));

string resp = sendHTTPRequest(sbReq.ToString());
```

3. **Start Trusted Server**: In frmForm1.cs, an instance of TrustedServer is created.

```
TrustedServer server = new TrustedServer();
```

In the method `loadUserSettings()`, the `TrustedServer` object is initialized:

```
server.tcp_port = (int)Application.UserAppDataRegistry.GetValue("TcpPort",
9090);
server.api_log_level = reader.api_log_level;
```

After configuring the reader in step 2, the trusted server should be started to receive event notification from the reader. It is done by invoking the `Start()` method of `TrustedServer`.

```
server.Start();
```

4. **Handle Tag Event**: An event handler `AccessControl_TagReceiveEvent` is added to the trusted server to handle tag events.

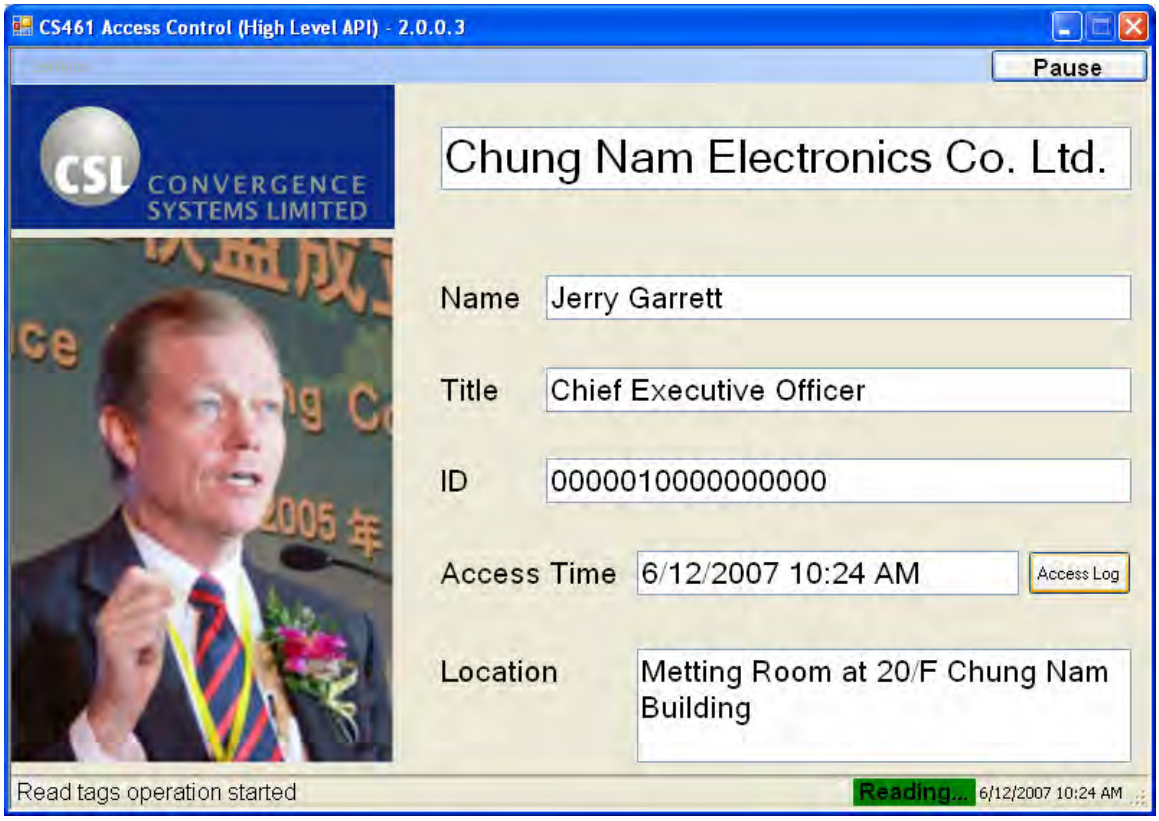
```
server.TagReceiveEvent += new
TagReceiveEventHandler(this.AccessControl_TagReceiveEvent);
```

When a tag event is received, the handler updates the information on screen according to the tag ID:

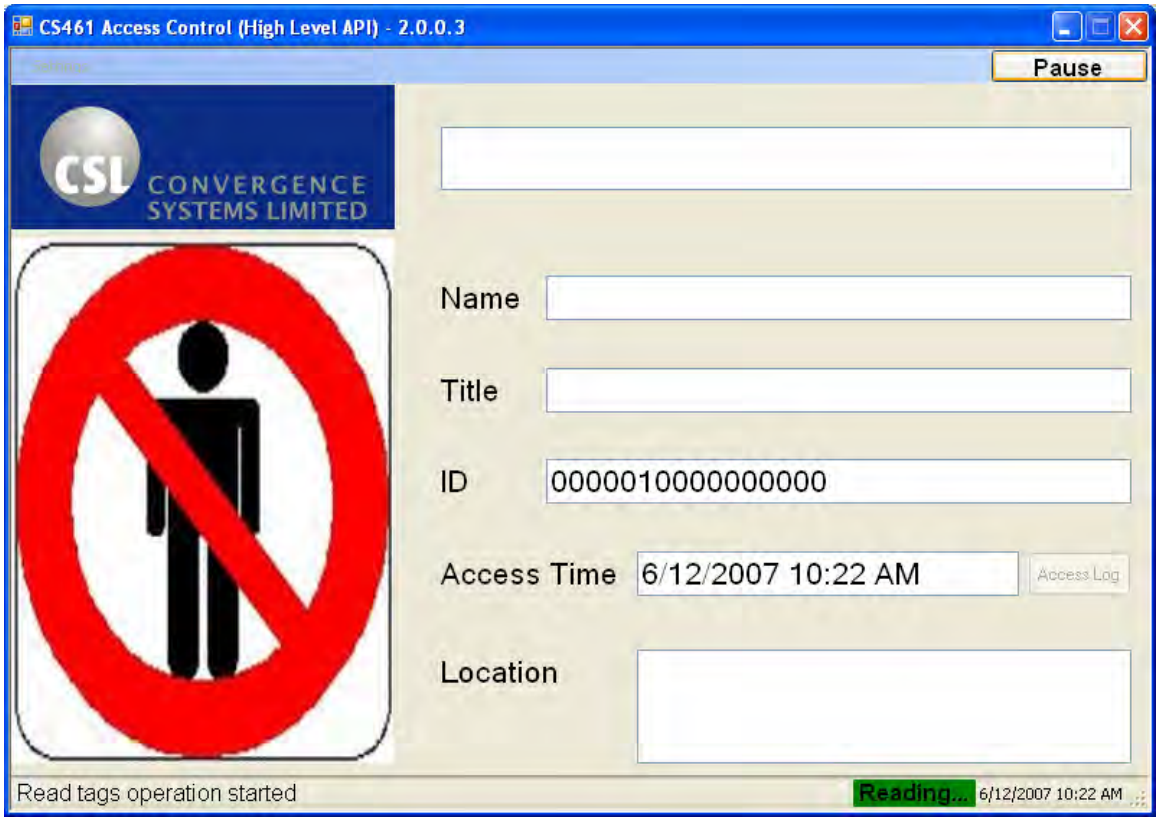
```
public void AccessControl_TagReceiveEvent(object sender, TagReceiveEventArgs e)
{
    if (e.rxTag != null)
    {
        TAG t = (TAG)e.rxTag;
        update_UserInfo(t.TagOrigId);
        reader.saveToLogInfo(String.Format("Tag Receive Event received: {0}",
t.TagOrigId));
    }
    else
    {
        reader.saveToLogInfo("Tag Receive Event received: None");
    }
}
```

If the tag ID exists in the file “db.csv”, the following screen is shown:

Copyright © Convergence Systems Limited, All Rights Reserved



Otherwise, the following screen is shown:



Copyright © Convergence Systems Limited, All Rights Reserved

5.1.6 Sample Usage Scenario – Conveyor Belt

Development Platform

The demo program is developed in Microsoft Visual Studio 2005 Professional Edition. It is written in Visual C# 2005 and utilizes Microsoft .Net Framework 2.0. SQLExpress is used as database in this demo.

File List of Source Code

Filename	Type	Description
CS461_HL_API.cs	Source code	Class for High Level API. It implements API using .Net framework. It could be modified to become a class library and used in other projects.
CS461 Conveyor Belt Demo.csproj	Project file	Project file used by VS2005
frmForm1.cs	Source code	Code for the main screen
frmProductInformation.cs	Source code	Code for the production information dialogue
frmSettings.cs	Source code	Code for the Settings dialogue
frmWelcome.cs	Source code	Code for the welcome dialogue during application start up
Program.cs	Source code	Code for application startup
Properties/AssemblyInfo.cs	Source code	Assembly information

All files with filename ended with “.Designer.cs” and “.resx” are generated by VS2005.

Demo Description

This demo illustrats how to control the I/O ports of the reader. It simulates the conveyor belt in which a tag ID is read and then one of the four LED is turned on according to the shipment destination.

1. **Reader connection**: Once the application starts, it connects to the reader device. In `frmForm1.cs`, an instance of `CS461_HL_API` is created. This instance is for connecting to the reader using High Level API:

```
CS461_HL_API reader = new CS461_HL_API();
```

Copyright © Convergence Systems Limited, All Rights Reserved

Then, a method `loadUserSettings()` is invoked. This method retrieves reader information such as URI, login name and password, and database information from user settings:

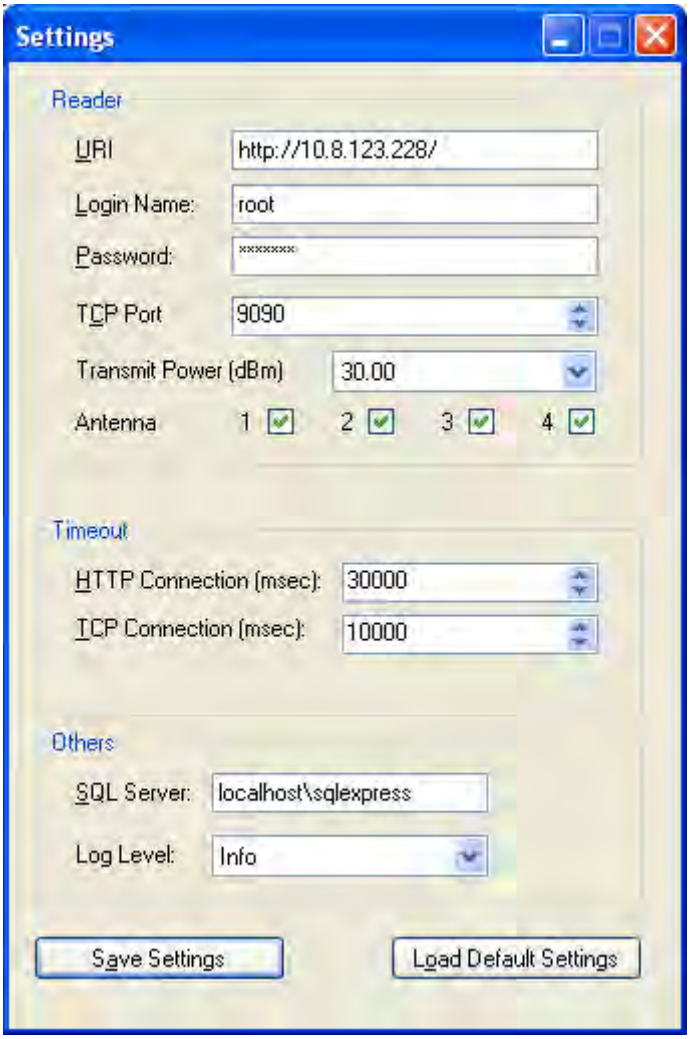


Figure 5-3

The information is set to the `CS461_HL_API` object:

```
reader.login_name =
(string)Application.UserAppDataRegistry.GetValue("LoginName", "root");
reader.login_password =
(string)Application.UserAppDataRegistry.GetValue("LoginPassword", "csl2006");
reader.http_timeout =
(int)Application.UserAppDataRegistry.GetValue("HttpTimeout", 30000);
reader.setURI((string)Application.UserAppDataRegistry.GetValue("URI",
"http://192.168.25.208/"));
reader.api_log_level =
reader.LogLevel((string)Application.UserAppDataRegistry.GetValue("LogLevel",
"Info"));
```

The reader is then connected by invoking the `connect()` method of `CS461_HL_API`:

Copyright © Convergence Systems Limited, All Rights Reserved

```
reader.connect();
```

In the `connect()` method, `login()` method is invoked which in turn calls the "login" command of High Level API using HTTP by supplying the username and password as parameters:

```
string cmd = "login";
...
StringBuilder sbReq = new StringBuilder();
sbReq.Append(String.Format("{0}API?command={1}&username={2}&password={3}",
                                                    httpUri.AbsoluteUri, cmd,
                                                    LoginName, LoginPassword));
string resp = sendHTTPRequest(sbReq.ToString());
```

If login successful, the reader will return an ACK message as follows:

```
<?xml version="1.0" ?>
<CSL>
   <Command>login</Command>
   <Ack>OK: session_id=4c531266</Ack>
</CSL>
```

The `login()` method then retrieves the `session_id` as all commands afterward must contain this id to maintain the login session.

2. **Setup Database**: Database is required for this demo. In the method `loadUserSettings()`, the location of SQL server is retrieved from user's setting:

```
dbServer = (string)Application.UserAppDataRegistry.GetValue("SQLServer",
"localhost\\sqlexpress");
```

On application start, the method `checkDatabase()` is invoked. This method connects to the SQL server and then creates the database and tables required.

3. **Setup Trigger, Action and Event**: Once the reader is connected in `frmForm1.cs`, the method `setupReader()` is invoked. This method set up the Trigger, Action and Trigger required for the application.

   i) Disable all Events that are currently running on reader:

```
//Disable all events
System.Collections.ArrayList eventList;
eventList = reader.listEvent();
if (eventList != null)
{
   foreach (EVENT_INFO e in eventList)
   {
      reader.enableEvent(e.id, false);
   }
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
}
```

The above code segment first retrieves the Event list by invoking the `listEvent()` method of `CS461_HL_API`. This method in turn calls the "listEvent" command of the High Level API using HTTP:

```
string cmd = "listEvent";
...
StringBuilder sbReq = new StringBuilder();
sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));

string resp = sendHTTPRequest(sbReq.ToString());
```

Response of the "listEvent" command will be an XML containing the information of all Event settings current on the reader:

```
<?xml version="1.0" ?>
<CSL>
    <Command>listEvent</Command>
    <EventMode mode="0" />
    <AutomaticConfigure desc="automatic configure CSLEvent for DSPI" enable="true" />
    <EventList>
        <event desc="Event Demo" enable="false" event_id="DemoEvent" event_log="false"
inventoryDisablingTrigger="Never Stop" inventoryEnablingTrigger="Always On"
operProfile_id="Default Profile" resultant_action="DemoAction" triggering_logic="DemoTrigger"
/>
    </EventList>
</CSL>
```

The `listEvent()` method then parses the XML and return an ArrayList of `EVENT_INFO`. For each `EVENT_INFO`, the method `enableEvent()` of `CS461_HL_API` is invoked. This method will call the "enableEvent" command of High Level API using HTTP to disable the event:

```
public bool enableEvent(string id, bool enable)
{
  string cmd = "enableEvent";
  ...
  StringBuilder sbReq = new StringBuilder();

  sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
  sbReq.Append(String.Format("&event_id={0}&enable={1}", id, (enable) ?
"true" : "false"));

  string resp = sendHTTPRequest(sbReq.ToString());
  ...
}
```

The reader will return an ACK message of the command as follows:

```
<?xml version="1.0" ?>
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
<CSL>
    <Command>enableEvent</Command>
    <Ack>OK: </Ack>
</CSL>
```

ii) Setup Operation Profile. The operation profile controls the behavior of the reader such as antenna used and the RF power. In this example, "Autonomous Time Trigger" is used. It allows duplicate elimination which prevents the same tag being sent more than once within the same time window (in this example, the time window is set to 1000 ms). Note that the parameters for operation profile are case sensitive:

```
//Setup Operation Profile
OPERATION_PROFILE profile = new OPERATION_PROFILE();

profile.profile_id = "Default Profile";
profile.profile_enable = true;
profile.modulation_profile = "Profile2";
profile.population = 64;
profile.session_no = 3;
profile.transmit_power =
(string)Application.UserAppDataRegistry.GetValue("TxPower", "30.00");
profile.window_time = 1000;
profile.capture_mode = "Time Window";
profile.ant1_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant1",
0) == 1) ? true : false;
profile.ant2_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant2",
0) == 1) ? true : false;
profile.ant3_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant3",
0) == 1) ? true : false;
profile.ant4_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant4",
0) == 1) ? true : false;
profile.trigger = "Autonomous Time Trigger";

if (reader.setOperProfile(profile) == false)
{
    tsslStatus.Text = "Fail to set operation profile";
    return false;
}
```

In the above code segment, the method setOperProfile() of CS461_HL_API is invoked which in turn calls the "setOperProfile" command using HTTP:

```
string cmd = "setOperProfile";
...
sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
         httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&profile_id={0}&captureMode={1}&duplicateElimina
         tionTime={2}", profile.profile_id, profile.capture_mode,
         profile.window_time));
sbReq.Append(String.Format("&modulationProfile={0}&populationEst={1}&session
         No={2}", profile.modulation_profile, profile.population,
         profile.session_no));
sbReq.Append(String.Format("&transmitPower={0}&antennaPort={1}&enable={2}",
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
          profile.transmit_power, antennaPort, enable));
sbReq.Append(String.Format("&triggerMethod={0}", profile.trigger));

string resp = sendHTTPRequest(sbReq.ToString());
```

iii) Setup Trusted Server. In order to receive event notification, the machine running the application must be set as the trusted server of the reader. In this example, the trusted server mode is set to "Listening Port on Server Side". It means that the reader will try to connect to the IP and port provided when event occurs. Again, this value is case sensitive.

```
//Setup Trusted Server
SERVER_INFO svr = new SERVER_INFO();
svr.id = "Conveyor Belt Server";
svr.desc = "Conveyor Belt Server";
IPHostEntry he = Dns.GetHostEntry(System.Environment.MachineName);
svr.ip = he.AddressList[0].ToString();
svr.server_port = server.tcp_port.ToString();
svr.mode = "Listening Port on Server Side";
svr.enable = true;

if (reader.setServerID(svr) == false)
{
    tsslStatus.Text = "Fail to set trusted server";
    return;
}
```

The method `setServerID()` is invoked which calls the "setServerID" command of High Level API using HTTP to create the trusted server:

```
string cmd = "setServerID";
...
StringBuilder sbReq = new StringBuilder();

string enable = "false";
if (svr.enable)
    enable = "true";

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&server_id={0}&desc={1}", svr.id, svr.desc));
sbReq.Append(String.Format("&server_ip={0}&server_port={1}&enable={2}",
svr.ip, svr.server_port, enable));
sbReq.Append(String.Format("&reader_ip={0}&mode={1}", svr.reader_port,
svr.mode));

string resp = sendHTTPRequest(sbReq.ToString());
```

iv) Setup Resultant Action. In this example, the action mode is set to "Batch Alert to Server" which means the reader will send the tag event report to the trusted servers in a batch at the end of the time window.

```
//Setup Resultant Action
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
reader.delResultantAction("Conveyor Belt Action");

RESULTANT_ACTION_INFO action = new RESULTANT_ACTION_INFO();
action.id = "Conveyor Belt Action";
action.desc = " Conveyor Belt Demo";
action.mode = "Batch Alert to Server";
action.server_id = svr.id;
action.report_id = "Default Report";

if (reader.addResultantAction(action) == false)
{
   tsslStatus.Text = "Fail to set resultant action";
   return;
}
```

The method `addResultantAction()` is invoked which calls the "addResultantAction" command of High Level API using HTTP:

```
string cmd = "addResultantAction";
...
StringBuilder sbReq = new StringBuilder();

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&action_id={0}&desc={1}&action_mode={2}",
info.id, info.desc, info.mode));
sbReq.Append(String.Format("&server_id={0}&report_id={1}", info.server_id,
info.report_id));

string resp = sendHTTPRequest(sbReq.ToString());
```

v) Setup Event. Add an event with "DemoTrigger" as the trigger logic and the action created in the previous step as the resultant action. The "DemoTrigger" used is pre-set which triggers event when any tag is read in any antenna.

```
//Setup Event
reader.delEvent("Conveyor Belt Event");

EVENT_INFO eventInfo = new EVENT_INFO();
eventInfo.id = " Conveyor Belt Event";
eventInfo.desc = " Conveyor Belt Demo";
eventInfo.profile = profile.profile_id;
eventInfo.trigger = "DemoTrigger";
eventInfo.action = action.id;
eventInfo.log = false;
eventInfo.enable = true;
eventInfo.enabling = "Always On";
eventInfo.disabling = "Never Stop";

if (reader.addEvent(eventInfo) == false)
{
    tsslStatus.Text = "Fail to set event";
    return;
}
```

Copyright © Convergence Systems Limited, All Rights Reserved

The method `addEvent()` is invoked which calls the "addEvent" command of High Level API using HTTP:

```
string cmd = "addEvent";
...
StringBuilder sbReq = new StringBuilder();
string eventEnable = "false";
if (info.enable)
   eventEnable = "true";
string eventLog = "false";
if (info.log)
   eventLog = "true";

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
         httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&event_id={0}&desc={1}&triggering_logic={2}&oper
         Profile_id={3}", info.id, info.desc, info.trigger, info.profile));
sbReq.Append(String.Format("&resultant_action={0}&event_log={1}&enable={2}",
         info.action, eventLog, eventEnable));
sbReq.Append(String.Format("&inventoryEnablingTrigger={0}&inventoryDisabling
         Trigger={1}", info.enabling, info.disabling));

string resp = sendHTTPRequest(sbReq.ToString());
```

4. **Reset I/O Ports**: After setting up the reader in the previous step, the application will reset all of the four IO ports of the reader:

```
reader.setIOPort(IO_PORT.Port1, IO_LOGIC.Low);
reader.setIOPort(IO_PORT.Port2, IO_LOGIC.Low);
reader.setIOPort(IO_PORT.Port3, IO_LOGIC.Low);
reader.setIOPort(IO_PORT.Port4, IO_LOGIC.Low);
```

In the `setIOPort()` method, the `runIO_output()` method is invoked which calls the "runIO_output" command of the reader is called using HTTP:

```
public bool runIO_output(int io, int data, string mode)
{
   string cmd = "runIO_output";
   ...
   StringBuilder sbReq = new StringBuilder();
   sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
   if (mode == "run")
   {
      sbReq.Append(String.Format("&mode=run&port={0}&oper_logic={1}", io,
data));
   }
   else
   {
      sbReq.Append(String.Format("&mode=check", mode, io, data));
   }
   string resp = sendHTTPRequest(sbReq.ToString());
   ...
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
}
```

5. **Data Preparation**: When the application is first started, there is no data in the database. To input test data, click the "Database" menu in main screen:

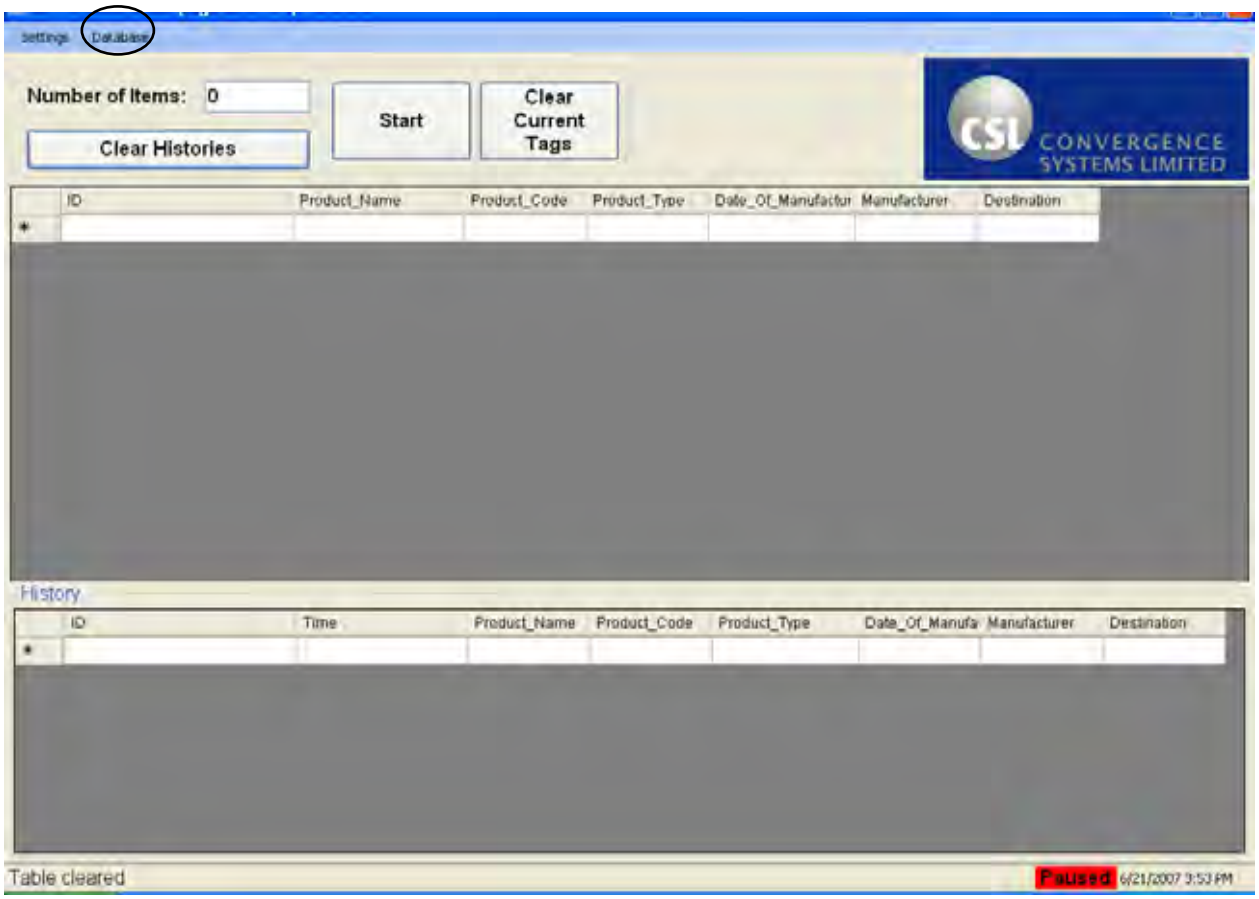


Figure 5-4

The following window will pop-up. Put some tags to the antenna, then clicks the "Read from reader" button. Tag ID will be shown in the window. Input the information for each tag ID. In the destination column, input "Hong Kong", "Beijing", "Shanghai" or "Guangdong". Click the "Save to database" button to save the data.

Copyright © Convergence Systems Limited, All Rights Reserved

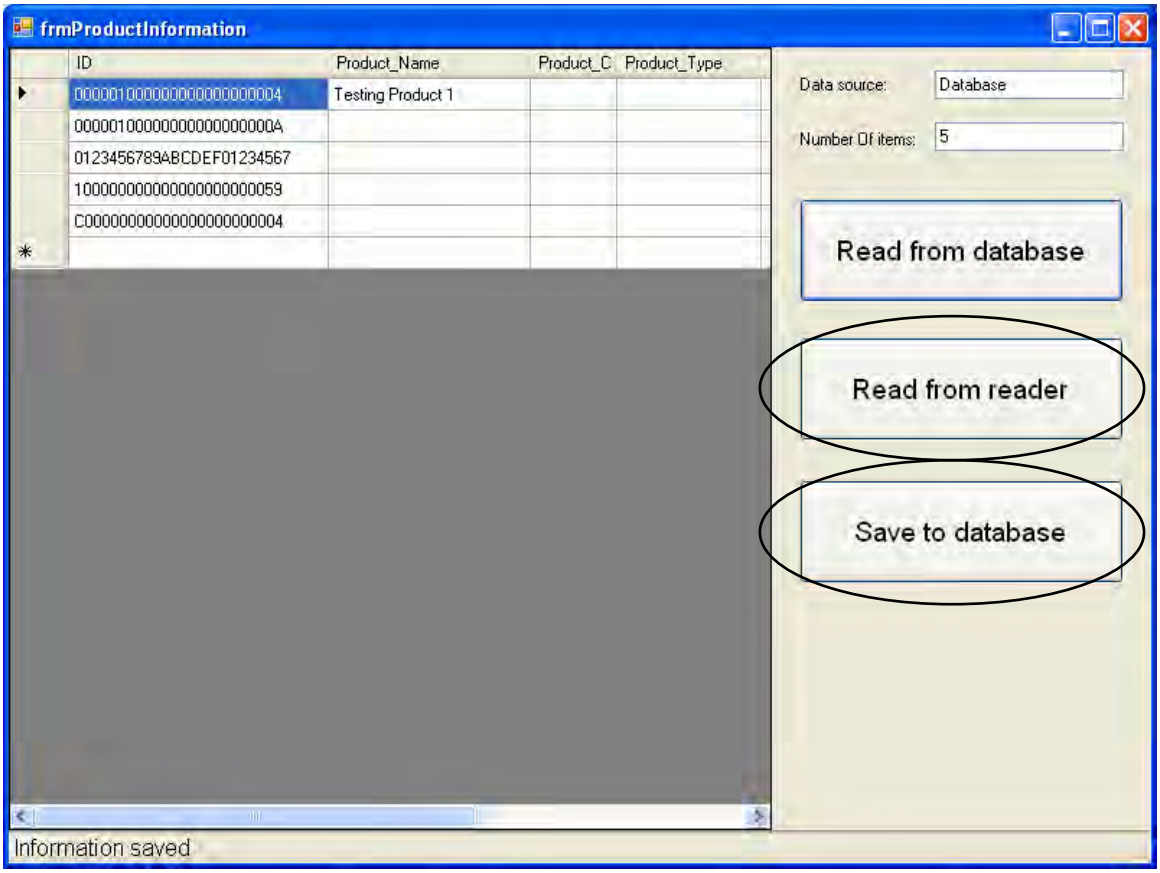


Figure 5-5

6. **Start Trusted Server**: In frmForm1.cs, an instance of TrustedServer is created.

```
TrustedServer server = new TrustedServer();
```

In the method `loadUserSettings()`, the `TrustedServer` object is initialized:

```
server.tcp_port = (int)Application.UserAppDataRegistry.GetValue("TcpPort",
9090);
server.api_log_level = reader.api_log_level;
```

Click the “Start” button to start reading tags:

Copyright © Convergence Systems Limited, All Rights Reserved

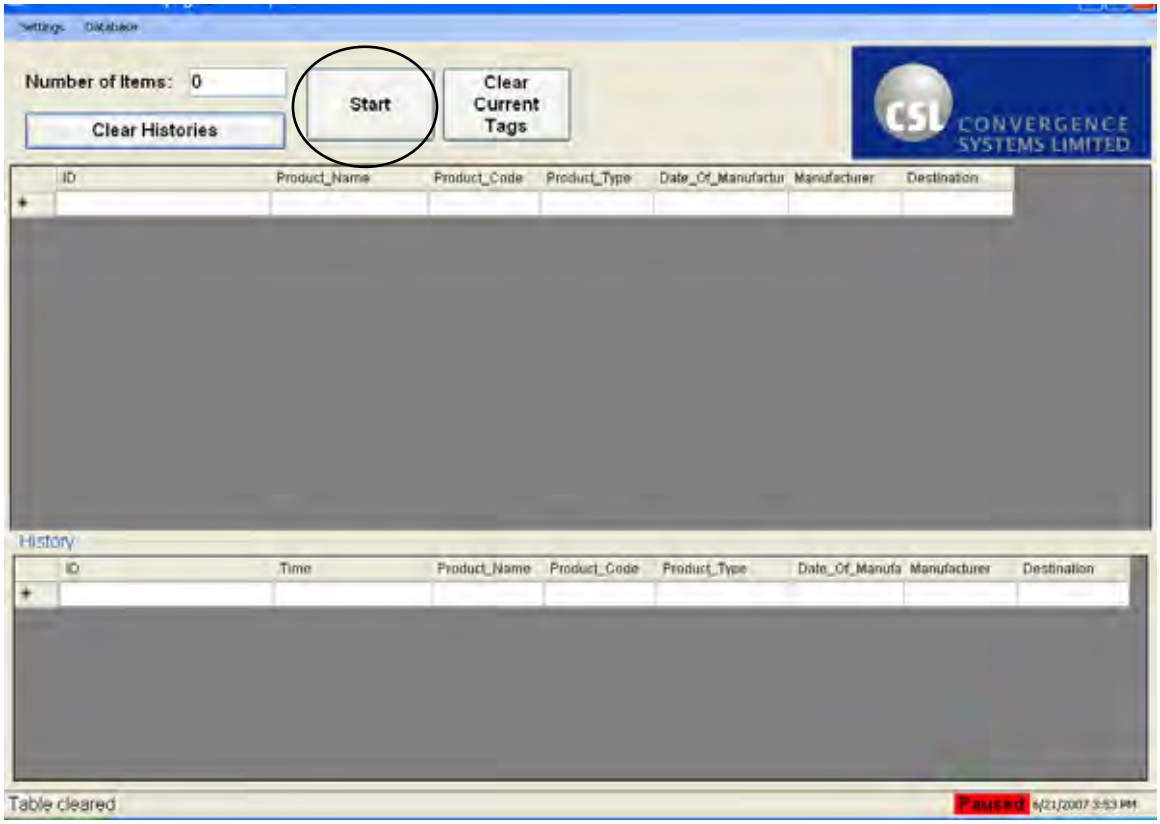


Figure 5-6

When the application starts reading tags, the trusted server is started to receive event notification from the reader. It is done by invoking the `Start()` method of `TrustedServer`.

```
server.Start();
```

7. **Handle Tag Event**: An event handler `server_TagListEvent` is added to the trusted server to handle tag events.

```
server.TagListEvent += new TagListEventHandler(server_TagListEvent);
```

When tags are read, tag event notifications are sent to trusted server in batch with a batch end notification at the end. The tags are then passed to the event handler as `TagListEventArgs`. The event handler clears the tag data in database and adds the newly received tags to the database:

```
public void server_TagListEvent(object sender, TagListEventArgs e)
{
    if (e.TagsList != null)
    {
        lock (dbLock)
        {
            if (e.TagsList.Count > 0)
            {
                delTagsFromDatabase();
                addTagsToDatabase(e.TagsList);
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
                }
        }
        reader.saveToLogInfo(String.Format("Tag List received: {0}",
e.TagsList.Count));
    }
    else
    {
        reader.saveToLogInfo("Tag Receive Event received: None");
    }
}
```

8. **Output Control**: There is a timer task which updates the screen and controls the output ports. It invokes the `showTagsDatabase()` method which retrieves the tag information from the database, update the information on screen and turns on the LED according to the destination of the first record.

```
private void showTagsDatabase()
{
...
        //Update output ports based on 1st record.
        if (reader.connect() == true)
        {
           try
           {
               string o = (string)dgvResult.Rows[0].Cells[6].Value;
               string dest = (string)o;
               if (dest.Equals("Hong Kong",
StringComparison.OrdinalIgnoreCase))
               {
                   reader.setIOPort(IO_PORT.Port1, IO_LOGIC.High);
                   reader.setIOPort(IO_PORT.Port2, IO_LOGIC.Low);
                   reader.setIOPort(IO_PORT.Port3, IO_LOGIC.Low);
                   reader.setIOPort(IO_PORT.Port4, IO_LOGIC.Low);
               }
...
            }
        }
}
```

The information of the tags read is shown on the screen:

Copyright © Convergence Systems Limited, All Rights Reserved

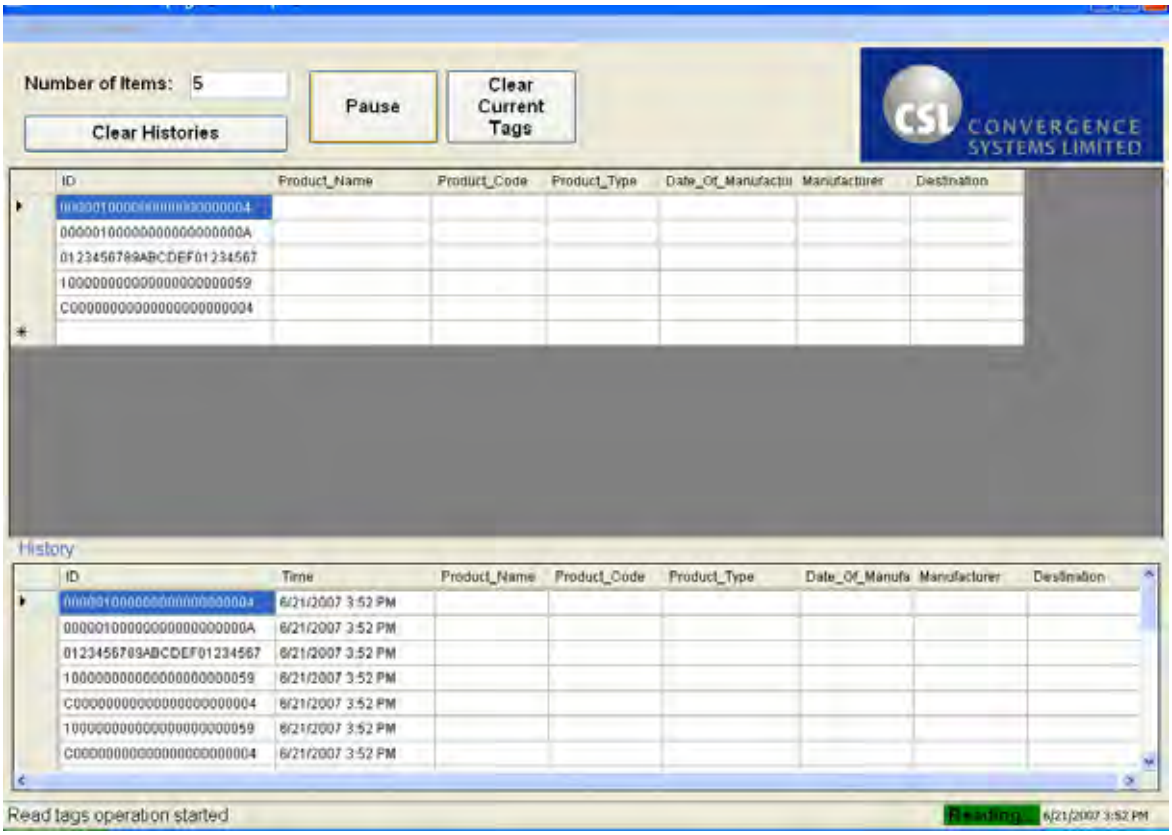

Number of Items: 5
Clear Histories
Pause
Clear Current Tags
CSL CONVERGENCE SYSTEMS LIMITED
ID
Product_Name
Product_Code
Product_Type
Date_Of_Manufactu
Manufacturer
Destination
History
Time
Date_Of_Manufa
Read tags operation started
Reading...
6/21/2007 3:52 PM

Figure 5-7

5.1.7 Sample Usage Scenario – Gambling

Development Platform

The demo program is developed in Microsoft Visual Studio 2005 Professional Edition. It is written in Visual C# 2005 and utilizes Microsoft .Net Framework 2.0.

File List of Source Code

Filename	Type	Description
CS461_HL_API.cs	Source code	Class for High Level API. It implements API using .Net framework. It could be modified to become a class library and used in other projects.
CS461 Fan Tan.csproj	Project file	Project file used by VS2005
frmForm1.cs	Source code	Code for the main screen
frmSettings.cs	Source code	Code for the Settings dialogue
frmWelcome.cs	Source code	Code for the welcome dialogue during application start up
Program.cs	Source code	Code for application startup
Properties/AssemblyInfo.cs	Source code	Assembly information

All files with filename ended with ".Designer.cs" and ".resx" are generated by VS2005.

Demo Description

This demo illustrats how to read tag using "Polling Trigger by Client". It simulates the gambling game which counts the number of tags in field, divides the number by 4, and shows the remainder.

1. **Reader connection**: Once the application starts, it connects to the reader device. In `frmForm1.cs`, an instance of `CS461_HL_API` is created. This instance is for connecting to the reader using High Level API:

```
CS461_HL_API reader = new CS461_HL_API();
```

Then, a method `loadUserSettings()` is invoked. This method retrieves reader information such as URI, login name and password from user settings:

Copyright © Convergence Systems Limited, All Rights Reserved

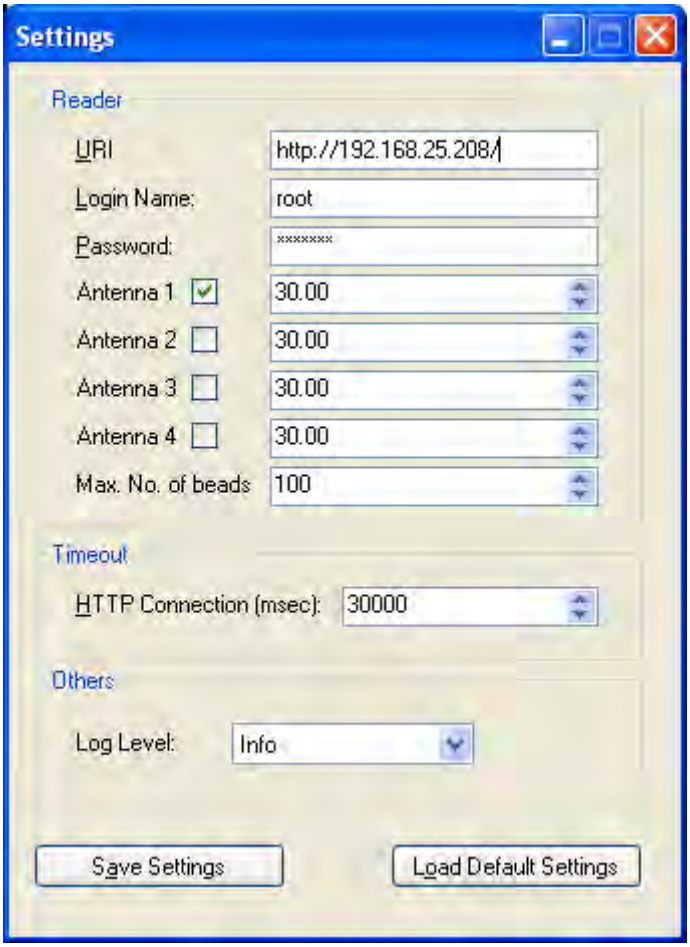


Figure 5-8

The information is set to the `CS461_HL_API` object:

```
reader.login_name =
(string)Application.UserAppDataRegistry.GetValue("LoginName", "root");
reader.login_password =
(string)Application.UserAppDataRegistry.GetValue("LoginPassword", "csl2006");
reader.http_timeout =
(int)Application.UserAppDataRegistry.GetValue("HttpTimeout", 30000);
reader.api_log_level =
reader.LogLevel((string)Application.UserAppDataRegistry.GetValue("LogLevel",
"Info"));
reader.setURI((string)Application.UserAppDataRegistry.GetValue("URI",
"http://192.168.25.208/"));
```

The reader is then connected by invoking the `connect()` method of `CS461_HL_API`:

```
reader.connect();
```

In the `connect()` method, `login()` method is invoked which in turn calls the “login” command of High Level API using HTTP by supplying the username and password as parameters:

```
string cmd = "login";
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
...
StringBuilder sbReq = new StringBuilder();
sbReq.Append(String.Format("{0}API?command={1}&username={2}&password={3}",
httpUri.AbsoluteUri, cmd,  LoginName, LoginPassword));
string resp = sendHTTPRequest(sbReq.ToString());
```

If login successful, the reader will return an ACK message as follows:

```
<?xml version="1.0" ?>
<CSL>
   <Command>login</Command>
   <Ack>OK: session_id=4c531266</Ack>
</CSL>
```

The `login()` method then retrieves the `session_id` as all commands afterward must contain this id to maintain the login session.

2. **Setup Trigger, Action and Event**: Once the reader is connected in `frmForm1.cs`, the method `setupReader()` is invoked. This method set up the Trigger, Action and Trigger required for the application.

   i) Disable all Events that are currently running on reader:

```
//Disable all events
System.Collections.ArrayList eventList;
eventList = reader.listEvent();
if (eventList != null)
{
   foreach (EVENT_INFO e in eventList)
   {
      reader.enableEvent(e.id, false);
   }
}
```

   The above code segment first retrieves the Event list by invoking the `listEvent()` method of `CS461_HL_API`. This method in turn calls the "listEvent" command of the High Level API using HTTP:

```
string cmd = "listEvent";
...
StringBuilder sbReq = new StringBuilder();
sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));

string resp = sendHTTPRequest(sbReq.ToString());
```

   Response of the "listEvent" command will be an XML containing the information of all Event settings current on the reader:

```
<?xml version="1.0" ?>
<CSL>
   <Command>listEvent</Command>
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
    <EventMode mode="0" />
    <AutomaticConfigure desc="automatic configure CSLEvent for DSPI" enable="true" />
    <EventList>
        <event desc="Event Demo" enable="false" event_id="DemoEvent" event_log="false"
inventoryDisablingTrigger="Never Stop" inventoryEnablingTrigger="Always On"
operProfile_id="Default Profile" resultant_action="DemoAction" triggering_logic="DemoTrigger"
/>
    </EventList>
</CSL>
```

The `listEvent()` method then parses the XML and return an ArrayList of `EVENT_INFO`. For each `EVENT_INFO`, the method `enableEvent()` of `CS461_HL_API` is invoked. This method will call the "enableEvent" command of High Level API using HTTP to disable the event:

```
public bool enableEvent(string id, bool enable)
{
  string cmd = "enableEvent";
  ...
  StringBuilder sbReq = new StringBuilder();

  sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
  sbReq.Append(String.Format("&event_id={0}&enable={1}", id, (enable) ?
"true" : "false"));

  string resp = sendHTTPRequest(sbReq.ToString());
  ...
}
```

The reader will return an ACK message of the command as follows:

```
<?xml version="1.0" ?>
<CSL>
    <Command>enableEvent</Command>
    <Ack>OK: </Ack>
</CSL>
```

ii) Setup Operation Profile. The operation profile controls the behavior of the reader such as antenna used and the RF power. In this example, "Polling Trigger by Client" is used. Tag events are not sent to trusted server but the client application initiates polling of the tags. A tag will only be reported once for each polling trigger sent by the client application Note that the parameters for operation profile are case sensitive:

```
//Setup Operation Profile
OPERATION_PROFILE profile = new OPERATION_PROFILE();

profile.profile_id = "Default Profile";
profile.profile_enable = true;
profile.modulation_profile = "Profile0";
profile.population =
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
int.Parse((string)Application.UserAppDataRegistry.GetValue("MaxBeads",
"100"));
profile.session_no = 3;
profile.transmit_power = "30.00";
profile.window_time = 1000;
profile.capture_mode = "Time Window";
profile.ant1_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant1",
0) == 1) ? true : false;
profile.ant2_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant2",
0) == 1) ? true : false;
profile.ant3_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant3",
0) == 1) ? true : false;
profile.ant4_enable = ((int)Application.UserAppDataRegistry.GetValue("Ant4",
0) == 1) ? true : false;
profile.trigger = "Polling Trigger by Client";
profile.ant1_power =
(string)Application.UserAppDataRegistry.GetValue("TxPower1", "30.00");
profile.ant2_power =
(string)Application.UserAppDataRegistry.GetValue("TxPower2", "30.00");
profile.ant3_power =
(string)Application.UserAppDataRegistry.GetValue("TxPower3", "30.00");
profile.ant4_power =
(string)Application.UserAppDataRegistry.GetValue("TxPower4", "30.00");

if (reader.setOperProfile_TxPowers(profile) == false)
{
    tsslStatus.Text = "Fail to set operation profile";
    return false;
}
```

In the above code segment, the method setOperProfile() of CS461_HL_API is invoked which in turn calls the “setOperProfile” command using HTTP:

```
string cmd = "setOperProfile";
...
sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
         httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&profile_id={0}&captureMode={1}&duplicateElimina
         tionTime={2}", profile.profile_id, profile.capture_mode,
         profile.window_time));
sbReq.Append(String.Format("&modulationProfile={0}&populationEst={1}&session
         No={2}", profile.modulation_profile, profile.population,
         profile.session_no));
sbReq.Append(String.Format("&transmitPower={0}&antennaPort={1}&enable={2}",
         profile.transmit_power, antennaPort, enable));
sbReq.Append(String.Format("&triggerMethod={0}", profile.trigger));

string resp = sendHTTPRequest(sbReq.ToString());
```

iii) Setup Trusted Server. Though this demo will not use trusted server to receive tag notification, a trusted server is still setup in the reader. In this example, the trusted server mode is set to “Listening Port on Server Side”. It means that the reader will try to connect to the IP and port provided when event occurs. Again, this value is case sensitive.

Copyright © Convergence Systems Limited, All Rights Reserved

```
//Setup Trusted Server
SERVER_INFO svr = new SERVER_INFO();
svr.id = "Fan Tan Server";
svr.desc = "Fan Tan Server";
IPHostEntry he = Dns.GetHostEntry(System.Environment.MachineName);
svr.ip = he.AddressList[0].ToString();
svr.server_port = "9090";
svr.mode = "Listening Port on Server Side";
svr.enable = true;

if (reader.setServerID(svr) == false)
{
    if (reader.modServerID(svr) == false)
    {
        tsslStatus.Text = "Fail to set trusted server";
        return false;
    }
}
```

The method `setServerID()` is invoked which calls the "setServerID" command of High Level API using HTTP to create the trusted server:

```
string cmd = "setServerID";
...
StringBuilder sbReq = new StringBuilder();

string enable = "false";
if (svr.enable)
   enable = "true";

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&server_id={0}&desc={1}", svr.id, svr.desc));
sbReq.Append(String.Format("&server_ip={0}&server_port={1}&enable={2}",
svr.ip, svr.server_port, enable));
sbReq.Append(String.Format("&reader_ip={0}&mode={1}", svr.reader_port,
svr.mode));

string resp = sendHTTPRequest(sbReq.ToString());
```

iv) Setup Triggering Login. In this example, any tag read in the selected antenna will trigger the event:

```
//Setup Triggering Logic
reader.delTriggeringLogic("Fan Tan Logic");

TRIGGER_INFO trigger = new TRIGGER_INFO();
trigger.id = "Fan Tan Logic";
trigger.desc = "Fan Tan Demo";
trigger.mode = "Read Any Tags (any ID, 1 trigger per tag)"; //For firmware 2.1.0
or later
trigger.capture_point = "";
trigger.capture_point +=
((int)Application.UserAppDataRegistry.GetValue("Ant1", 0) == 1) ? "1" : "";
trigger.capture_point +=
((int)Application.UserAppDataRegistry.GetValue("Ant2", 0) == 1) ? "2" : "";
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
trigger.capture_point +=
((int)Application.UserAppDataRegistry.GetValue("Ant3", 0) == 1) ? "3" : "";
trigger.capture_point +=
((int)Application.UserAppDataRegistry.GetValue("Ant4", 0) == 1) ? "4" : "";
trigger.logic = "";

if (reader.addTriggeringLogic(trigger) == false)
{
    trigger.mode = "Read Any Tags";     //For firmware 2.0.9, 2.0.10
    if (reader.addTriggeringLogic(trigger) == false)
    {
        tsslStatus.Text = "Fail to set triggering logic";
        return false;
    }
}
```

The method `addTriggeringLogic()` is invoked which calls the "addTriggeringLogic" command of High Level API using HTTP:

```
string cmd = "addTriggeringLogic";
StringBuilder sbReq = new StringBuilder();

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&logic_id={0}&desc={1}&mode={2}", info.id,
info.desc, info.mode));
if (info.capture_point != "")
    sbReq.Append(String.Format("&capture_point={0}", info.capture_point));
if (info.logic != "")
    sbReq.Append(String.Format("&logic={0}", info.logic));

string resp = sendHTTPRequest(sbReq.ToString());
```

v) Setup Resultant Action. In this example, the action mode is set to "Batch Alert to Server" which means the reader will send the tag event report to the trusted servers in a batch at the end of the time window.

```
//Setup Resultant Action
reader.delResultantAction("Fan Tan Action");

RESULTANT_ACTION_INFO action = new RESULTANT_ACTION_INFO();
action.id = "Fan Tan Action";
action.desc = "Fan Tan Demo";
action.mode = "Batch Alert to Server";
action.server_id = svr.id;
action.report_id = "Default Report";

if (reader.addResultantAction(action) == false)
{
    tsslStatus.Text = "Fail to set resultant action";
    return false;
}
```

Copyright © Convergence Systems Limited, All Rights Reserved

The method `addResultantAction()` is invoked which calls the "addResultantAction" command of High Level API using HTTP:

```
string cmd = "addResultantAction";
...
StringBuilder sbReq = new StringBuilder();

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&action_id={0}&desc={1}&action_mode={2}",
info.id, info.desc, info.mode));
sbReq.Append(String.Format("&server_id={0}&report_id={1}", info.server_id,
info.report_id));

string resp = sendHTTPRequest(sbReq.ToString());
```

vi) Setup Event. Add an event with the trigger logic and the action created in the previous steps:

```
//Setup Event
reader.delEvent("Fan Tan Event");

EVENT_INFO eventInfo = new EVENT_INFO();
eventInfo.id = "Fan Tan Event";
eventInfo.desc = "Fan Tan Demo";
eventInfo.profile = profile.profile_id;
eventInfo.trigger = "Fan Tan Logic";
eventInfo.action = action.id;
eventInfo.log = false;
eventInfo.enable = true;
eventInfo.enabling = "Always On";
eventInfo.disabling = "Never Stop";

if (reader.addEvent(eventInfo) == false)
{
    tsslStatus.Text = "Fail to set event";
    return false;
}
```

The method `addEvent()` is invoked which calls the "addEvent" command of High Level API using HTTP:

```
string cmd = "addEvent";
...
StringBuilder sbReq = new StringBuilder();
string eventEnable = "false";
if (info.enable)
   eventEnable = "true";
string eventLog = "false";
if (info.log)
   eventLog = "true";

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}",
         httpUri.AbsoluteUri, cmd, SessionId));
sbReq.Append(String.Format("&event_id={0}&desc={1}&triggering_logic={2}&oper
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
          Profile_id={3}", info.id, info.desc, info.trigger, info.profile));
sbReq.Append(String.Format("&resultant_action={0}&event_log={1}&enable={2}",
          info.action, eventLog, eventEnable));
sbReq.Append(String.Format("&inventoryEnablingTrigger={0}&inventoryDisabling
          Trigger={1}", info.enabling, info.disabling));

string resp = sendHTTPRequest(sbReq.ToString());
```

3. **Start Inventory**: After the reader is setup in the previous step, `startInventory()` is invoked which Polling Trigger the event..

```
reader.startInventory();
```

This method calls the "startInventory" command of High Level API using HTTP with parameter mode=pollingTrigger:

```
public bool startInventory()
{
   string cmd = "startInventory";
...
   StringBuilder sbReq = new StringBuilder();

         sbReq.Append(String.Format("{0}API?command={1}&session_id={2}&mode=
         pollingTrigger", httpUri.AbsoluteUri, cmd, SessionId));

   string resp = sendHTTPRequest(sbReq.ToString());
...
}
```

4. **Get Tags Read**: When the application starts, the following screen is shown:

Copyright © Convergence Systems Limited, All Rights Reserved

Figure 5-9

To start the game, put a number of tags on the antenna, then click the "Show Result" button. The following code segment will run:

```
if (reader.connect())
{
    System.Collections.ArrayList list = reader.getCaptureTagsRaw("getEPC");
    reader.startInventory();
    reader.logout();

    if (list != null)
    {
        update_ShowResult(list.Count);
    }
}
```

This method `getCaptureTagsRaw()`retrieves the tag list scanned after the previous "startInventory" command. Then, `startInventory()` is invoked again to start the next round of tag capturing.

In `getCaptureTagsRaw()`, "getCaptureTagsRaw" command is sent to the reader using HTTP:

```
StringBuilder sbReq = new StringBuilder();

sbReq.Append(String.Format("{0}API?command={1}&session_id={2}&mode=pollingTr
igger", httpUri.AbsoluteUri, cmd, SessionId));
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
string resp = sendHTTPRequest(sbReq.ToString());
```

The result of the command will be an XML including a list of tags:

```
<?xml version="1.0" ?>
<CSL>
    <Command>getCaptureTagsRaw</Command>
    <TagList>
        <tagEPC capturepoint_id="Antenna1" capturepoint_name="Capture Point 1"
event_id="Fan Tan Event" freq="1215" index="A0" reader_ip="10.8.123.228" rssi="-38"
tag_id="7018000000000018" time="1182748685" />
        <tagEPC capturepoint_id="Antenna1" capturepoint_name="Capture Point 1"
event_id="Fan Tan Event" freq="855" index="A1" reader_ip="10.8.123.228" rssi="-51"
tag_id="300833B2DDD9014035050000" time="1182748691" />
        <tagEPC capturepoint_id="Antenna1" capturepoint_name="Capture Point 1"
event_id="Fan Tan Event" freq="885" index="A2" reader_ip="10.8.123.228" rssi="-39"
tag_id="7018000000000044" time="1182749944" />
…
    </TagList>
</CSL>
```

5. **Show result**: After retrieving the tag list, the application counts the total number of the tags, divide it by 4 and show the reminder:

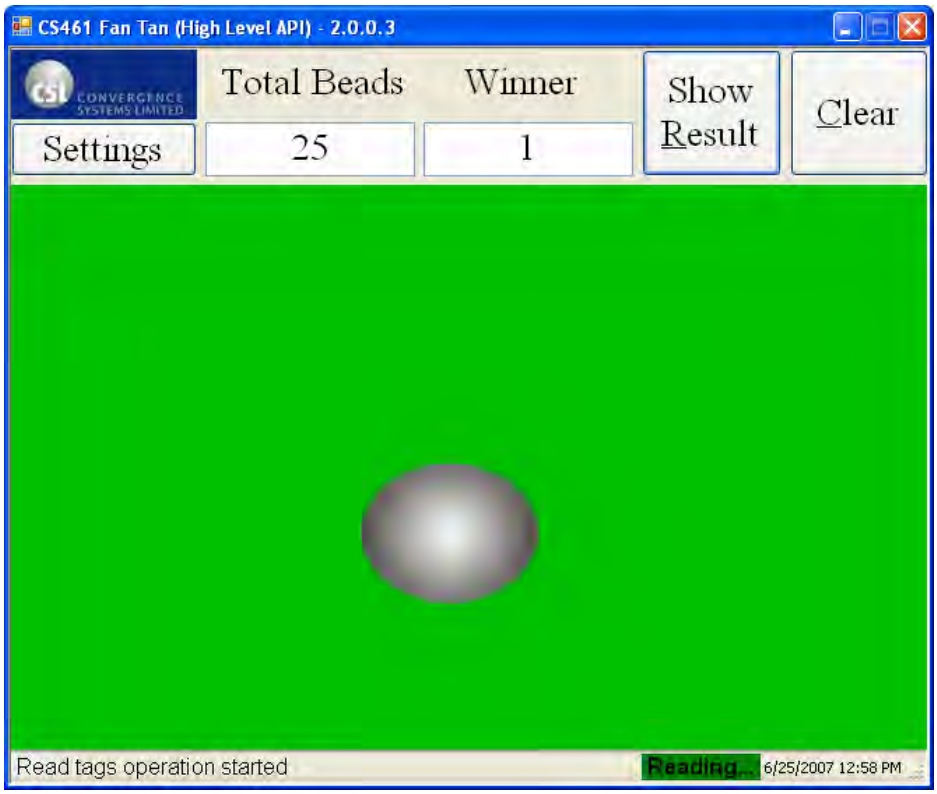


Figure 5-10

Copyright © Convergence Systems Limited, All Rights Reserved

5.2 Low Level API

In Low level API mode, server application connects to the CSL CS-461 reader using TCP connection directly. Reader receives requests, called Commands, from server application and responds to each Command with one Response or Notification. The reader may also generate any number of asynchronous Notifications that are not response to any particular Command.

The Low Level MACH1 API includes the following command sets:

1) Management Command Set (MCS)
2) Operating Command Set (OCS)

With the Low Level MACH1 API, operations such as read tags, write tags, lock tags and kill tags can be performed. A C-based library implemented using the Low Level MACH1 API is available for application development.

5.2.1 Modem States

A modem is referring to specific reader components. It has the following seven states:

1. **Off**: The modem system is turned off, and unable to accept Mach1 commands except commands from Mach1-MCS.
2. **Init**: The modem system has been started but has not had regulatory information set into it yet. In this state it cannot perform RFID operations and must be configured properly before continuing.
3. **Idle**: The modem system is idle, and able to accept Mach1 commands
4. **Active**: The modem system is busy and unable to accept Mach1 commands, except modem-control commands that instruct the modem to exit the active state and return control to the CPU application (e.g. Modem-StopCmd).
5. **Halt**: The modem system has opened a tag for access and is waiting for the CPU application to provide it with an operation to execute on the accessed tag.
6. **Test**: The modem system is busy performing tests or proprietary operations.
7. **Access**: The modem is in the process of performing an access operation on a tag.

The modem state changes as follows:

Copyright © Convergence Systems Limited, All Rights Reserved

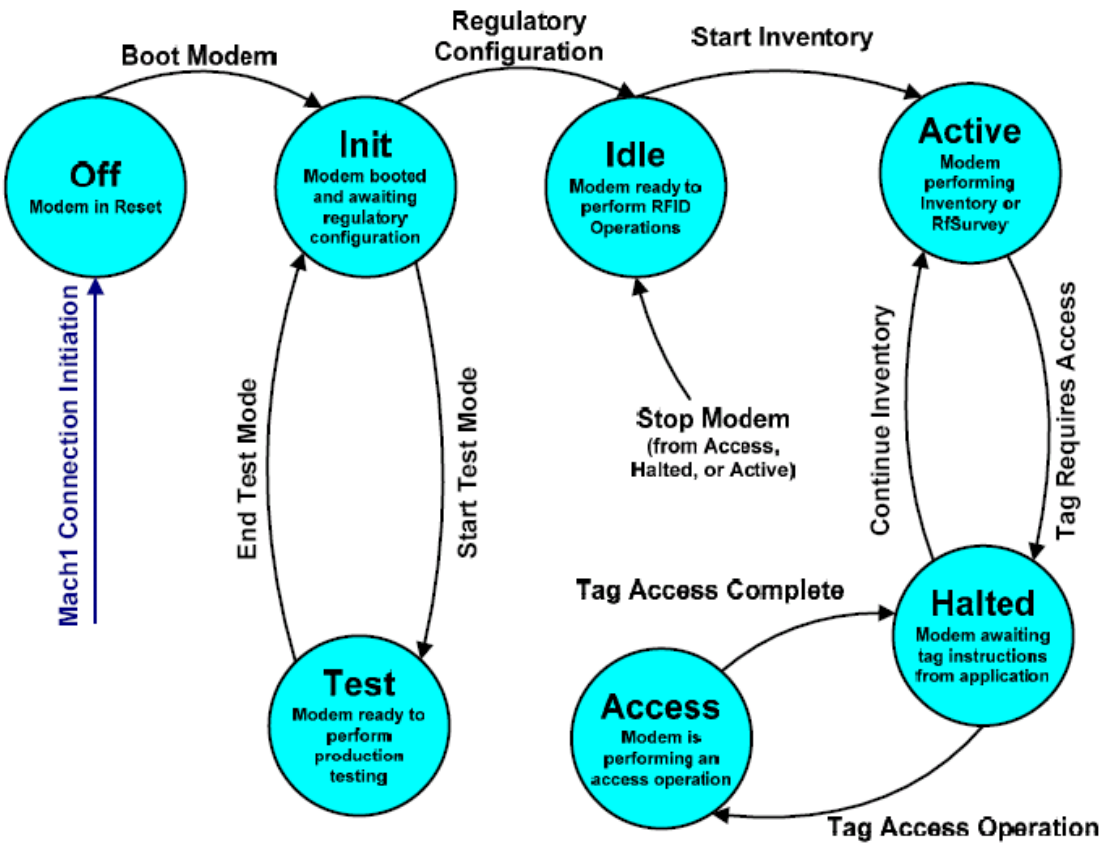


Figure 5-11 March1 State Machine

5.2.2 Sample Usage Scenario – Start Inventory

The following table lists the flow of API method calls using the C library of Low Level API to perform start inventory. State change of modem system and Response/Notification of the API calls are also described in the table:

Copyright © Convergence Systems Limited, All Rights Reserved

Command Flow	Before	After	Rsp or Ntf from Reader	Remarks
Reader = MAPI_Connect(myErrorCallback, ipAddress) Resp = MAPI_BootModemCmd(Reader)	OFF OFF	OFF Init	 BootModemRsp BootModemNtf	Check for non null reader Check response
MAPI_SetRegulatoryRegionCmd(Reader,0)	Init	Idle	SetRegulatoryRegionNtf	0 for FCC
BootNTF = MAPI_GetData(Reader) MAPI_Free(ReturnedData, Data)	Idle	Idle	DATA_TYPE_BOOT_MODEM_NTF	Check for success boot
Resp = MAPI_LoadFromProfileCmd(Reader, ProfileData)	Idle	Idle	LoadFromProfileRsp	Set Profile and check resp for success
MAPI_SetGen2ParamsCmd(Reader, gen2ParamsSettings) MAPI_SetAntennaCmd(Reader, antennaSettings) MAPI_SetTxPowerCmd(Reader, txPowerSettings)	Idle	Idle	SetGen2ParamsRsp SetAntennaRsp SetTxPowerRsp	
MAPI_InventoryCmd(Reader, inventorySettings)	Idle	Active	InventoryRsp InventoryNtf	Start Inventory run

Copyright © Convergence Systems Limited, All Rights Reserved

6 CSL Demo Programs

6.1 High Level API Demo Program

There is a Windows-based program comes with the reader for user to test the reader in "High-level API Access Mode". The demo program has the following features:

- Set / Get Reader's ID.
- Set / Get Operation Profile.
- Set / Get Capture Points Name.
- Set / Get Trusted Server (Notification).
- Set / Get Triggering Method (Notification).
- Set / Get Resultant Action (Notification).
- Set / Get Event (Notification).
- Receive and display Tag information. (One TCP connection only)
- Receive and display Antenna mismatch notification.
- Save received tags information to a file.
- Save "High Level API" data to a log file.

All the source codes for this program (written in C#) is freely downloadable. The user is advised to follow this sample and develop his/her codes.

6.1.1 Installing Demo Program

Please make sure the demo program version is compatible with the firmware version of reader. Refer to the file "compatibility matrix.xls" for the compatibility of demo program and reader firmware.

Please make sure "Microsoft .NET Framework Version 2.0 Redistributable Package" is installed before using the demo program.

Normally, the executed file of demo program is archived as RAR or ZIP file. The archived file is distributed through email, ftp server or website.

Please extract the demo program to a directory (e.g. "C:\CS461 DEMO\"). Then, run the demo program from the installed directory.

Copyright © Convergence Systems Limited, All Rights Reserved

6.1.2 Using Demo Program

Run the demo program from installed directory. Once the program started correctly, the following screen should be shown.

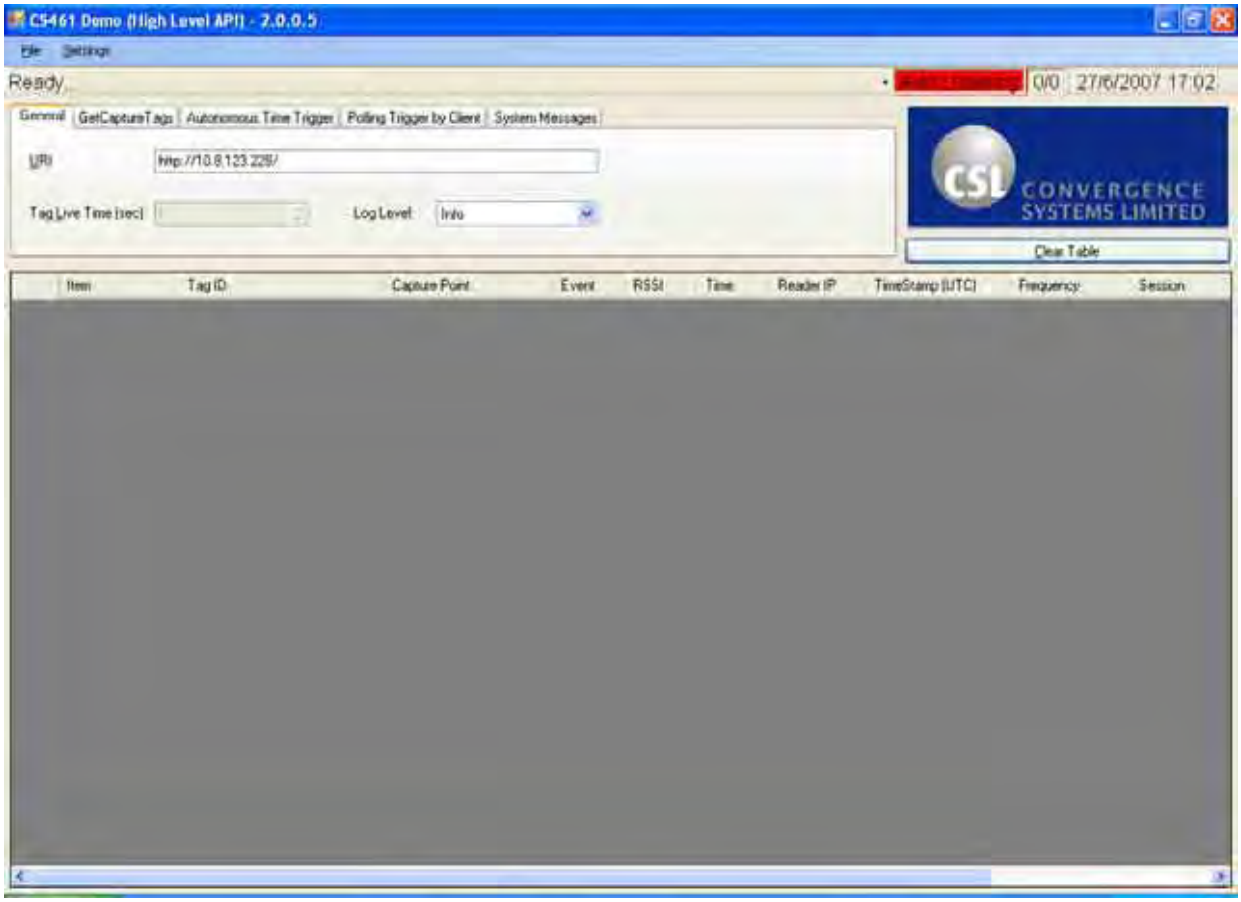


Figure 6-1

Please enter reader's address in URI box. An incorrect formatted URI will show in red color. (

). The address must be started with "http://".

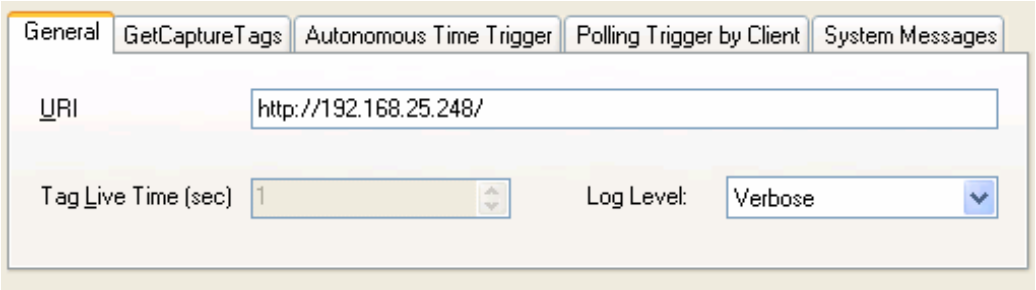


Figure 6-2

Copyright © Convergence Systems Limited, All Rights Reserved

Please enter an Alert port number in Autonomous Time Trigger Tab that receives Alert (Notification) information from reader.

Click "Start Read Alert" button to start waiting for reader's Autonomous Alert (Notification). Any tag information received will be shown in the table. The "Tag counters" box shows number of unique tags received and total number of tags received. Click "Clear Table" to remove all tags from table and reset the Tag counters to zero.

If the reader is set to "Autonomous Time Trigger Mode", **tags information will be received periodically.** The period is defined in "Operation Profile".

Click "Stop Read Alert" to stop waiting for Alert (Notification).

Figure 6-3

In "Polling Trigger by Client Mode", tags are buffered in reader's memory. The buffered tag information will not send to demo program periodically. User should enter an Alert port number in Polling Trigger by Client Tab. Then click "Start Polling Trigger" button to start waiting for reader's Alert (Notification).

The buffered tags are received on demand. User shall click "Read Tags" button to send a request to reader such that the reader return a set of tag information received in last polling session using notification and start a new polling session. Received tag information received will be shown in the table. The "Tag counters" box shows number of unique tags received and total number of tags received. Click "Clear Table" to remove all tags from table and reset the Tag counters to zero.

Copyright © Convergence Systems Limited, All Rights Reserved

Click “Stop Polling Trigger” to stop waiting for Alert (Notification).

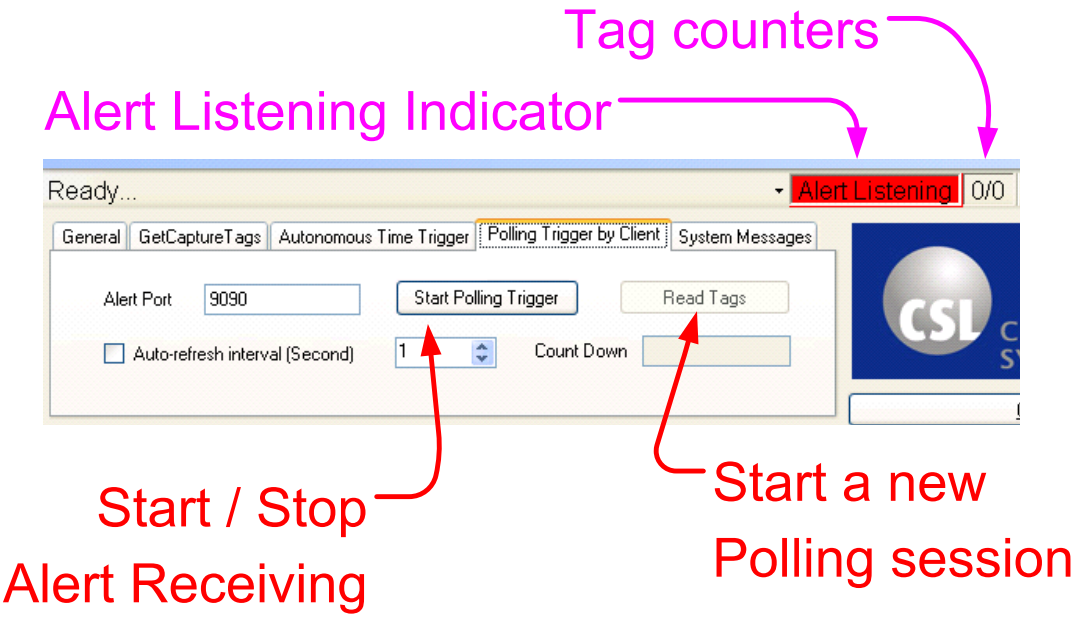


Figure 6-4

6.1.2.1 Autonomous Time Trigger Mode

Reader in this mode will report received tags information to notification server periodically as shown in the following figure. The notification period is controlled by a parameter called "Duplicate Elimination Time".

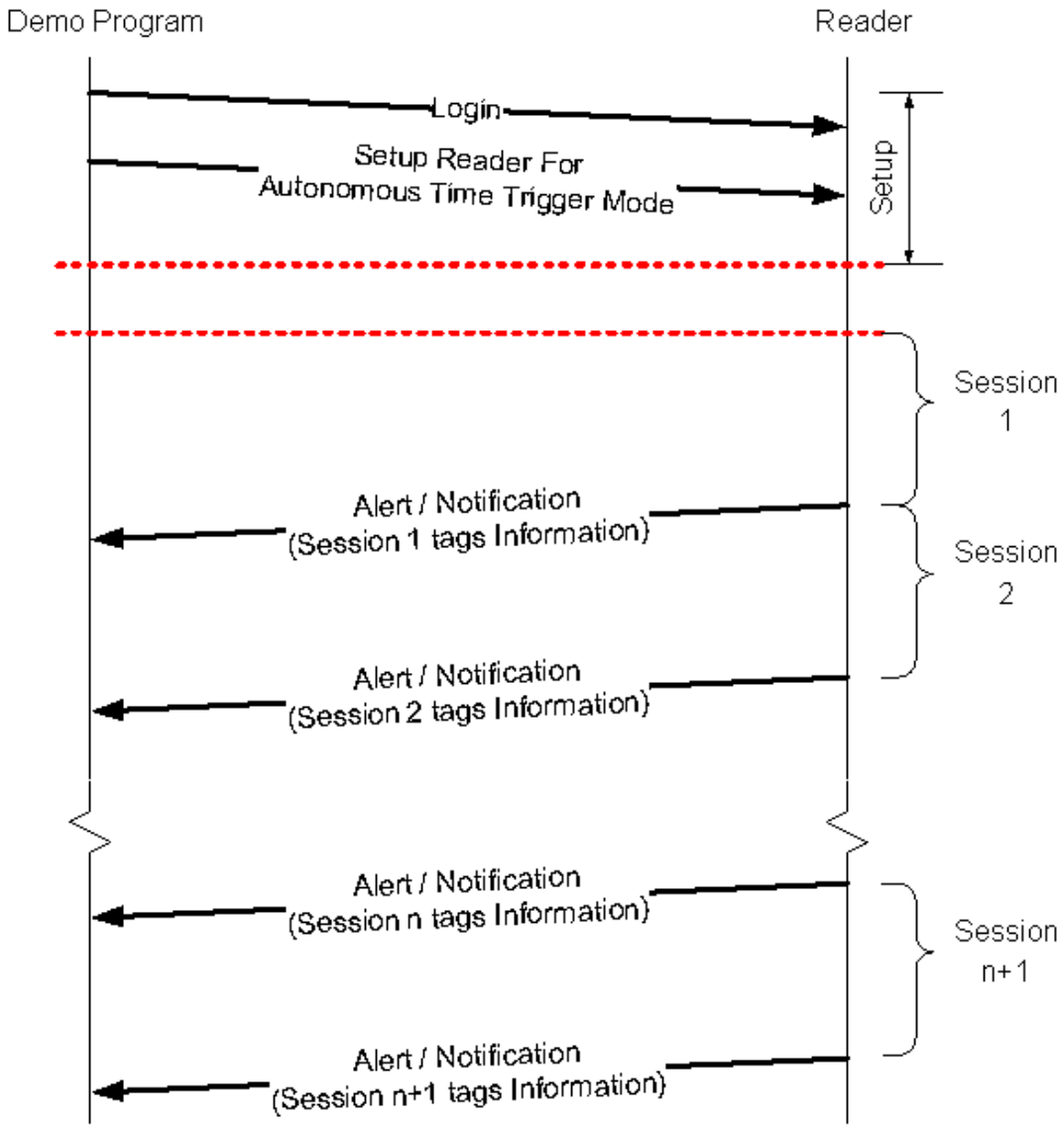


Figure 6-5

Copyright © Convergence Systems Limited, All Rights Reserved

Set the reader to "Autonomous Time Trigger" mode

1. Open "Operation Profile Dialog" from pull-down menu. Then, set the "Duplicate Elimination Triggering Method" to "Autonomous Time Trigger". Set the notification period in "Duplicate Elimination Time" as well.

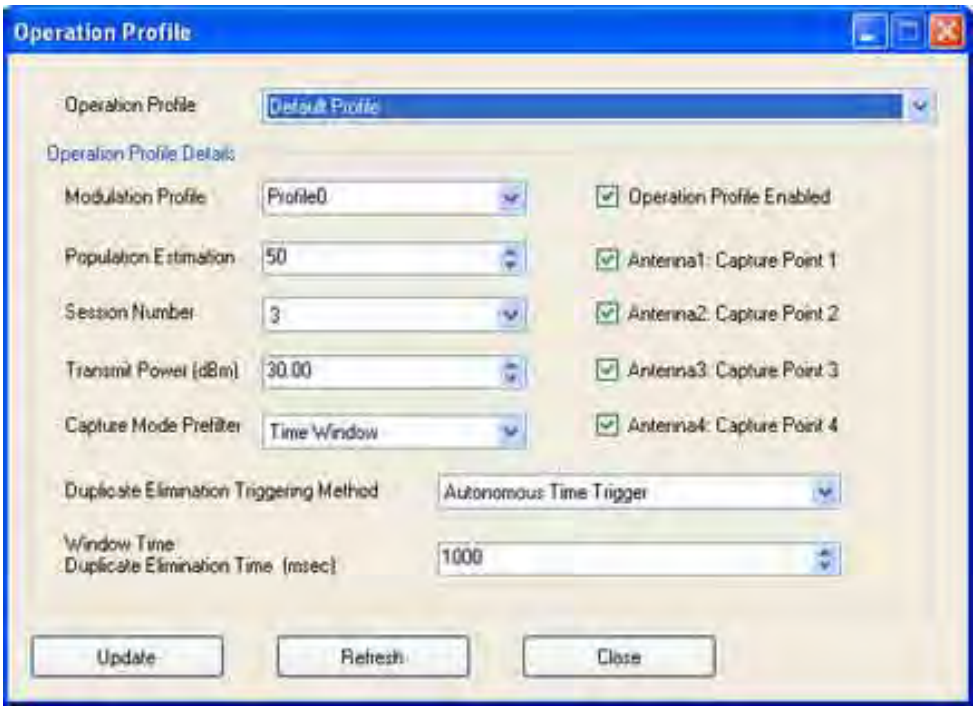


Figure 6-6

2. If the PC is not one of the Trusted Server in reader, please add the PC to reader's trusted server list by opening "Trusted Server Dialog" from pull-down menu. Then enter correct information in the entries and click "Update" button.

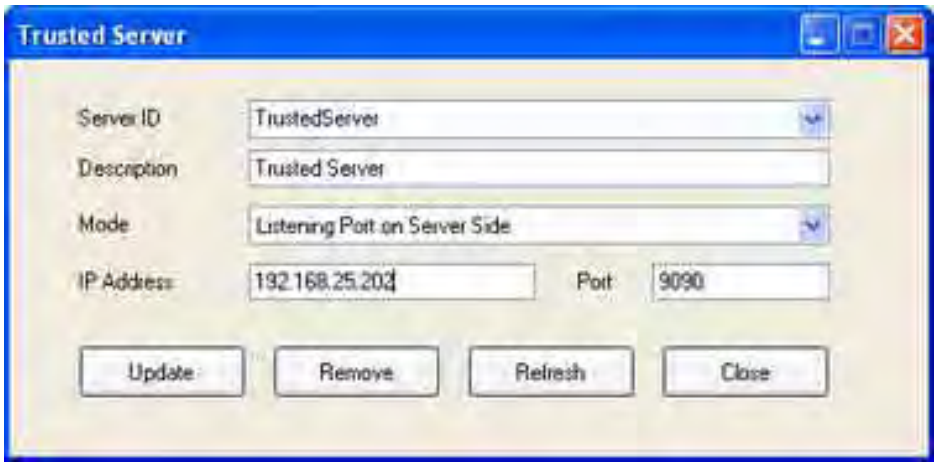


Figure 6-7

Copyright © Convergence Systems Limited, All Rights Reserved

3. Add a number of “Trigger” if necessary. Please open “Trigger Dialog” from pull-down menu. Then enter correct information in the entries and click “Update” button.

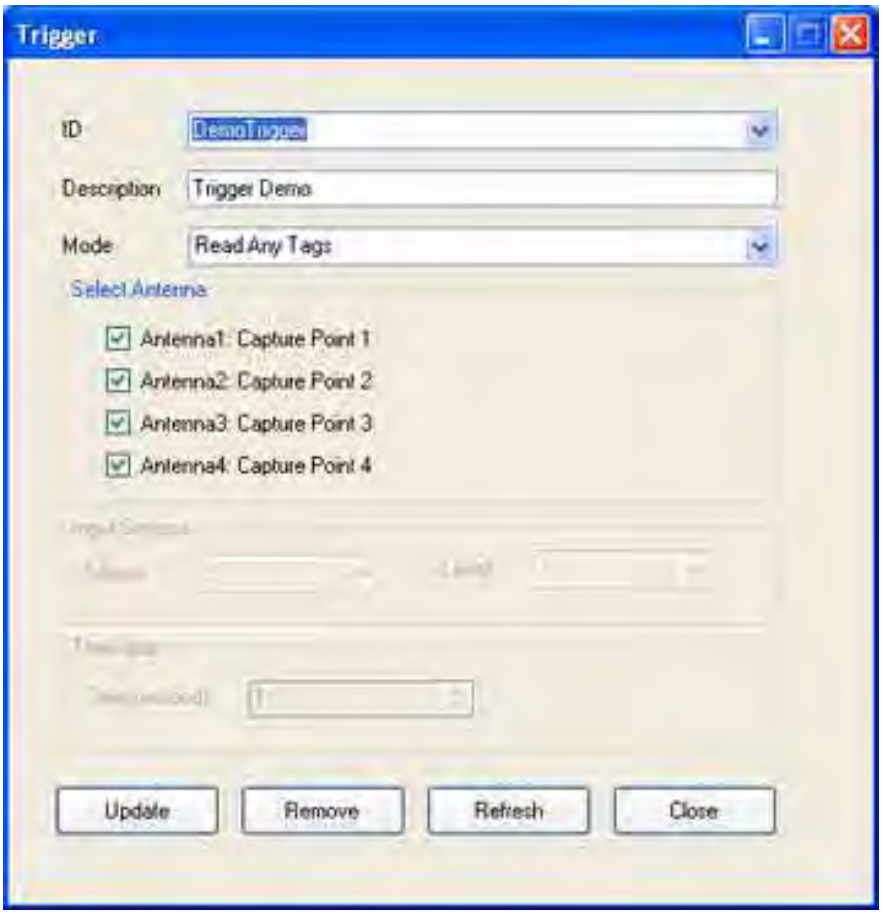


Figure 6-8

4. Add a new “Resultant Action” to use this PC if it is not done before. Please open “Resultant Action Dialog” from pull-down menu. Then enter correct information in the entries and click “Update” button.

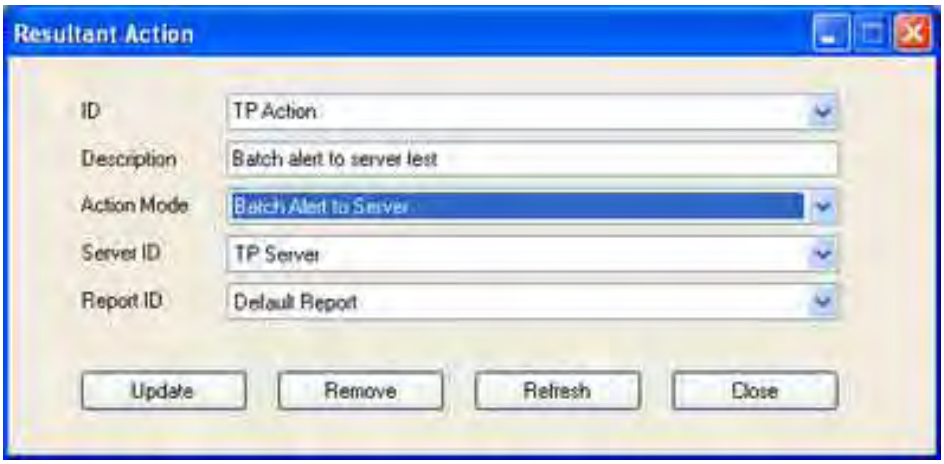


Figure 6-9

Copyright © Convergence Systems Limited, All Rights Reserved

5. Add a new “Event” to use this PC if it is not done before. Please open “Event Dialog” from pull-down menu. Then enter correct information in the entries and click “Update” button.

Figure 6-10

6. The reader is now operated in “Autonomous Time Trigger” mode. Click “Start Read Alert” button to start waiting for notification. If any notification contains tag information, the information will display on table.

Copyright © Convergence Systems Limited, All Rights Reserved

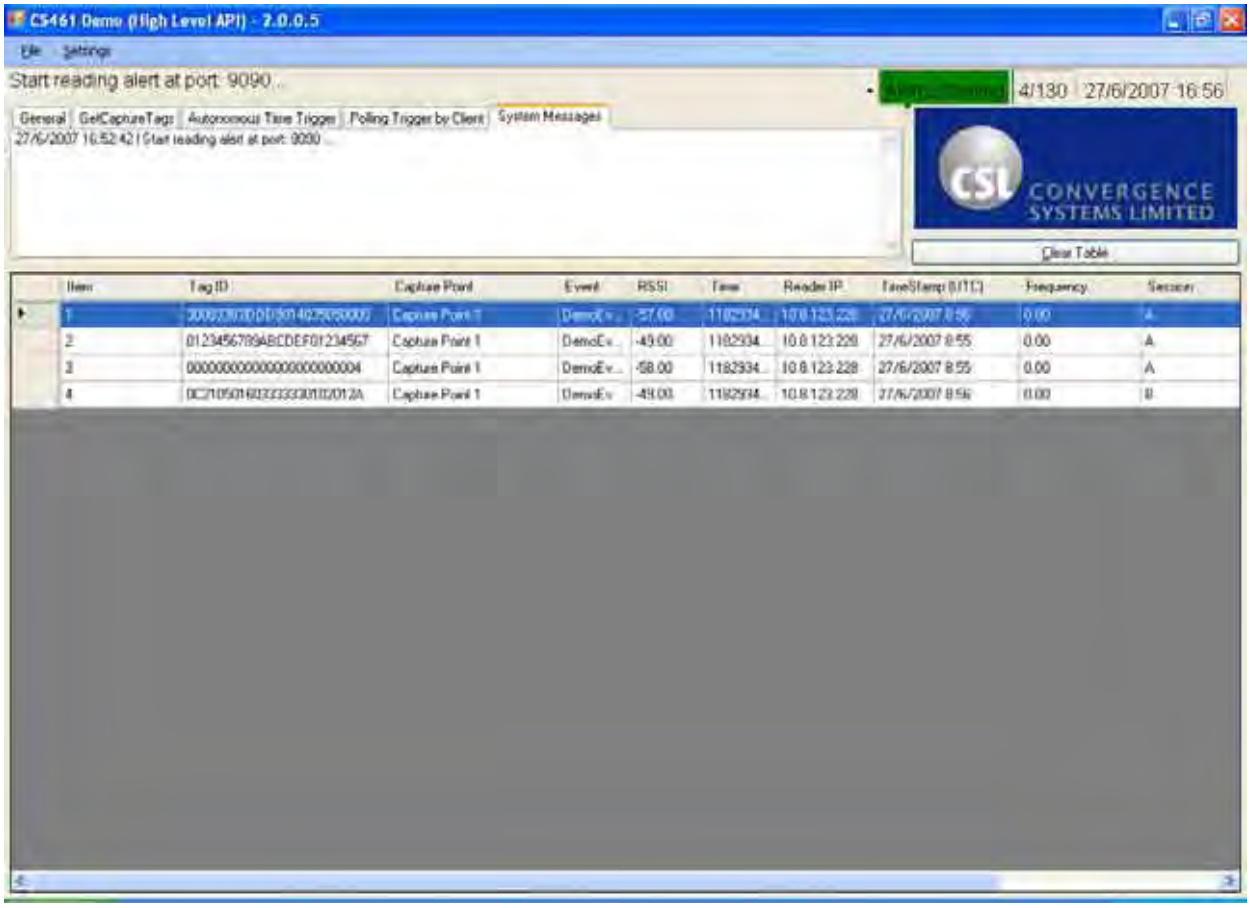


Figure 6-11

Copyright © Convergence Systems Limited, All Rights Reserved

6.1.2.2 Polling Trigger by Client Mode

Reader in this mode will store all received tags information in reader's memory until a "Start Inventory" command is received. When "Start Inventory" command received, the reader will start a new session to store received tags information. The tags information stored in last session will then send to notification server as shown in the following figure.

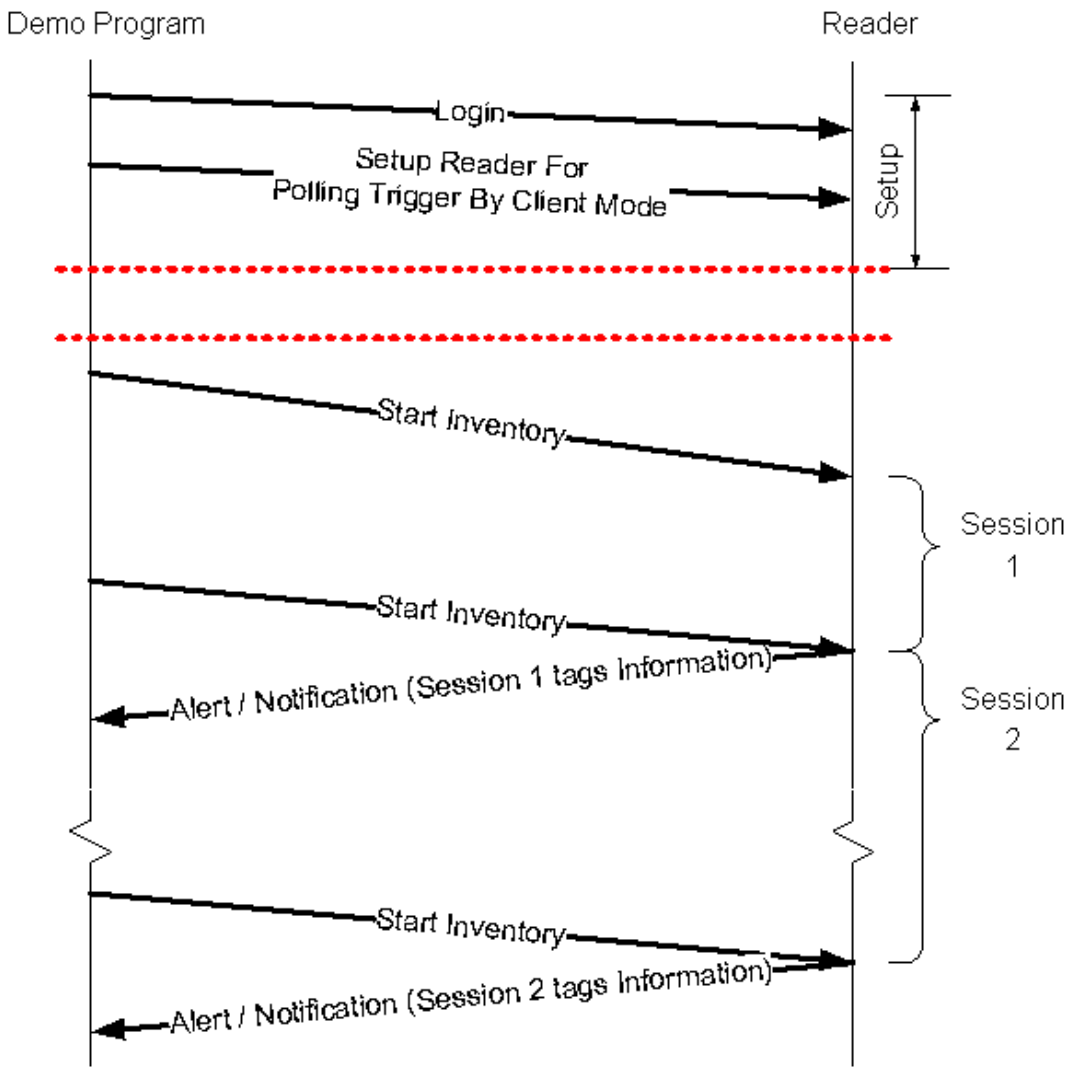


Figure 6-12

Copyright © Convergence Systems Limited, All Rights Reserved

Set the reader to “Polling Trigger by Client” mode

1. Open “Operation Profile Dialog” from pull-down menu. Then, set the “Duplicate Elimination Triggering Method” to “Polling Trigger by Client”.

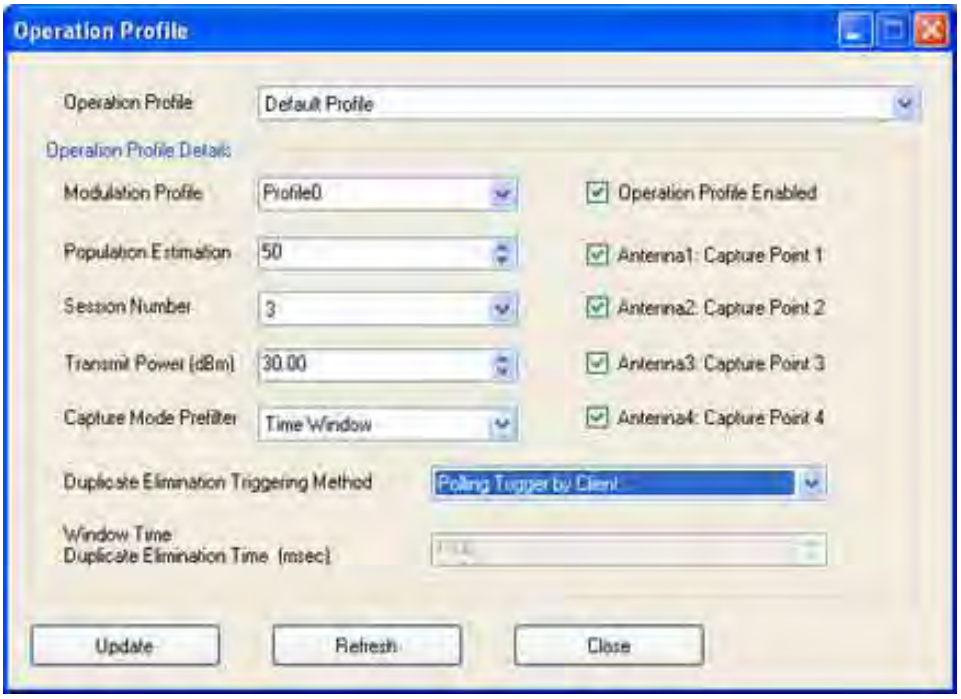


Figure 6-13

2. Please follow step 錯誤! 找不到參照來源。 to step 錯誤! 找不到參照來源。 in chapter 錯誤! 找不到參照來源。 to setup the Trusted Server, Trigger, Resultant Action and Event.

3. The reader is now operated in “Polling Trigger by Client” mode. Click “Start Read Alert” button putting demo program monitoring for notification. Click “Polling” button to send “Start Inventory” command to reader. If there are tags received in last session, demo program will receive notification of tags information and display them in table.

Copyright © Convergence Systems Limited, All Rights Reserved

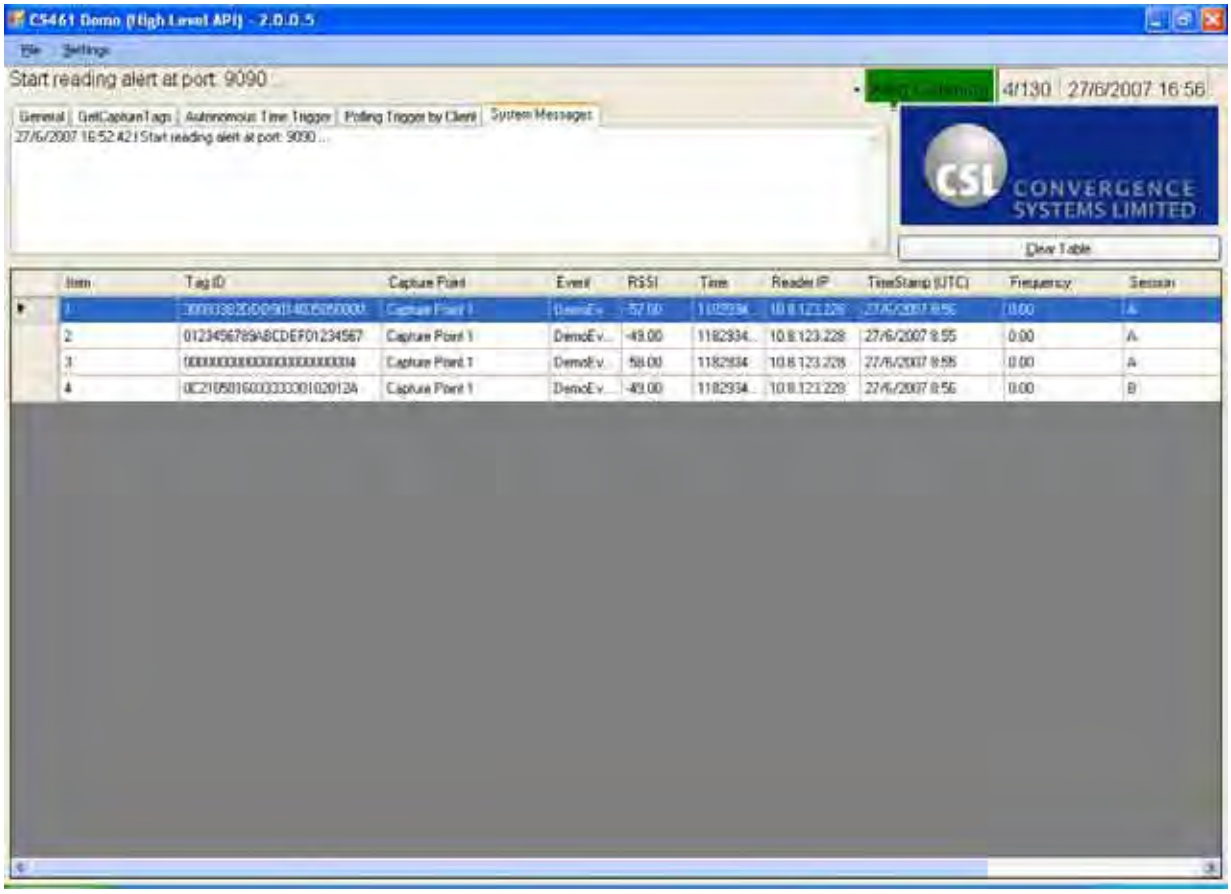
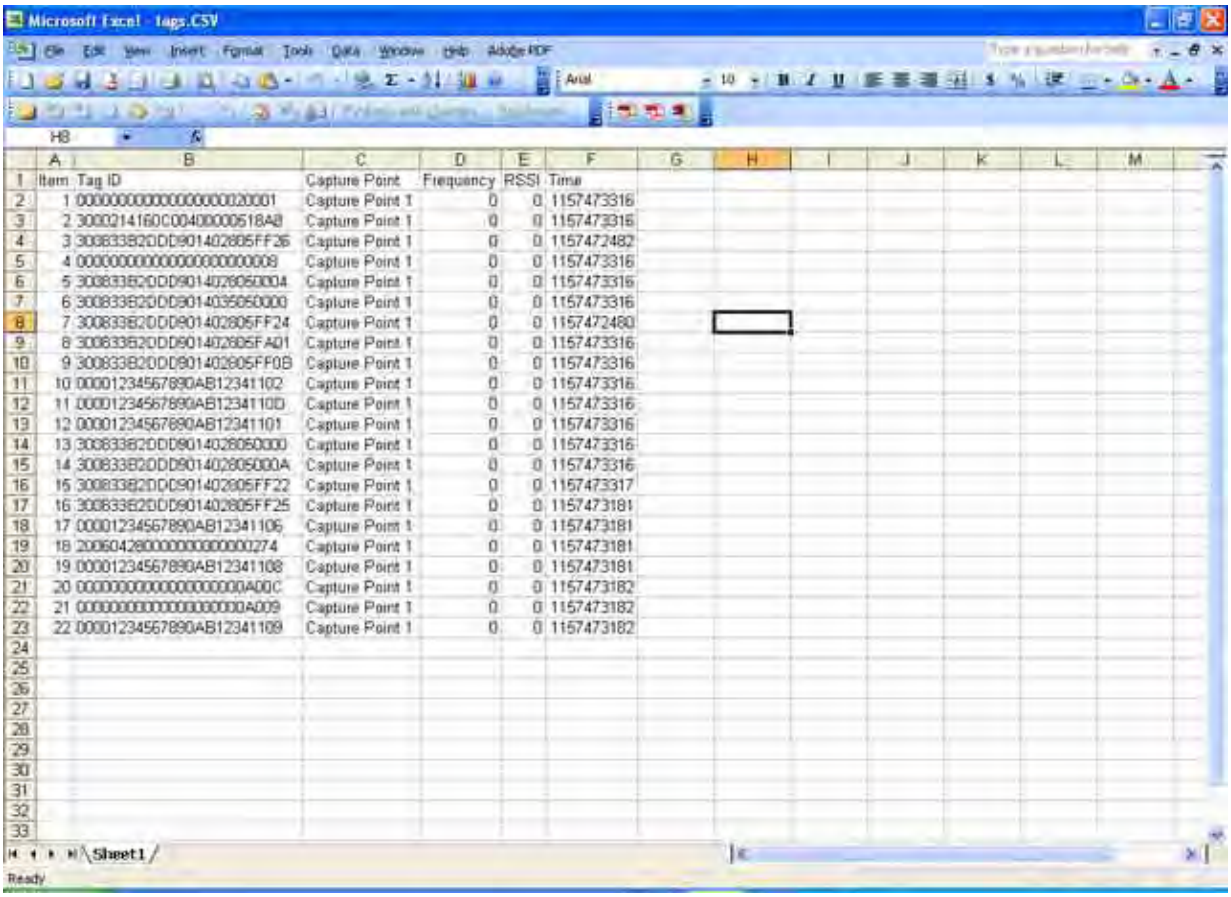


Figure 6-14

Copyright © Convergence Systems Limited, All Rights Reserved

6.1.2.3 Save Read Tags

Received tag information can be stored in a CSV file by clicking on “Save tags to file”. The file can be read by Excel as shown below.



Item	Tag ID	Capture Point	Frequency	RSSI	Time
1	00000000000000000000020001	Capture Point 1	0	0	1157473316
2	3000214160C00400000518AB	Capture Point 1	0	0	1157473316
3	300833B2DDD901402805FF26	Capture Point 1	0	0	1157472482
4	00000000000000000000000008	Capture Point 1	0	0	1157473316
5	300833B2DDD9014028050004	Capture Point 1	0	0	1157473316
6	300833B2DDD9014035050000	Capture Point 1	0	0	1157473316
7	300833B2DDD901402805FF24	Capture Point 1	0	0	1157472480
8	300833B2DDD901402805FA01	Capture Point 1	0	0	1157473316
9	300833B2DDD901402805FF0B	Capture Point 1	0	0	1157473316
10	00001234567890AB12341102	Capture Point 1	0	0	1157473316
11	00001234567890AB12341100	Capture Point 1	0	0	1157473316
12	00001234567890AB12341101	Capture Point 1	0	0	1157473316
13	300833B2DDD9014028050000	Capture Point 1	0	0	1157473316
14	300833B2DDD901402805000A	Capture Point 1	0	0	1157473316
15	300833B2DDD901402805FF22	Capture Point 1	0	0	1157473317
16	300833B2DDD901402805FF25	Capture Point 1	0	0	1157473181
17	00001234567890AB12341106	Capture Point 1	0	0	1157473181
18	20060428000000000000274	Capture Point 1	0	0	1157473181
19	00001234567890AB12341108	Capture Point 1	0	0	1157473181
20	0000000000000000000A00C	Capture Point 1	0	0	1157473182
21	0000000000000000000A009	Capture Point 1	0	0	1157473182
22	00001234567890AB12341109	Capture Point 1	0	0	1157473182

Figure 6-15

Copyright © Convergence Systems Limited, All Rights Reserved

6.2 Low Level API Demo Program

In addition to the web-based interface, a Windows-based program also comes with the reader for users' quick testing (reader must be set in "Low-level API Access Mode"). Moreover, this program allows users to control up to 2 readers simultaneously to demonstrate the Multi-Reader mode (or Dense-Reader mode).

6.2.1 Installing Demo Program

The demo program can be found in the manual CDROM disk and it should be installed onto a PC before using:

1) Double-click the file "CSLReaderDemoXXX.msi" on the manual CD, where XXX is the version number.
2) Follow the instructions to install the program on your PC
3) After installation, an icon "ReaderDemo XXX" appears on your PC. Double-click on it to run the demo program.

6.2.2 Configuring Reader(s)

When the program opens, the first (and if connected, a second) reader's IP address must be entered in the program.

Copyright © Convergence Systems Limited, All Rights Reserved

Figure 6-16

This is done by clicking the "Configure Readers" button. The screen shown below should open. Enter the reader's IP address in the appropriate window.

Copyright © Convergence Systems Limited, All Rights Reserved

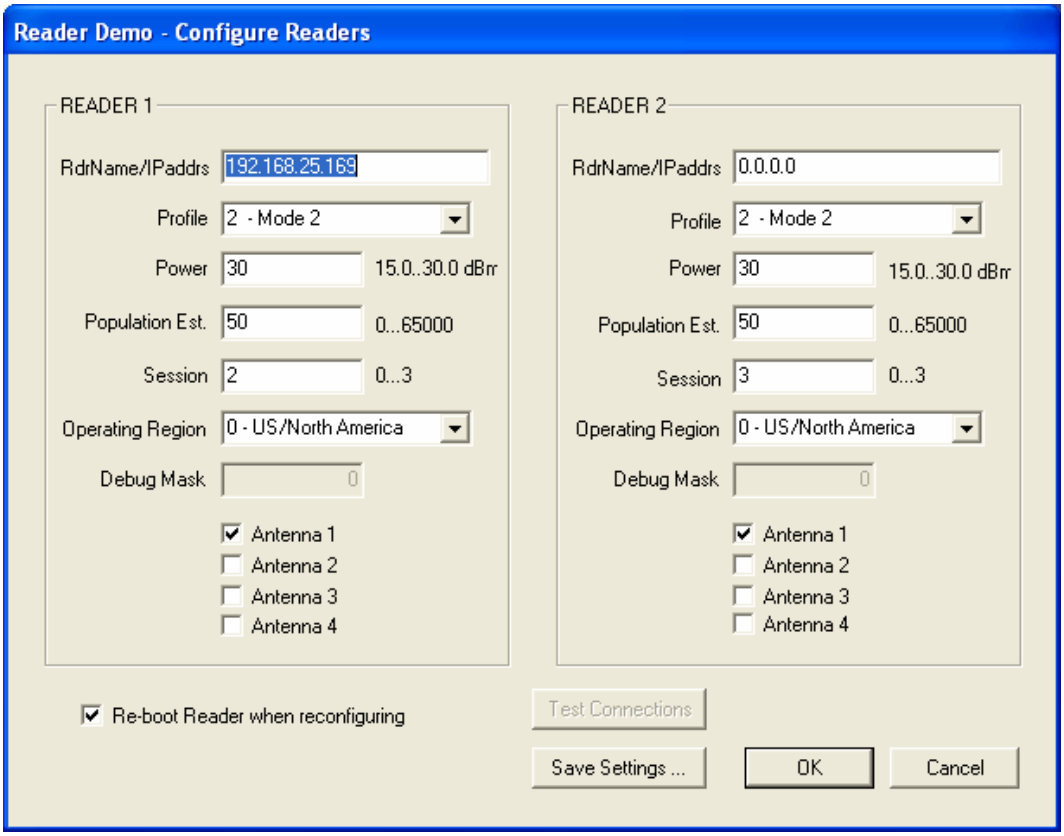


Figure 6-17

At this point, the reader operating mode can be selected. Presently, 4 modes are supported by the reader as shown in the table below.

Operating Mode	Reader Parameters
Mode 0: (Max Throughput)	Tari 7.14 us / PIE 1.5:1 / Fwd Modulation PR-ASK / PW 0.5 (long) / LF 640kbps / Rev Mod FM0
Mode 1:	Tari 12.5 us / PIE 1.5:1 / Fwd Modulation PR-ASK / PW 0.33 (short) / LF 160kbps / Rev Mod FM0
Mode 2: (Dense Reader mode)	Tari 25.0 us / PIE 2.0:1 / Fwd Modulation PR-ASK / PW 0.5 (long) / LF 256kbps / Rev Mod Miller M=4
Mode 3: (Dense Reader mode)	Tari 25.0 us / PIE 2.0:1 / Fwd Modulation PR-ASK / PW 0.5 (long) / LF 256kbps / Rev Mod Miller M=8

Select the desired mode. For dock door applications where two readers are used, Mode 2, (Dense Reader) must be selected.

It is also very important to select, by checking the appropriate box, the antennas that are

Copyright © Convergence Systems Limited, All Rights Reserved

currently connected to the reader. The program will not operate if an antenna box is checked and no antenna is connected to that port. However no damage to the reader will result.

Other reader parameters can also be selected in this window.

Power	Transmit power of the reader (from 15dBm to 30dBm)
Population Est.	The estimated maximum number of tags to be read by the reader at the same time Please input a value as accurate as possible because it can optimize the performance of tag read
Session	Session number that the program connect with the reader. It should be unique for each reader
Operating Region	The region that the application is in operation

After configuring the reader(s), press the “Save Settings …” button and then “OK” button to close this window. It will return to the main screen and attempt to communicate with the reader and setup the selected parameters. If successful, the bottom of the screen will indicate **Reader 1: Connected**. (or “Reader 2: Connected” also if multi-reader is set).

6.2.3 Reading Tags

Once the readers are connected, it will read tags placed in the field of the reader. The figure below shows a successful read operation.

Copyright © Convergence Systems Limited, All Rights Reserved

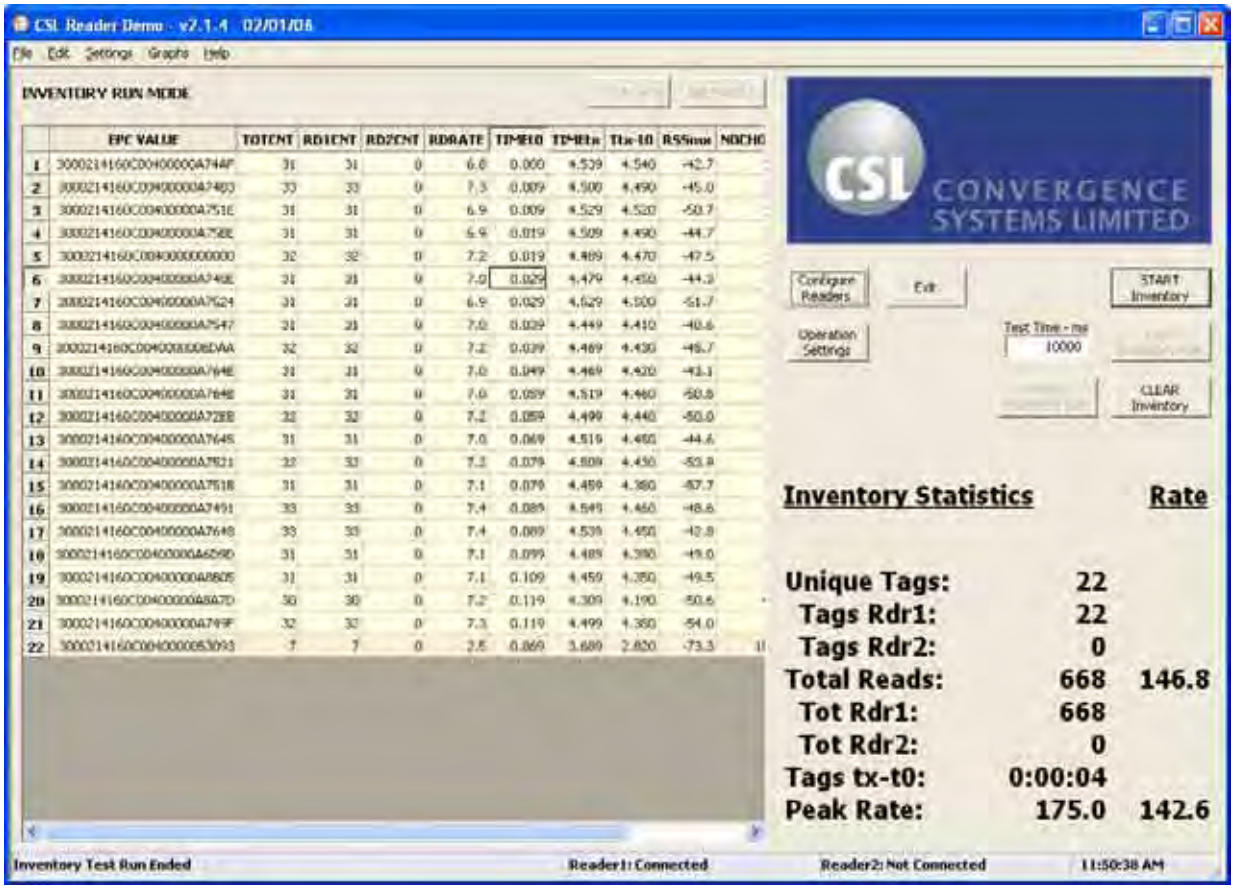


Figure 6-18

The program defaults to a 10 second run, after which the reader will stop. This parameter can be changed by selecting the desired operating time (in milliseconds) in the “Test Time – ms” field. Entering the value “0” results in continuous operation.

To stop the reading operation, press the “STOP Inventory Run” button.
To start the reading operation, press the “START Inventory” button.
To clear the list of read tag on the left hand side, press the “CLEAR Inventory” button.

To study the detail information of each read of a specific tag, click on the EPC value of that tag on the list and then right-click the mouse. A detail record of every read of that tag is displayed.

Copyright © Convergence Systems Limited, All Rights Reserved

	TIMESTAMP	RUNNUM	RDRID	FREQ	RSSI
1	1143205055.352	0	0	910.75	-62.19
2	1143205055.388	0	0	910.75	-62.04
3	1143205055.650	0	0	904.75	-62.58
4	1143205055.678	0	0	904.75	-62.38
5	1143205055.804	0	0	925.25	-63.92
6	1143205055.827	0	0	925.25	-63.92
7	1143205056.053	0	0	918.25	-63.12
8	1143205056.070	0	0	918.25	-63.20
9	1143205056.136	0	0	918.25	-63.12
10	1143205056.158	0	0	918.25	-63.20
11	1143205056.426	0	0	909.75	-62.86
12	1143205056.447	0	0	909.75	-62.74
13	1143205056.689	0	0	911.75	-63.46

Print... Save... Cancel

Figure 6-19

6.2.4 Tag Reading Graph

Different kinds of graph for the tag reading operations of the reader(s) can be generated for further analysis. Click the “Graphs” on the top to select different kinds of graph.

Copyright © Convergence Systems Limited, All Rights Reserved

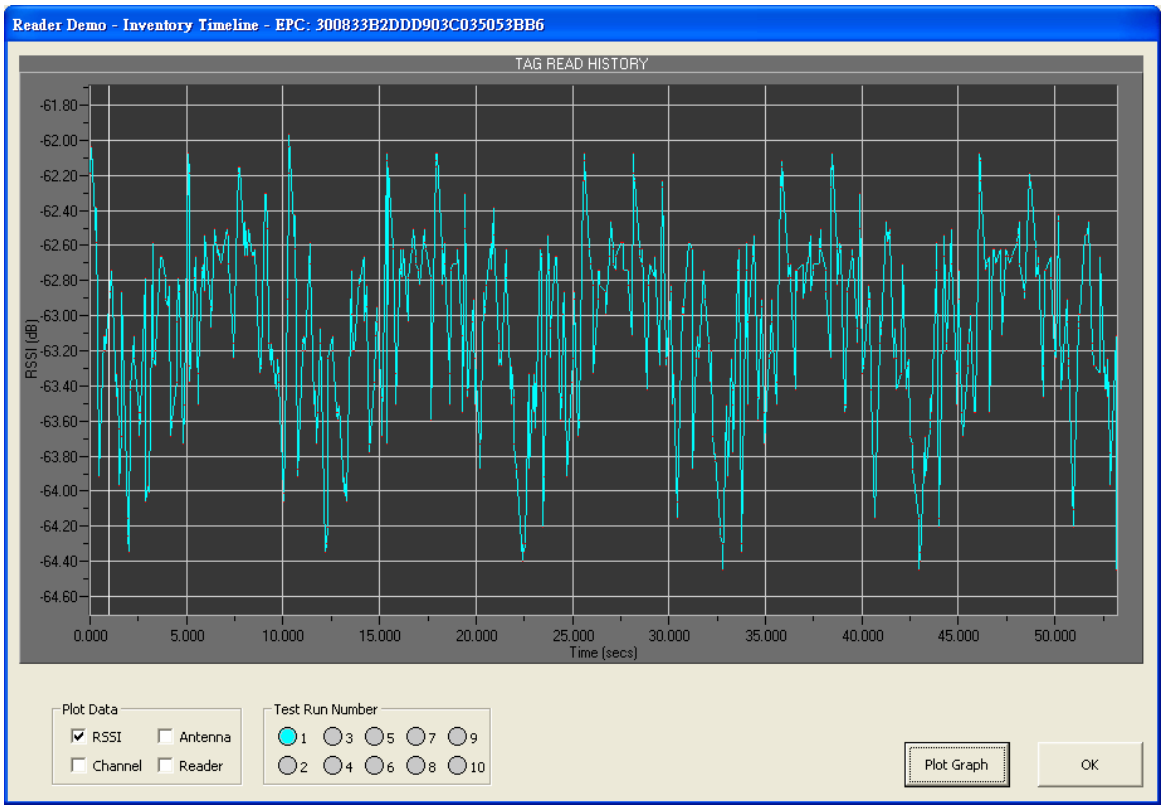


Figure 6-20 Tag Read History

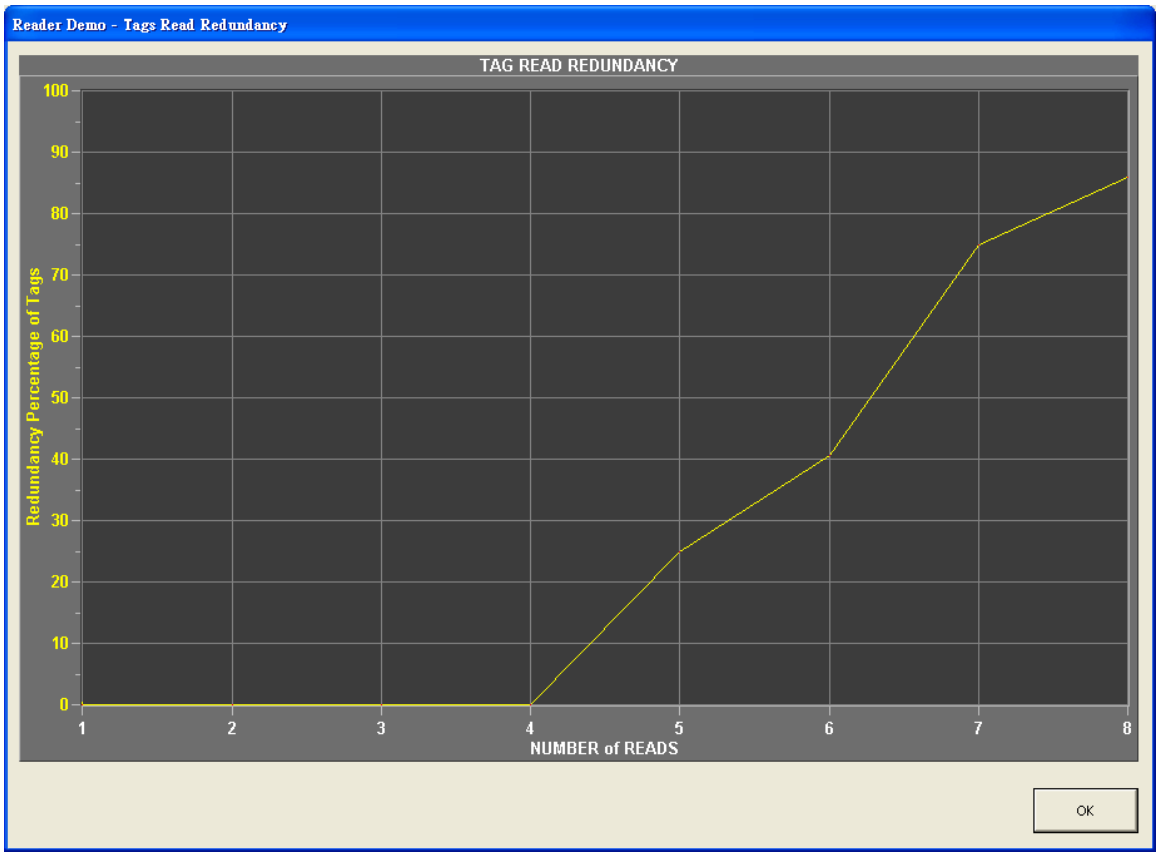


Figure 6-21 Tag Read Redundancy

Copyright © Convergence Systems Limited, All Rights Reserved

7 Usage Tips for CS461

7.1 Introduction

The objective of this chapter is to recommend the best practices of using the CSL CS-461 Reader. The following areas will be covered in this document

- General usage
- Write tag
- Event and alert
- System

7.2 General Tips

1. Disconnect or connect the antenna after power off the reader to protect the antenna
2. If antenna port is not used, disable it in Operation Profile. Otherwise it may degrade the reader performance.

7.3 System Tips

1. Rebooting time: system does not reboot immediately after the restart command is sent. It will reboot in about 1 minute. All reader operations should be performed after the reader reboots.

7.4 Write Tag Tips

1. Always define halt filter to prevent the same tag being written repeatedly.
2. Be careful when defining the halt filter. If the condition is always true, the write tag operation may perform indefinitely without being stopped.

Copyright © Convergence Systems Limited, All Rights Reserved

7.5 Event Engine Tips

1. The alert pattern of Batch Alert to Server and Instant Alert to Server are different. For Batch Alert to Server, tags are sent in batch at the end of inventory cycle or time window and then followed by a batch end notification. For Instant Alert to Server, a batch end notification is sent first at the beginning of inventory cycle or time window, and then each tag is reported instantly at the time it reads. Server application may handle differently for these two alert patternsThe alert pattern of Batch Alert to Server and Instant Alert to Server are different. For Batch Alert to Server, tags are sent in batch at the end of inventory cycle or time window and then followed by a batch end notification. For Instant Alert to Server, a batch end notification is sent first at the beginning of inventory cycle or time window, and then each tag is reported instantly at the time it reads. Server application may handle differently for these two alert patterns
2. If two resultant actions are defined for an event, note that the second one will perform right after the first one is invoked instead of after its completion.
3. If there are two resultant actions in an event and one of them is I/O control, set the I/O control action to be the first one. This will result in a perception of faster response.The maximum number of enabled events allowed for one reader is two. Make sure there is no conflict between the events. E.g. the same input sensor is used as the enable trigger of one event and the disable trigger of another event.

Copyright © Convergence Systems Limited, All Rights Reserved

8 RFID Cookbook

8.1 Introduction

RFID (radio frequency identification) is a wireless means to obtain a unique ID that can identify a product (similar to barcode that however requires optical line of sight). Since 2004, it was applied by companies in USA and Europe successfully to various business processes and brought major cost benefits. Because of the success of these early adopters, such as Walmart (USA) and Mark & Spencer (Europe), there is a growing trend throughout the world to replace barcode (or augment) with RFID. The advantages of RFID over barcode are widely publicized, consisting of the following:

Features	RFID	Barcode
Line of Sight	Line of sight is not required	Must be line-of-sight visible – items must be tediously separated out for reading, very inconvenient
Storage	Store data up to 1 Kbyte	No storage capability
Anti-Counterfeit Ability	Hard to counterfeit, hard to find (can be stowed inside item)	Easy to counterfeit, always exposed outside and therefore easy to copy
Processing Speed	Automatic processing possible at very high speed	Processing has to be manual in most cases, with very low speed and throughput
Bulk Reading	Many tags can be read at the same time – virtually parallel reading	Must be read sequentially
Durability	Durable, usually safely stowed inside item.	Easily scratched, wrinkled or wetted beyond reading.

Copyright © Convergence Systems Limited, All Rights Reserved

RFID can be applied with the following purposes:

1. Supply chain optimization
2. Asset tracking
3. Inventory control
4. etc.

Benefits of RFID include:

1. Increase supply chain velocity
2. Reduce human involvement (cost, error, hiring cycle and other issues)
3. Enhanced visibility (tracking, scheduling, planning)
4. Enhanced security (total visibility monitoring, zonal tracking)
5. Real time supply chain re-route (dynamic multi-destination fulfillment)
6. etc.

Physical locations where RFID can be applied include:

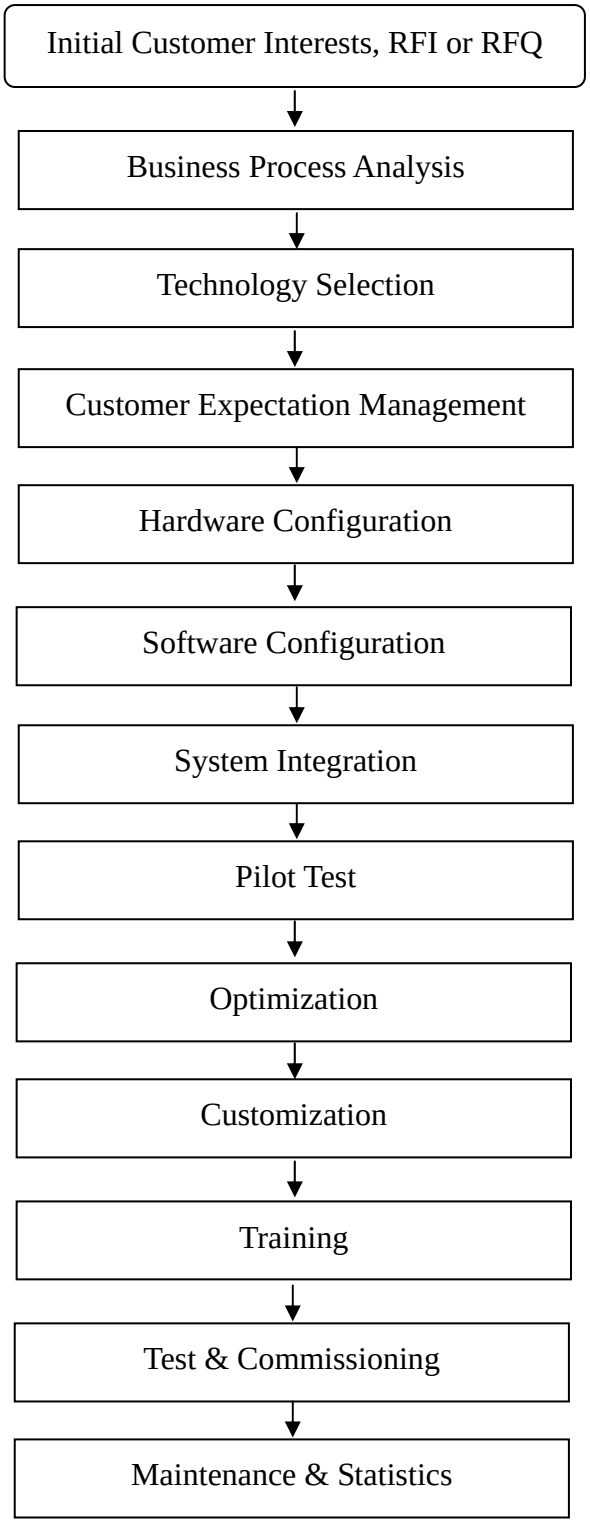
1. Distribution centers
2. Warehouses Shelves
3. Warehouse Loading/Unloading Zone (Yard Management)
4. Retail shops in conjunction with fulfillment center
5. Returns & warranty processing office
6. Vehicle windshields
7. etc.

It is widely believed that the adoption of RFID will happen in the following sequence in terms of company category:

1. Mandate affected units (suppliers to Walmart, DoD, etc.)
2. High value products
3. Fast moving assets
4. etc.

The application of RFID to a company or a group of companies in a supply chain has to be executed systematically and methodically. The following is a flowchart that describes a typical application process:

Copyright © Convergence Systems Limited, All Rights Reserved

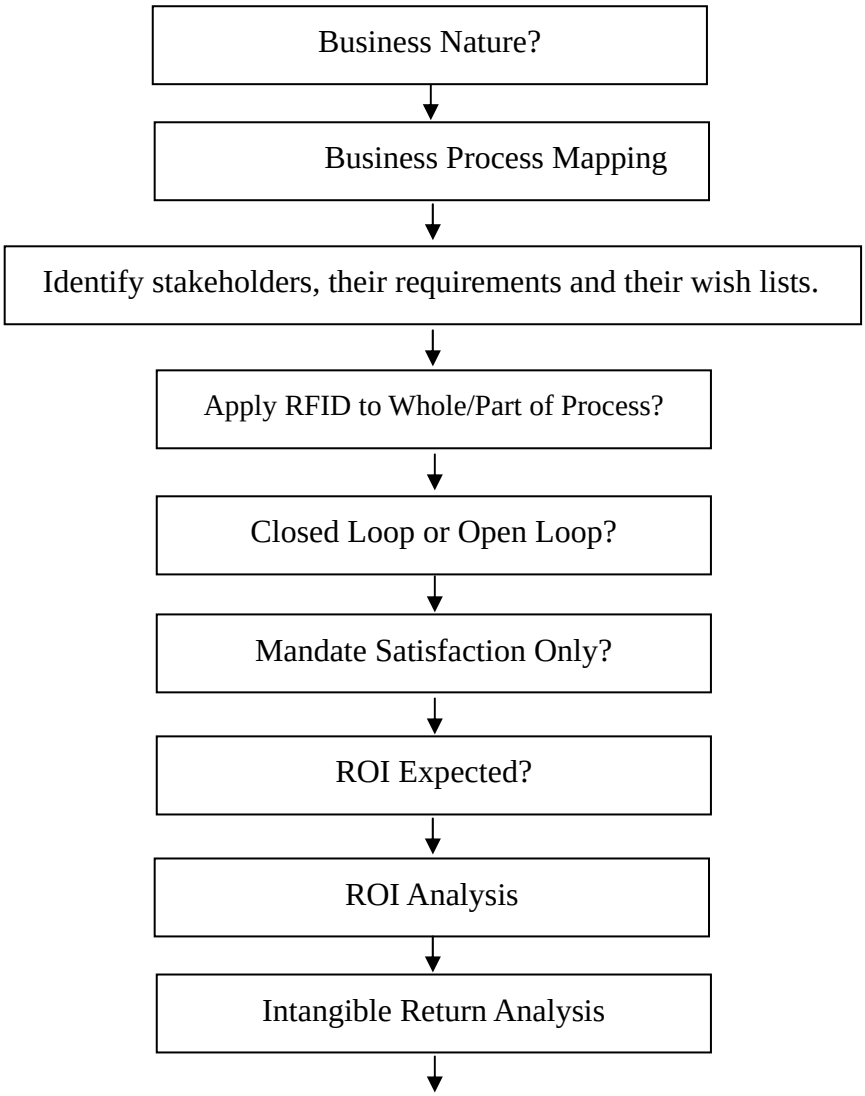


Copyright © Convergence Systems Limited, All Rights Reserved

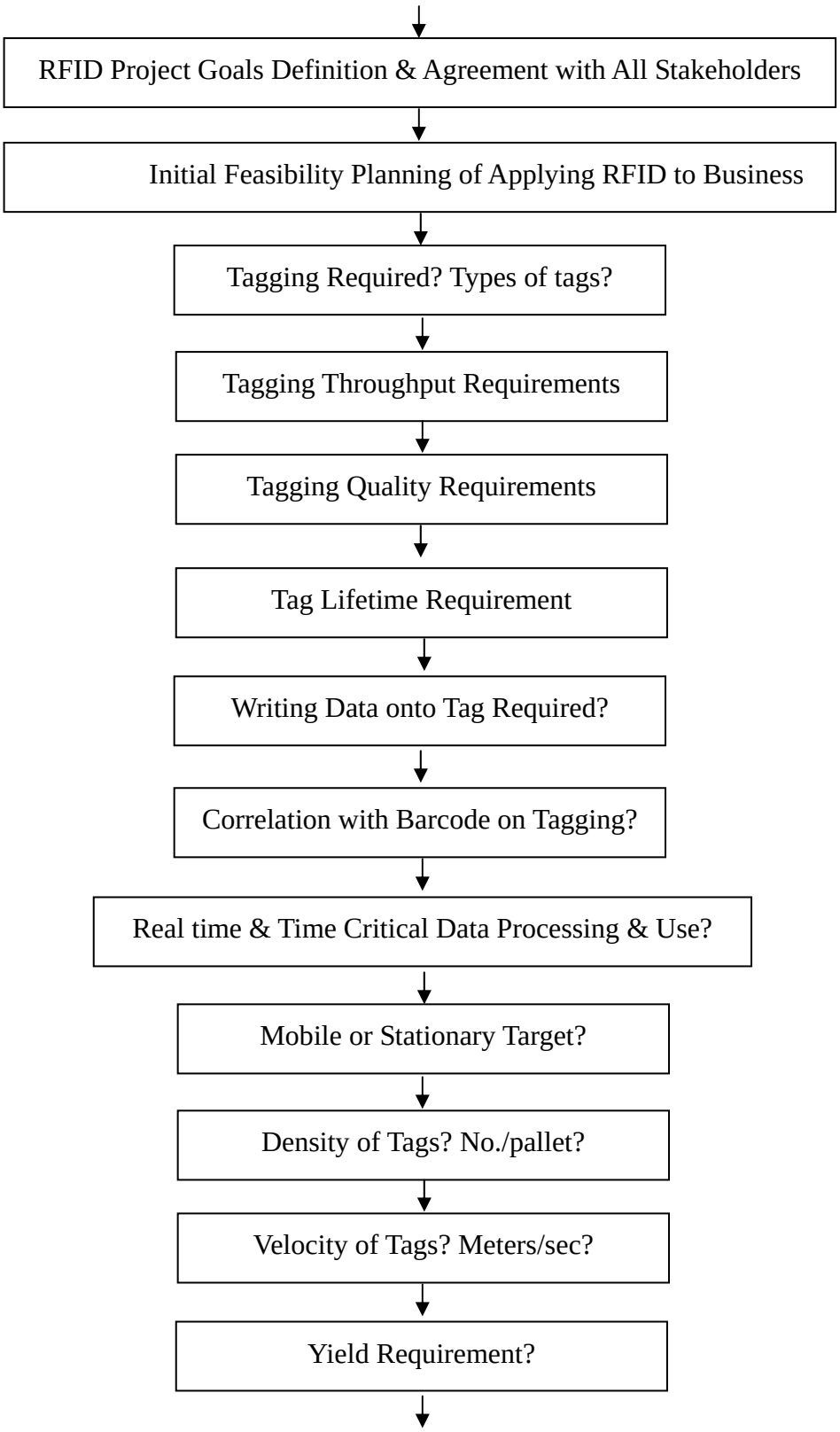
8.2 Application Details

8.2.1 Business Process Analysis

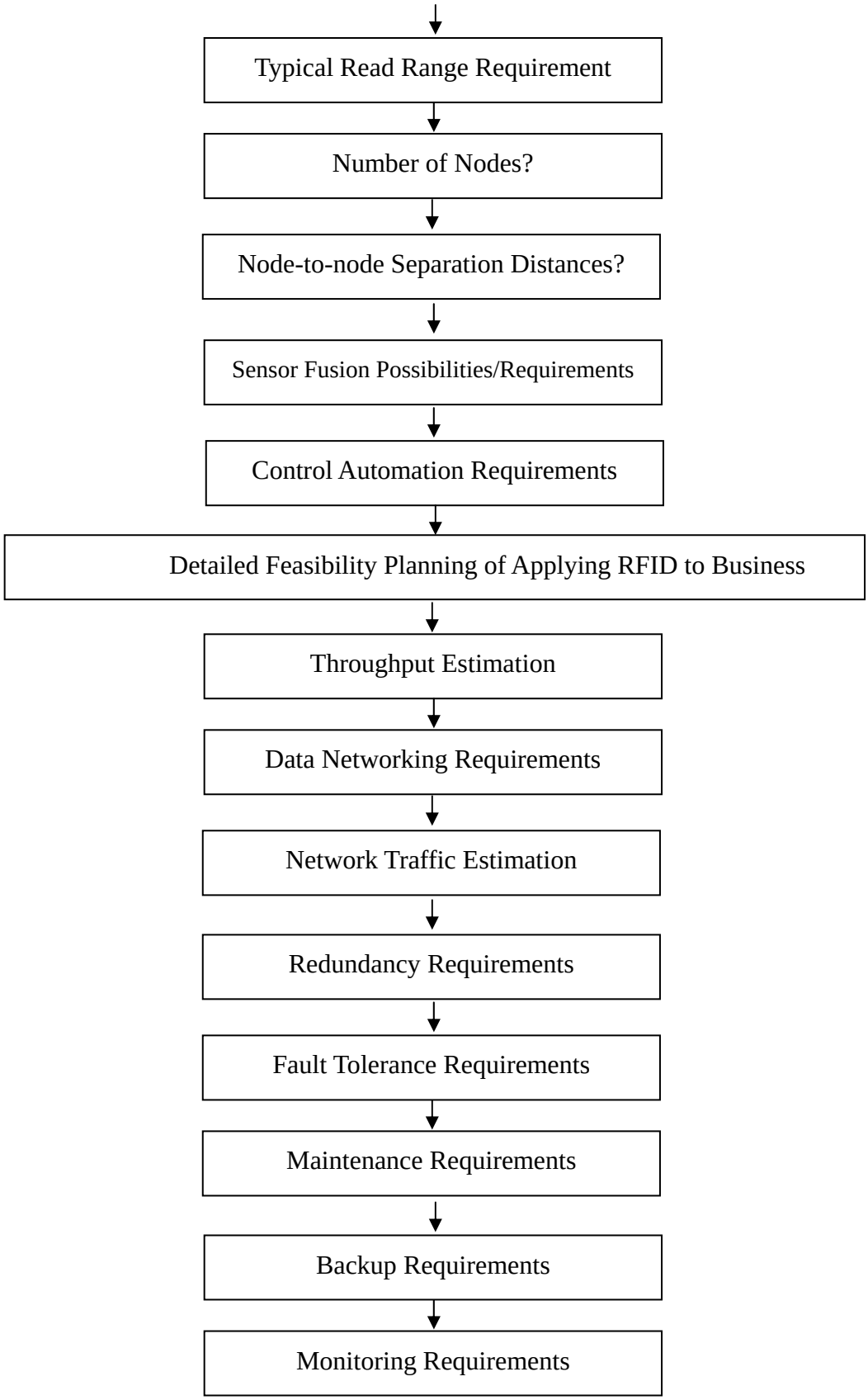
The business process of the customer must be analyzed carefully to find places where the RFID tagging and reading can occur. The system integrator may be applying RFID to the whole process or may only be able to apply RFID to part of the process. The most important principle is NOT to force change the business process to adapt for RFID implementation, but to have RFID implementation slip in as effortlessly and as un-noticeably as possible.



Copyright © Convergence Systems Limited, All Rights Reserved



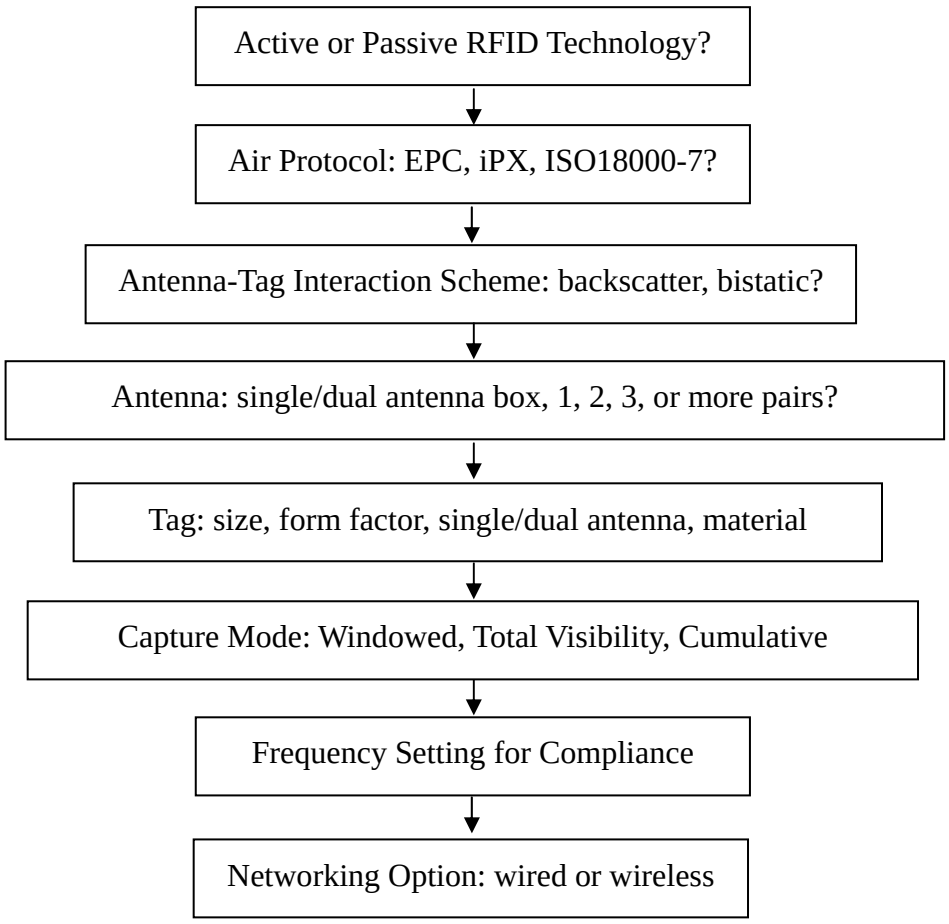
Copyright © Convergence Systems Limited, All Rights Reserved



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.2 Technology Selection

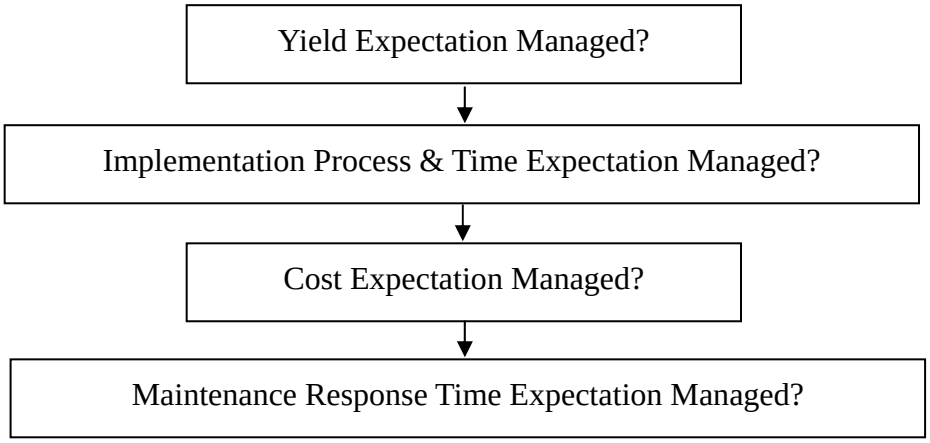
Once the points where the business process allows for RFID implementation is found, the most appropriate technology must be chosen for the job. The following are questions to help you choose the appropriate technology:



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.3 Customer Expectation Management

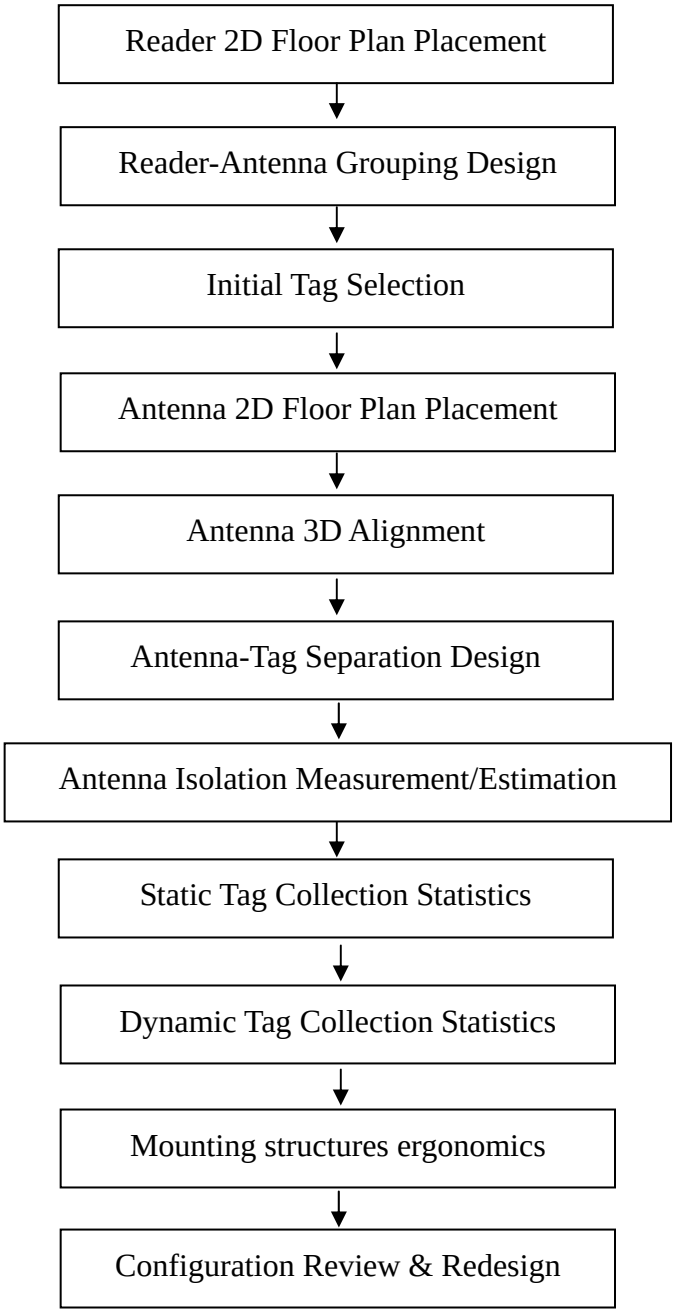
Customer expectation must be well managed. It is the job of the system integrator, particularly the sales person, to warn the customer away from expecting perfect scores. The truth is, even if 100% read is not achieved, the user can still benefit (in the sense of ROI, efficiency, lead time, cycle time, etc.) to a substantial extent. It is this extent that should be considered as the result, not a 100% score. It is almost like getting married to a man or woman – you will never find the perfect half, but even if she or he is not perfect, you still get to enjoy from the marriage.



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.4 Hardware Configuration

Hardware configuration consists of designing and defining what reader, antenna and tag combination will be implemented at each of the nodes in the business process. It is not a pure drawing board exercise, as some kind of minimally realistic testing must be implemented even at this stage to help better define the hardware configuration that in turn can give more insight for software configuration and system integration.



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.5 Software Configuration

Software configuration of the reader is very important – it ensures the reader will operate exactly as the business process requires, not more or not less.

The following page has a flowchart that the system integrator needs to go through in order to set up the software.

The first step is to configure the users parameter, such as operator name, ID, password, etc. The second step is to configure the networking parameters, such as IP addresses, access point SSID, etc. The third step is to configure system parameters, such as reader ID, frequency setting, tag baud rate, capture mode, etc.

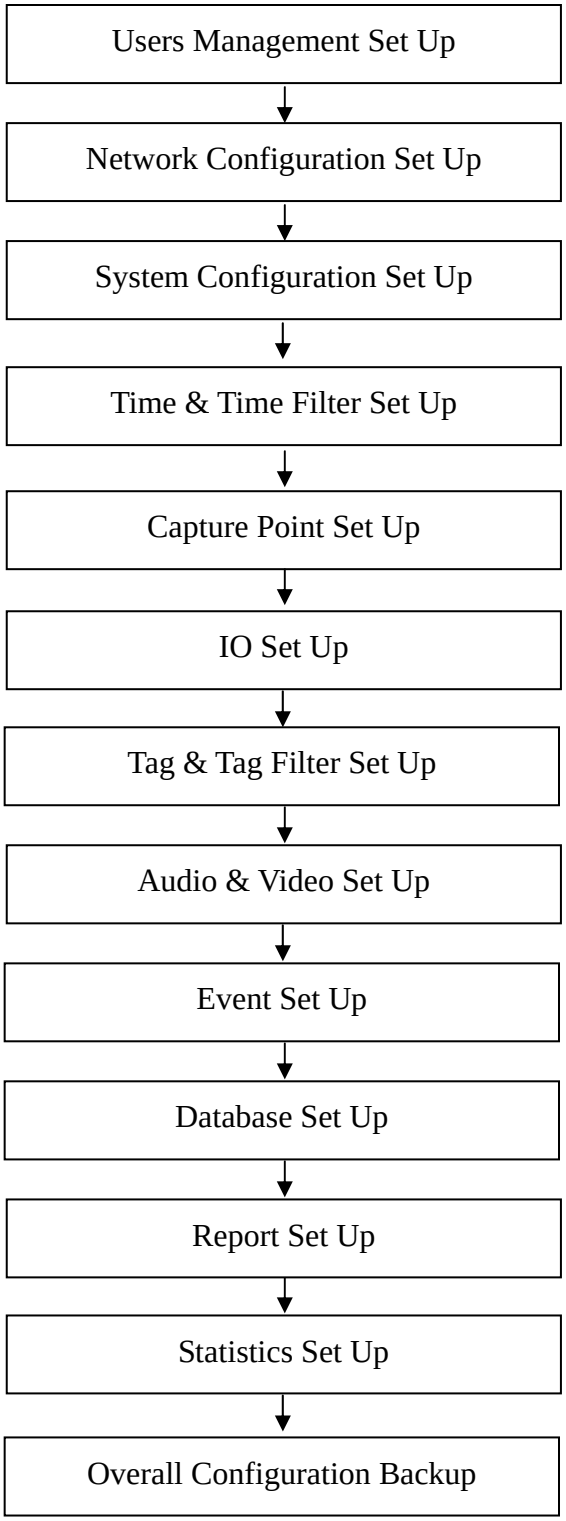
The third step is to configure time and time filter, such as system date and time (hour, minute and second), time filter (define various time intervals, time slots, repeat modes), etc. The fourth step is to configure capture point, such as capture point type, capture point area, capture point details.

The fifth step is to configure IO, such as sensor input name, control output name, default positions, etc. The sixth step is to configure tag and tag filtering, such as tag group, tag filter, etc. The seventh step is to configure audio and video, such as audio messages and video messages resident path (remote or local).

The eighth step is to configure event, such as triggering logic, resultant action, event sequencing, etc. The ninth step is to configure database, such as database fields, etc. The tenth step is to configure report, such as report definition, etc.

The eleventh step is to configure statistics, such as parameters for long term monitoring, etc. The twelfth step is to back up the set up into a standard configuration set up file.

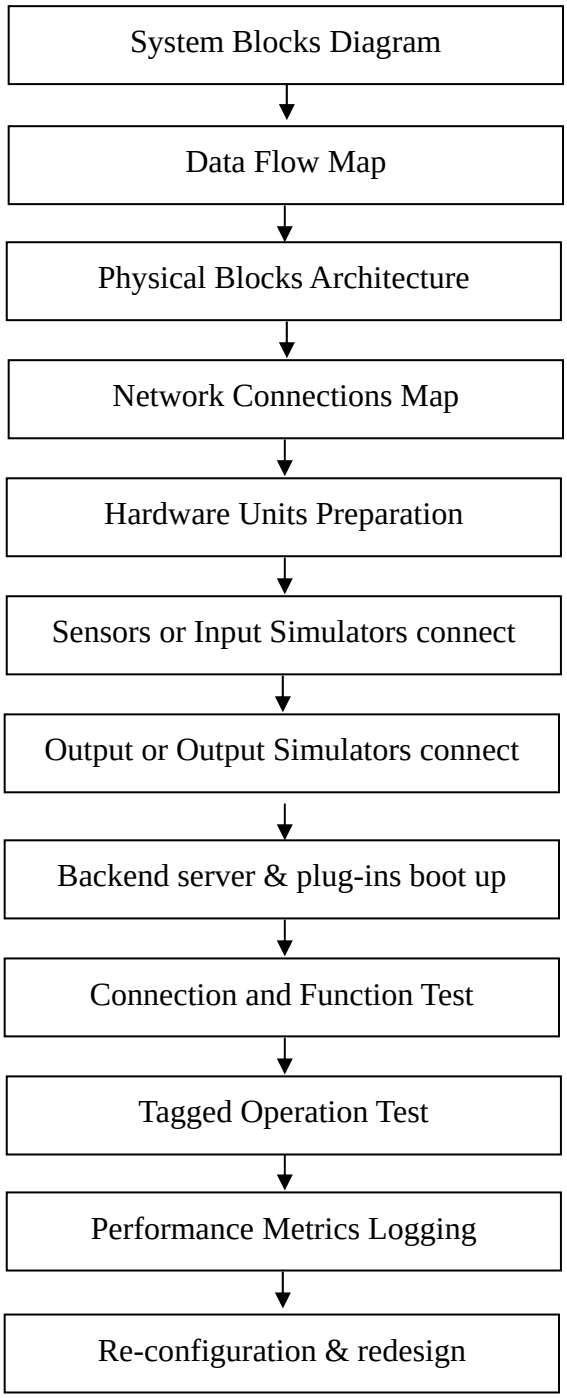
Copyright © Convergence Systems Limited, All Rights Reserved



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.6 System Integration

The actual system integration should most desirably be carried out in two steps: 1. in house integration and test; 2. onsite integration and test.



Copyright © Convergence Systems Limited, All Rights Reserved

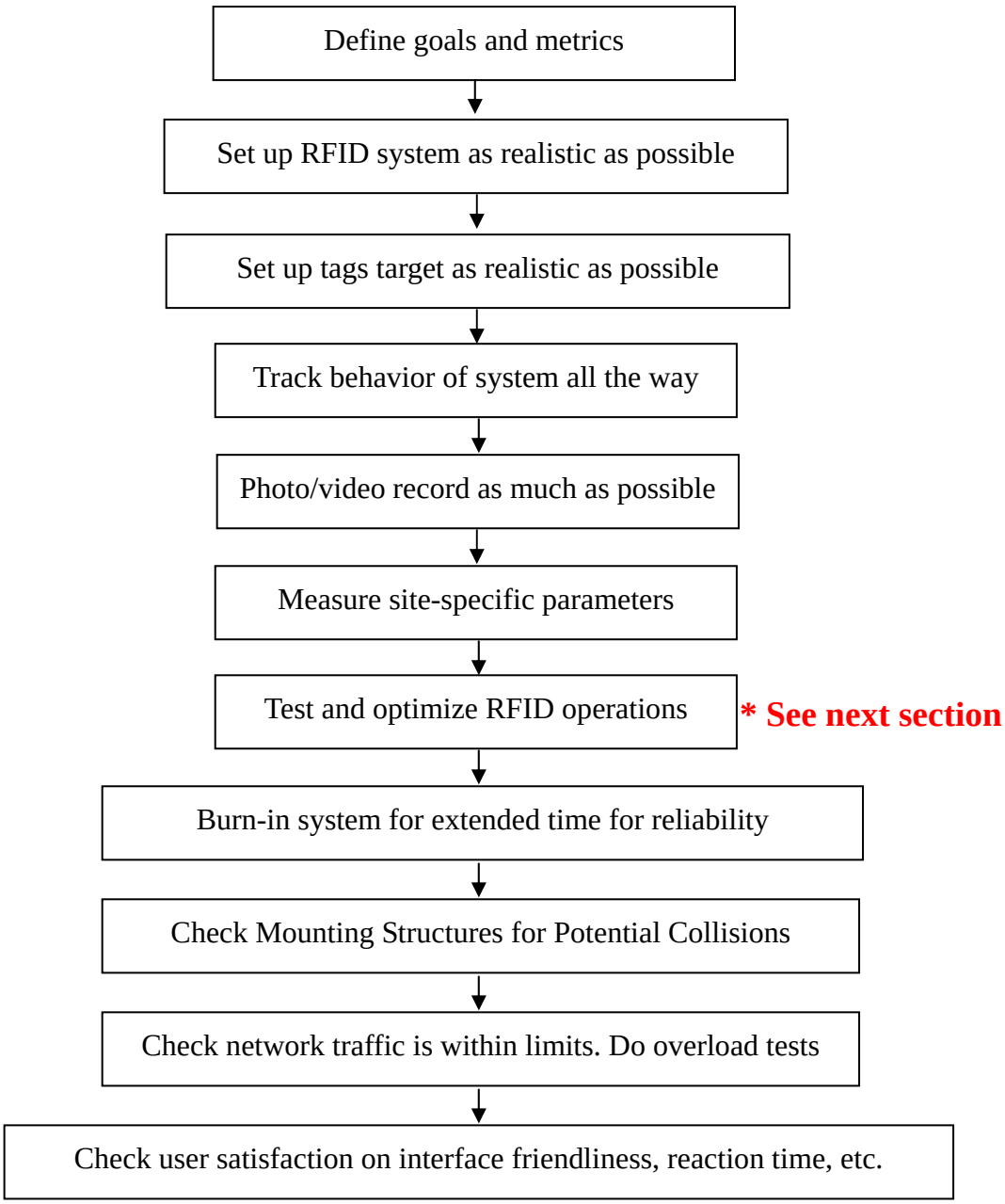
8.2.7 Pilot Test

Pilot test must of course be done on site. The unique building infrastructure and environment of the end-customer venue can result in dramatically different performance (worse, usually) scores compared to that in the system integrator's own office. Therefore pilot test must be done on site.

The system integrator, when testing the set up in end-customer's venue, should endeavor to put the set up directly at the position that it plans to be, or in a place that most closely resembles that of the final site. If the site does not run round-the-clock shifts, then it is OK to do the initial testing when it is off-shift and temporarily clearing up the site for testing (if something is in the way). Eventually when good enough results are obtained through tuning and optimization in off-shift time, then the testing should be conducted in the actual shift when the operation will happen in the future. The emphasis on having the environment as real and true as possible is due to the fact that wireless emission is a very site specific and dynamic event. The propagation and scattering behavior is different from site to site. The noise floor can be different in the day and in the night. There is no pilot test better than doing it right at the spot and right at that time.

The following are basic steps for pilot testing (please also refer to next section of optimization):

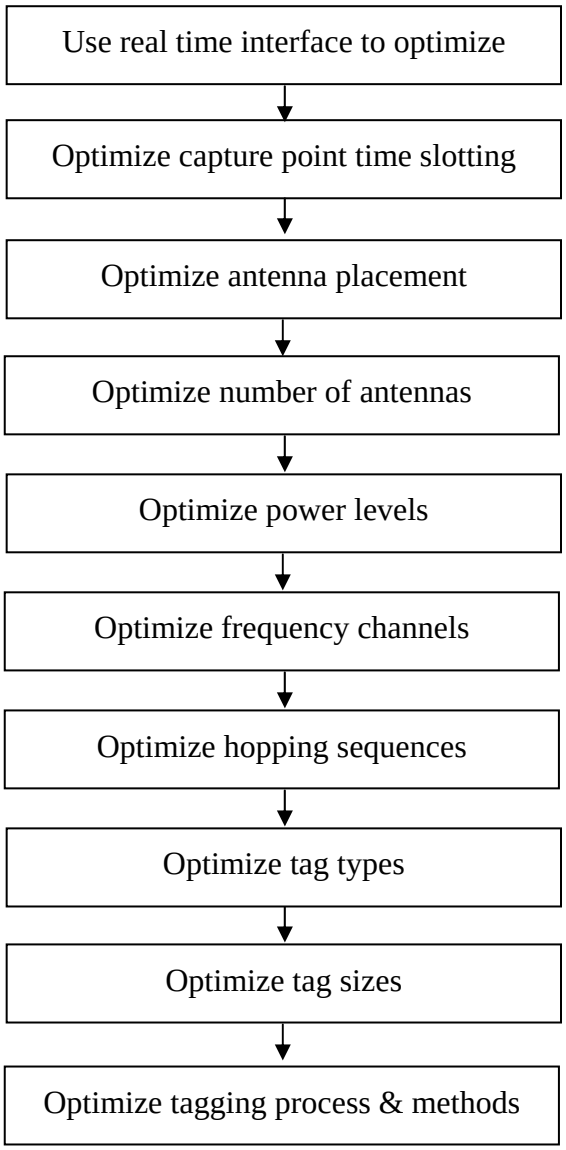
Copyright © Convergence Systems Limited, All Rights Reserved



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.8 Optimization

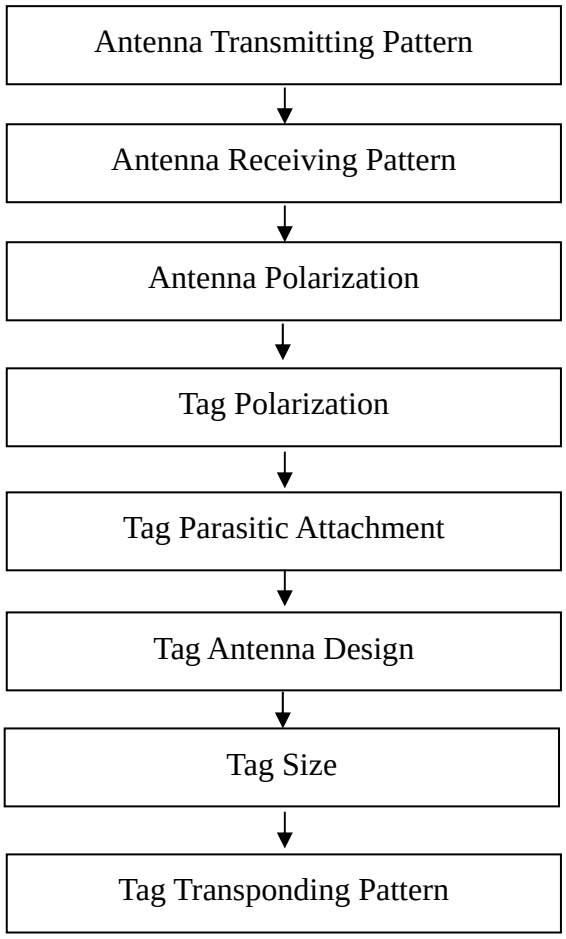
Optimization of the performance of the RFID application in business processes is the most difficult step. It is in this step where the variation of performance caused by the law of physics has to be tackled. The following are a few questions that may help. However, due to the unfortunate fact that RFID application involves too many topics: RF transmitter circuits, antennas, propagation (static and dynamic), scattering (backscatter and bistatic scattering), RF receiving circuits, software (all layers), it is not an easy task to give a “10 steps to successful RFID implementation” rule based implementation guideline that works in all environment!



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.9 Customization

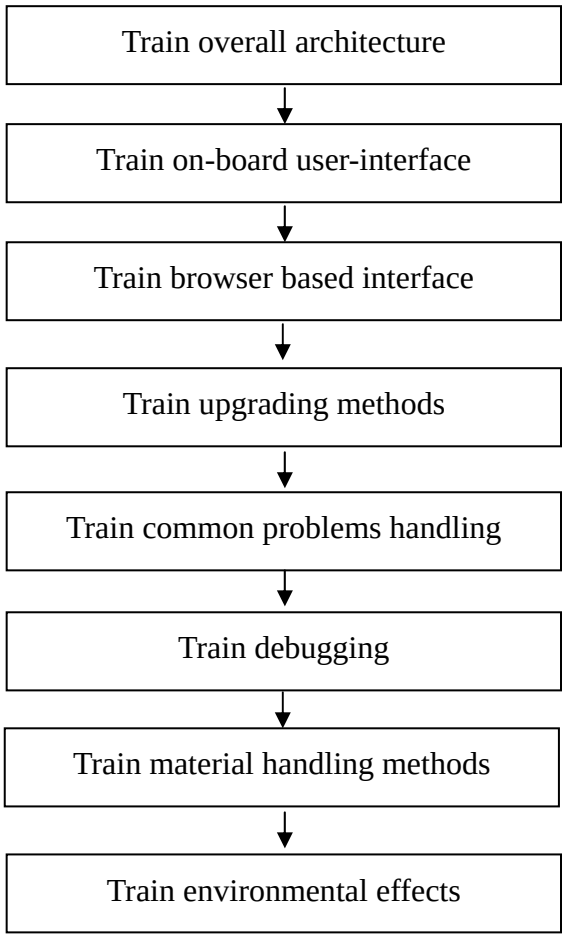
Customization is the step that comes out of optimization. If, after intense optimization, the performance still is not acceptable (or the customer will not accept a lowering of their performance expectation), then some customization may be necessary. The following are just a few possibilities and suggestions for customization. Note that these customizations require the cooperation of the solution provider (i.e. the manufacturer of the products). Very few solution providers are willing to do this without good business justification, though.



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.10 Training

Training is an extremely important step where the operators of the RFID system in the end-customer company must be taught the basics of the operation, plus the necessary tricks in day-to-day trouble shooting and fault isolation – up to a certain extent, of course.

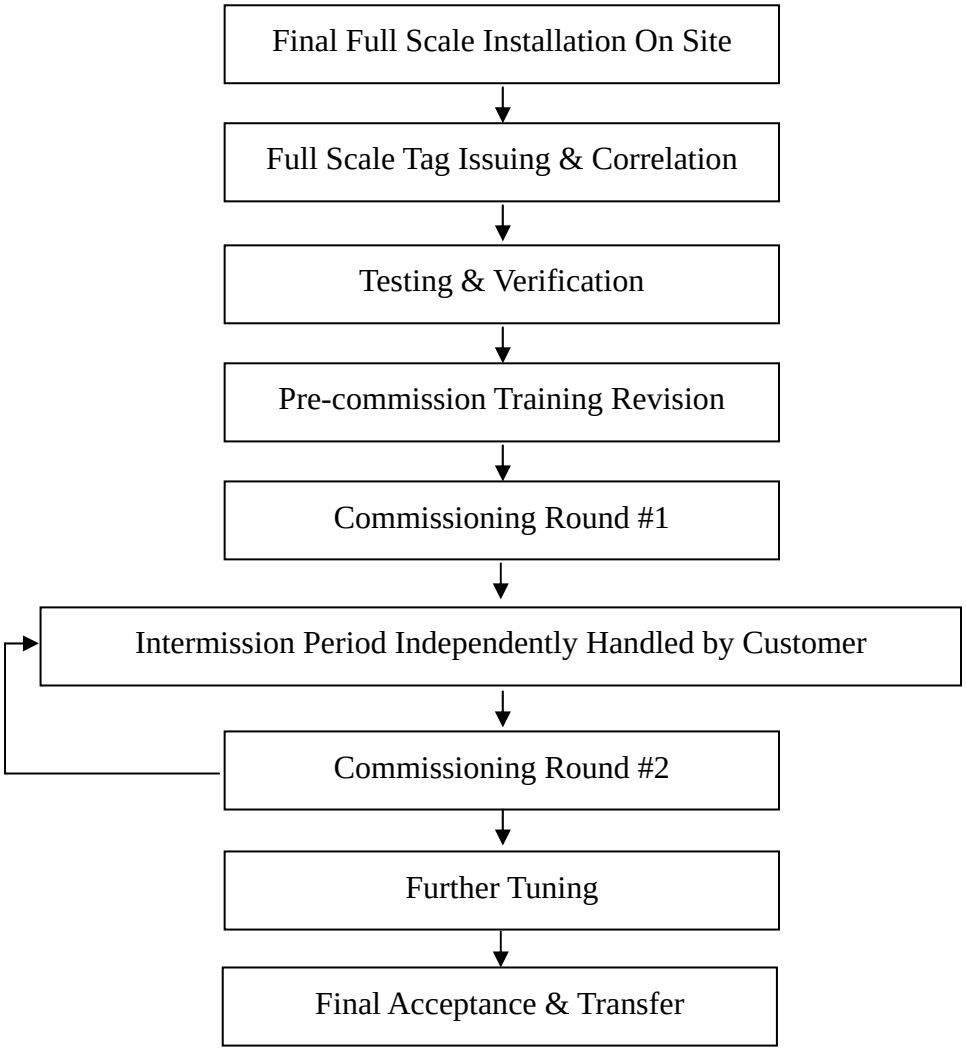


Copyright © Convergence Systems Limited, All Rights Reserved

8.2.11 Test & Commissioning

Test and commissioning is an important step to allow customer to verify the performance achieved, and formally approve the system to enter operational status. The most important part of test and commissioning is of course a mutually agreed test plan and commissioning criteria.

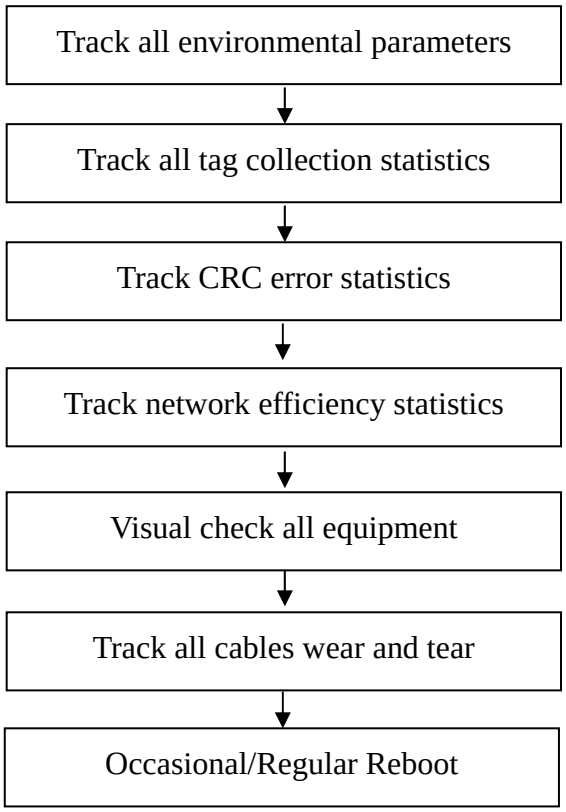
The experienced system integrator can probably propose this test and commissioning plan early in the project. This is particularly valid if the system integrator has done similar jobs before. However, sometimes a T&C document too early in the way will make it very difficult to accommodate for surprisingly low performances due to some uncontrollable environmental or business process related factors. So really it is at the system integrator's own discretion and wisdom when it should best be proposed.



Copyright © Convergence Systems Limited, All Rights Reserved

8.2.12 Maintenance & Statistics

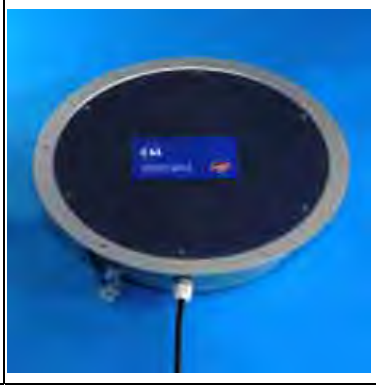
Maintenance of the RFID system is important. It includes preventive maintenance, collection and analysis of statistics of operation, etc.



Copyright © Convergence Systems Limited, All Rights Reserved

8.3 Antennas for Different Business Applications

Various antennas have been designed and optimized for different business processes, such as dock door, ware house, access control, and item level tracking.

Products	**Part Number**	**Photo**	**Business Application**
Antenna (Mono-static area or zonal antenna, long range)	CS-771-LHCP CS-771-RHCP		Logistics Warehouse management Distribution center Transportation management Asset management Baggage management
Antenna (Monostatic access control antenna)	CS-713		Access control Human & animal tracking
Antenna (Brickyard near-field antenna)	CS-777		Retail shop POS Document management Blood bag management Pharmaceutical bottle tracking

Copyright © Convergence Systems Limited, All Rights Reserved

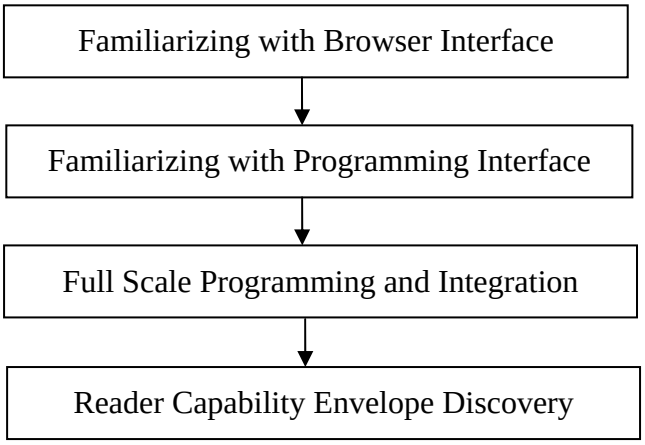
9 RFID Best Practices

9.1 Introduction

System integration of RFID operation is not a simple task. It involves processes such as software configuration, hardware setting, pilot testing, scaling, and more. A good integration is a crucial step to ensure successful ROI for the RFID investment. Improper integration process could affect the system performance as well as functionality. This section describes the best practice for system integrator to familiarize and integrate with an RFID reader, from getting the reader out of the box to deploying the system in production environment.

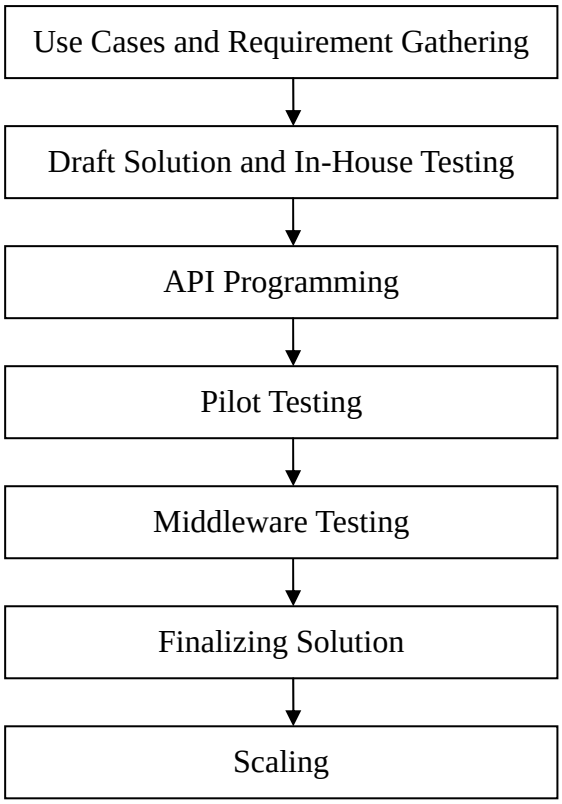
The following flowcharts show the typical familiarization and integration process of CSL CS-461 reader. They represent what a typical system integrator will go through when they adopt the CS-461 technology. By following the path described, the system integrator can quickly deploy CS461 and earn revenue within a very short period.

1. Familiarization Process



Copyright © Convergence Systems Limited, All Rights Reserved

2. Integration Process



Copyright © Convergence Systems Limited, All Rights Reserved

9.2 Integration Process Details

9.2.1 Familiarization Process

9.2.1.1 Familiarizing with Browser Interface

The CSL CS-461 reader comes with a browser interface. Once the reader is connected to the network, it can be accessed from any PC via the Internet Explorer browser. The browser interface allows configuration of reader in a convenient and user-friendly way. The browser interface also allows quick testing of the reader functionalities, including read tag, write tag and kill tag, with and without halt filter, and also for various Gen 2 profiles. Thus, it is an excellent starting point to get familiar with the reader's features in a relatively short period of time. In other words, browser interface allows and enables a good out-of-box experience for the user, even if he/she is a layman in the area of RFID.

With the browser interface, system integrators can try to configure the reader by setting up the operation profile, trigger, action and event. By collecting read tags result under various configurations, system integrators can experience the behavior and characteristics of the reader. For details of the usage of browser interface, please refer to chapter 4.

9.2.1.2 Familiarizing with Programming Interface

The CSL CS-461 reader provides two sets of Application Programming Interfaces (API). One is the High Level API which utilizes HTTP protocol and TCP connection for request/response and notification respectively. The other one is the Low Level API which utilizes TCP connection solely. The High Level API provides features such as event engine, machine automation, etc.

Before starting to program the reader, system integrators are recommended to go through the sample codes which are available for download in CSL web site. The sample codes allow ones to learn how to program the reader in a correct and effective way. The example program flow, API request making and result processing give a general idea of how to interface with the reader. Sample codes of the following demonstrations utilizing the High Level API are described in chapter 錯誤! 找不到參照來源。:

- Access Control

Copyright © Convergence Systems Limited, All Rights Reserved

- Conveyor Belt
- Gambling

9.2.1.3 Full Scale Programming and Integration

Full scale programming allows one to fully control the reader and receive data from the reader with the final goal of integrating the reader with existing business processes, operations and business intelligence software of the customer, such as middleware, ERP system, database, etc. Every system integrator has his own favorite such program, either developed by themselves or based on platforms available from the market, such as Websphere, Weblogic, Biztalk, SensorEdge, RFIDAnywhere, SAP, Oracle, DB2, Sybase, etc.

Once the system integrator passes through the two initial stages of experimenting with the browser interface and the programming interface, he/she needs to start looking at what subset of API calls are needed to enable RFID use in his/her typical customers' business environment. The complete library is rather large (CS461 API library is rich and flexible, and for initial customers may be not all commands are needed), and .

The API includes a number of commands with different parameters. When programming the reader, one should understand clearly the command's usage, effect and the meaning of each parameter since they affect the reader performance directly.

One example is the set operation profile command. The parameter "duplicateEliminationTime" is the time interval in which duplicate tags will be eliminated such that the same tag would not be reported repeatedly during that time interval. It should be set according to situation. Large value of this parameter does not introduce latency since tag is still reported to trusted server once it is read if the action mode is configured to "Instant Alert to Server". However, unnecessarily small value would increase the reader loading and network traffic. In the worst scenario, if this value is set to minimum (i.e. 0.5s) and all four antennas are enabled, note that 0.5s is not enough for the reader to switch over all four antennas for the reading operation, as a result, some tags may be lost.

9.2.1.4 Reader Capability Envelope Discovery

Once full scale programming is started, the user needs to map out the full "flight envelope" of the reader. Important parameters to figure out includes response time, maximum API sending rate, necessary and optimal combinations and sequences of API to achieve different states of

Copyright © Convergence Systems Limited, All Rights Reserved

the machines, fastest possible read and/or best possible yields for various profile combinations, etc. Once the capability envelope is discovered, the system integrator can then work on business projects knowing what the reader is capable of doing and knowing the projects are not requiring the reader to do something it cannot handle.

Copyright © Convergence Systems Limited, All Rights Reserved

9.2.2 Integration Process

9.2.2.1 Use Cases and Requirements Gathering

Before starting the development process, system integrators should fully understand the requirements from customer, such as the throughput requirement, latency requirement, bandwidth requirement. etc, that are specific to the reader. Besides, they could document the use cases which will help in decision making later on in the development process.

9.2.2.2 Draft Solution and In-House Testing

Once the requirements are gathered and use cases are defined, system integrators can develop a draft solution. Draft solution means that it is subjected to final adjustment or tuning after pilot testing. In-house testing allows system integrators to test the feasibility of the solution before deploying to customer’s site.

9.2.2.3 API Programming

The API Programming process here is different from the one in Familiarization Process. In Familiarization Process, system integrators should familiar with the configurations and functioning modes of the reader by using the API. In System Integration Process, they should determine and focus on the configurations and functioning modes to be used in the solution to fulfill user requirements.

9.2.2.4 Pilot Testing

RFID system is greatly affected by environmental factors. For example, background RF noise and metallic object around may affect the read range of antenna dramatically. The same RFID system may function well in the system integrator’s own office but fail in end-customer’s site. Therefore system integrators should conduct on-site pilot testing.

During the on-site pilot testing, system integrators should tackle the site-specific problems that affect the RFID system. For example, if there is metallic object around, position of the antenna

Copyright © Convergence Systems Limited, All Rights Reserved

should be adjusted to overcome the effect of it.

Apart from system settings, RFID tags should be tested as well. System integrators should select suitable tags to cater the business requirement. For example, 3D tag can be read from all directions, but it is less sensitive and large in size. Regular tag has better sensitivity but the read result is highly affected by orientation of the tag.

Some problems may not appear instantly, but only after the system running continuously for hours or days. To identify such problems, long time burn-in testing is required. If any problem related to the reader is found, the system integrator could send a bug report with reader settings, antenna setup and site-specific factors to CSL for troubleshooting.

9.2.2.5 Middleware Testing

Usually, a middleware is used between the reader and enterprise application. It plays an important role in the integration of reader and therefore it must be fully tested as well. CSL provides service for such testing. System integrators can give the executable of the middleware to CSL for long term testing to ensure that the middleware is free of problem after running continuously. Moreover, all API calls requested by the middleware are logged in the reader which allows CSL to analyst the cause of problem if there is any.

9.2.2.6 Finalizing Solution

The finalized solution should tackle all of the problems found in pilot test and fine tune the solution if necessary. Then it is ready for production running.

9.2.2.7 Scaling

Scaling process should be done after the system is tested to be stable. Moreover, scaling gradually at the end-customer site (if end-customer permits, of course) can reduce the chance of system failure due to overloading. For a large scale RFID system that involves hundred of readers, the system integrators should pay attention to the followings:

1. Readers that are close to each other are recommended to use Profile 2 or 3 of Modulation Profile. It allows the readers to work in dense reader mode such that jamming could be

Copyright © Convergence Systems Limited, All Rights Reserved

avoided. Remember to select different session numbers for readers to avoid tag replying wrongly to other reader.

2. If dense reader mode is not required, Profile 0 should be used as it allows the fastest tag read.

3. Adjust the power of reader to take a balance between read range and cross read effect.

4. Employ inspection process for identifying malfunction reader. For example, reading testing tags from all readers and then collecting the read data from edge server. Analysis of the data helps assessing the reader health.

5. Remote reboot of reader and remote control of power grid should be supported since the readers may distribute in vast area.

6. During network failure, reader is not able to send tags read to trusted server. If Network Failure Data Backlog is enabled, those tags are buffered in the reader. Backlog tags are sent to trusted server after the TCP connection is re-established. Therefore, system integrators should also provide application level failover for this feature.

Copyright © Convergence Systems Limited, All Rights Reserved

10 RFID Use Cases

10.1 Warehouse Real Time Inventory Tracking

Use Case

In warehouse with huge amount of inventory and fast turnover, acquiring real-time inventory data becomes a big challenge.

Current Approach

Stocktaking is done manually or using barcode system. The process is costly and slow. Inventory data are inaccurate due to human errors. Real-time visibility of inventory data is not available.

Suggested Approach

By equipping RFID read points in warehouse, inventory is being monitored continuously. Inventory data are updated in real time, giving warehouse manager real-time visibility to inventory level and status. This is particularly important for time sensitive merchandise. It also helps identifying potential theft for high-value merchandise, greatly reduce the labor cost and human error.

Recommendation

The CSL CS461 reader is powered by Impinj technology with extremely high inventory rate, which is important for providing high accuracy on inventory data. It is also able to manage large streams of tag data efficiently so that it can cope with tremendous amount of tags in warehouse environment. Moreover, the highly configurable buffering and tag filtering modes allow the elimination of redundant tag data so as to reduce network traffic and server loading.

Copyright © Convergence Systems Limited, All Rights Reserved

10.2 High Traffic Human Access Control

Use Case

Many companies world-wide already use RFID technology for employee access control systems. The access control system can fulfill purposes such as limiting access to a restricted area and capturing entry and exit time information for wages calculation.

Current Approach

HF technology is adopted in many access control systems. The read range of HF is short such that presenting of access card in front of the read point is required. This process can cause congestion under high traffic of access especially right before and after the office hour.

Suggested Approach

For access control system with high traffic of access, UHF has advantage over HF because the employees do not have to present the access card to the read point one by one, instead they can just walk by the read point and the access card can be read.

Recommendation

The CSL CS461 reader is powered by Impinj technology with extremely high inventory rate. This ensures the information captured is accurate and reliable.

Copyright © Convergence Systems Limited, All Rights Reserved

10.3 Reusable Pallet Tracking

Use Case

Reusable pallets travel through the supply chain many times in its life time. If the pallets can be tracked, they can be maintained in a better and manageable way.

Current Approach

Barcode system is used. Time of scanning the barcodes in large stack of pallets is long since only one barcode can be scanned at a time and line-of-sight is required.

Suggested Approach

Tagging of reusable pallets allows tracking them throughout the entire operation and maintenance cycle. This usage can even be extended to track movement of goods on the pallet throughout the distribution cycle. This offers the pallet providers as well as the goods distributors a complete visibility of their pallets and goods at every distribution point.

Recommendation

Powered by Impinj technology, the CSL CS461 reader has extremely high inventory rate which can read the tags in large stack of pallets accurately. This ensures accuracy of data about the pallets together with the goods.

Copyright © Convergence Systems Limited, All Rights Reserved

10.4 Work-In-Progress Monitoring

Use Case

The manufacturing process in factory can be long and complicated. Once the raw materials are sent into the manufacturing plant, they remain invisible until emerging as a finished product. Better visibility of work-in-progress is required for production decision-making.

Current Approach

Tracking of manufacturing process is not automated. Status of parts and work-in-progress are out-dated, distributed and manually collected.

Suggested Approach

The introduction of RFID technology to the manufacturing process in factory can improve the visibility of the work-in-progress. Parts and subassemblies within the manufacturing plant are tracked precisely such that more accurate part level and work-in-progress records are available. Moreover, automatic monitoring of work-in-progress status on semi-finished assemblies throughout the production cycle can reduce downtime and ensure on-time delivery. Combining RFID reader with output device can also help in decision making. For example, alarm is triggered when semi-finished items or batches are routed to the wrong manufacturing cell.

Recommendation

As powered by the advance and intelligent technology from Impinj, the CSL CS-461 reader has the unique feature of Dense Reader mode. It allows multiple readers to be used in very close separation or area without jamming each other. This is particularly suitable for deploying in manufacturing plant with RFID readers equipped in conveyors, gates...etc.

Copyright © Convergence Systems Limited, All Rights Reserved

10.5 Human Access Control by Autonomous Tag Groups in Reader

Use Case

RFID can be applied to access control system for preventing unauthorized access to a restricted area.

Current Approach

Most RFID access control systems rely on backend server and database for security controls. If the server is down or network service is not available, the whole access control system fails.

Suggested Approach

Autonomous access control system with embedded event engine for security controls.

Recommendation

With the embedded event engine in CSL CS-461 reader, autonomous tag group filtering for access control is achieved. Tag groups can be pre-programmed into the reader easily. Once set, the access control system can run autonomously even when network or server is down.

Copyright © Convergence Systems Limited, All Rights Reserved

10.6 Pallet/Carton Tagging Verification

Use Case

RFID implementation is growing in different industries. The automated handling solutions driven by RFID are very much relying on the tags. Therefore missing or failed tags can have a major impact on operating efficiencies.

Current Approach

Verification of tag on pallet or carton is not automated. Missing or failed tags are difficult to be identified and replaced.

Suggested Approach

By combining input and output devices, RFID technology can be applied to verification of tag existence in pallet or carton. In this case, infrared sensor is used to trigger the start inventory in reader. When the pallet or carton passes the read point, status of infrared sensor changes which triggers the reader to start inventory. If no tag is read after the trigger, output device such as alarm is turned on. This application helps to identify missing or failed RFID tags such that tag replacement action can be taken.

Recommendation

The CSL CS-461 reader contains I/O port which allows maximum four inputs and eight outputs. The embedded event engine also allows I/O trigger and event to be programmed into the reader easily. Once these are set, the tag verification process is done autonomously.

Copyright © Convergence Systems Limited, All Rights Reserved

10.7 Blood Bag Tracking

Use Case

RFID technology can be widely adopted in medical field. One of the applications is blood bag tracking and blood type verification for transfusion.

Current Approach

Blood bags in blood banks are managed manually or using barcode system. Real-time visibility of inventory data is not available.

Suggested Approach

Before transfusion, the nurse can check the blood type contained in blood bag against patient's blood type by cross checking the RFID tags on both the blood bag and patient's wristband. In this way, chance that a patient being transfused the wrong blood type due to human error is greatly reduced. Moreover, by equipping RFID read points in the blood bank, real time inventory data can be grasped, giving medical staff real-time visibility to inventory level and status.

Recommendation

One of the major challenges in implementing RFID system for blood bag tracking is the liquid content inside blood bag, as fluid can degrade the radio frequency signal. However, with CSL CS-777 near-field antenna, this problem can be overcome since it can achieve outstanding performance when reading and writing tags on container with different contents even liquid.

Copyright © Convergence Systems Limited, All Rights Reserved

10.8 Pharmaceutical Bottles Tracking and Anti-counterfeit

Use Case

Counterfeit problem is a major concern in pharmaceutical industry for years. It does not only threaten the public safety, but also poses economic damage to pharmaceutical manufacturers. An effective measure is required to combat the growing counterfeit problem.

Current Approach

Effective measure for anti-counterfeit is difficult to implement without the tracking of pharmaceutical bottles throughout supply chain.

Suggested Approach

With the introduction of RFID technology, item level supply chain visibility for pharmaceutical product can be facilitated. It provides the track and trace of drugs as distributed throughout the entire supply chain, which in turns protects the public health. Pharmaceutical manufacturers can also benefit from reduced liability, brand protection and additional revenue that was previously diverted to makers of counterfeit drugs.

Recommendation

The CSL CS-777 near-field antenna achieves outstanding performance when reading and writing tags on variety of packaging options including pharmaceutical bottles and metals found in blister packs. In the magnetic near-field, UHF Gen 2 tags works well with container of different contents such as powders, pills and even liquids. This ensures the technology can be applied to all kinds of materials characteristically found in pharmaceutical products and packaging including vials of vaccines and bottles of liquid medication.

Copyright © Convergence Systems Limited, All Rights Reserved

10.9 Vehicle Tracking in Maintenance Depot

Use Case

In maintenance depot, vehicles arrive for maintenance and checking. If the activities of vehicles inside the maintenance depot can be tracked, better arrangement of vehicles maintenance can be achieved.

Current Approach

Vehicle maintenance is tracked manually. Human errors may occur such as omitting particular maintenance checking on a vehicle.

Suggested Approach

RFID technology can be applied to track vehicles' activities inside the depot. Once a vehicle is tagged, it's movement can be recorded anywhere in the RFID enabled depot. The process is completely automatic in the sense that the vehicle does not have to stop for being recorded. Moreover, no staff is involved in the process and thus human errors can be eliminated. The vehicles' movement record gives accurate maintenance checking and repairing history which is important for vehicle management such as identifying obsolete parts.

Recommendation

One of the challenges in tracking vehicles in maintenance depot is that high tag resolution is required. Cross reading of tags by different entry points would affect the accuracy of identifying the vehicles in the lane. This problem can be overcome by shielding the capture points such that each capture point would only read tags that are corresponding to it. Furthermore, the CSL CS-461 reader allows filtering of tags by both RF Signal Strength Indicator (RSSI) and read count to prevent cross reading of tags by read points in multiple lanes.

Copyright © Convergence Systems Limited, All Rights Reserved

10.10 Vehicle Information System

Use Case

In many countries, the possibility of using an RFID tag as a license plate is very welcome because that enables a host of analysis, tracking and law enforcement operations.

Current Approach

Vehicle license has traditionally been tracked visually or optically.

Suggested Approach

RFID technology can be applied to the label on the windshield, or to a stand on the dashboard, or to the inside of the Taxi light box on top of a taxi, or even directly onto the front and back license plate. The reader antenna can be mounted either on a low overhang/footbridge or simply on a pole on the side of the road.

Recommendation

The CS461 has been tested with tagged vehicles traveling at 90 Km/hr and still achieves 100% read yield. Test beyond 90 Km/hr can be done by customers with such facilities (including a stretch of road for testing!!)

Copyright © Convergence Systems Limited, All Rights Reserved

10.11 Document Tracking

Use Case

In some organizations, costs associated with tracking documents are high. An automatic document management system is especially beneficial in those environments where the documents are of high value to the organization, and the loss of a document would have significant negative impact. Examples include hospitals, lawyer's offices and government departments.

Current Approach

Documents are tracked and managed manually. Human error may lead to lost of documents. Moreover, time spent in searching for document is long, especially when documents are not systematically well organized.

Suggested Approach

RFID technology has made a dramatic improvement in tracking and managing documents. By tagging the documents and equipping read points for checking in and out, status and location of documents can be traced easily. Other usages such as inventory checking and locating lost documents can also be achieved.

Recommendation

Different antennas are designed to be used with the CSL CS-461 reader to fulfill different requirements of document tracking. For example, for checking in and out of documents, short and constrained read range is required such that documents that are put near the read points would not be included accidentally. In this case, the CS-777 near-field antenna that is optimized to read near-field tags should be used. Oppositely, for inventory taking, longer read range is required such that all documents within the area are read rapidly. In this case, the CS-771 Mono-static Antenna with long read range should be used.

Copyright © Convergence Systems Limited, All Rights Reserved

11 Troubleshooting Guide

11.1 Common Problems and Possible Causes

The following are some of the most common problems and causes:

Problem	Possible Cause	Troubleshooting Procedure
Hardware		
Cannot Read Tag From Antenna	- Appropriate Antenna is not selected in Operation Profile - Appropriate Antenna is not selected in Trigger - Selected Frequency Channel is jammed - Population Estimation is not set properly - Estimated Tag Time In Field is not set properly - Selected Country is not the reader designed for. - Antenna mismatch (red light at antenna port is usually off)	Section 11.2.1.1
Short Read range	- Selected Channel is jammed - Transmit power is not enough	Section 11.2.1.2
No Read From Dense Readers	- Inappropriate Modulation Profile is used - Same session number is used	Section 11.2.1.3
I/O Device Not Work	- Device connection problem	Section 11.2.1.4
Web Browser Interface		
Cannot Access Browser Interface	- Reader not booted up yet - Incorrect IP address or port number is used - Configuration file is corrupted	Section 11.2.2.1
Health Check Fail	- Reader does not start up properly	Section 11.2.2.2
Write Tag Fail	- Antenna is not selected in configuration - Transmit power is not enough	Section 11.2.2.3
Low Level API Demo Program		

Copyright © Convergence Systems Limited, All Rights Reserved

Cannot Connect To Reader	- Reader is not set to Low Level Mach1 API Mode - Incompatible version of demo program is used with the reader's firmware - Another reader with the same IP is in the network	Section 11.2.3.1
Cannot Read Tag	- Population Estimation is not set properly - Estimated Tag Time In Field is not set properly - Selected Country is not the reader designed for.	Section 11.2.3.2
Programming Interface		
Command *getCaptureTagsRaw* Cannot Get Newly Captured Tag	- Command *startInventory* is not called before	Section 11.2.4.1

Copyright © Convergence Systems Limited, All Rights Reserved

11.2 Troubleshooting Procedure

11.2.1 Hardware

11.2.1.1 Cannot Read Tag From Antenna

Cause: Appropriate Antenna is not selected in Operation Profile

Check if the LED on the reader corresponds to the antenna port is on. If not, make sure the antenna is selected in Operation Profile, or else, it will not be enabled.

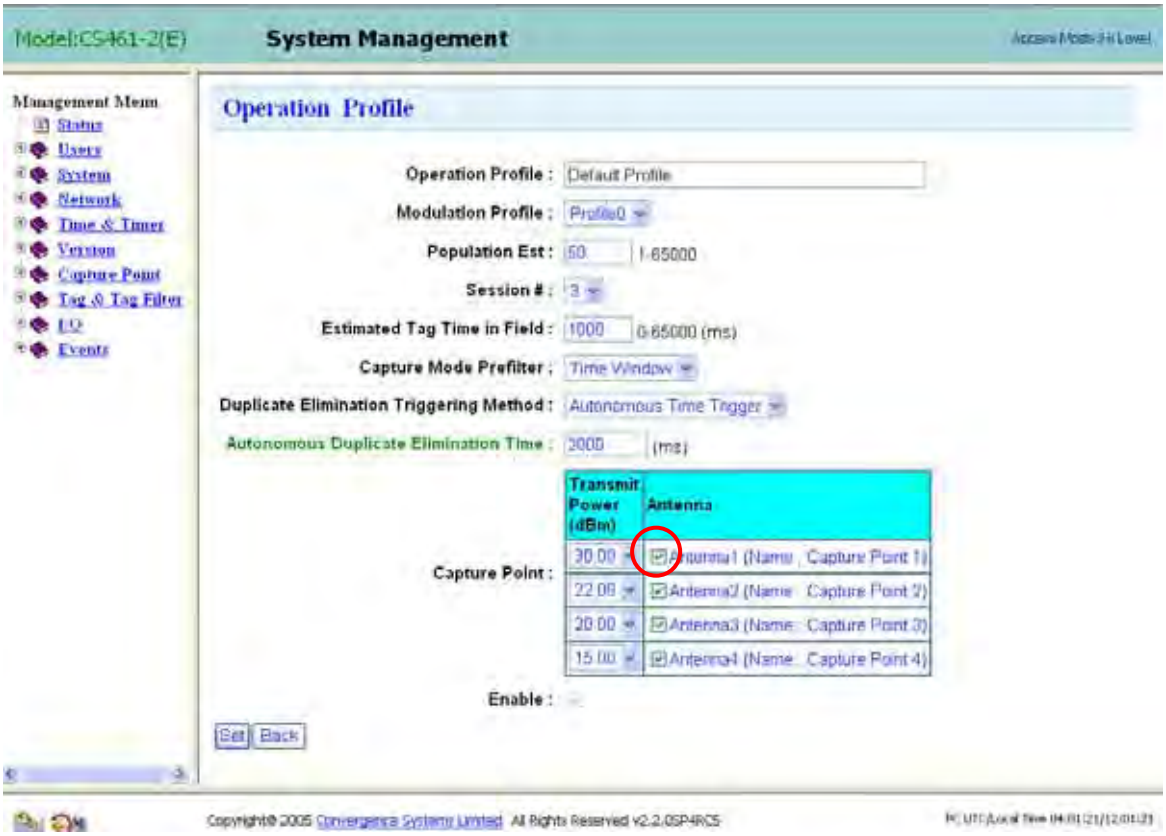


Figure 11-1

Cause: Appropriate Antenna is not selected in Trigger

Make sure the antenna is selected in the appropriate Trigger. Otherwise no event will be triggered even the antenna reads the tag:

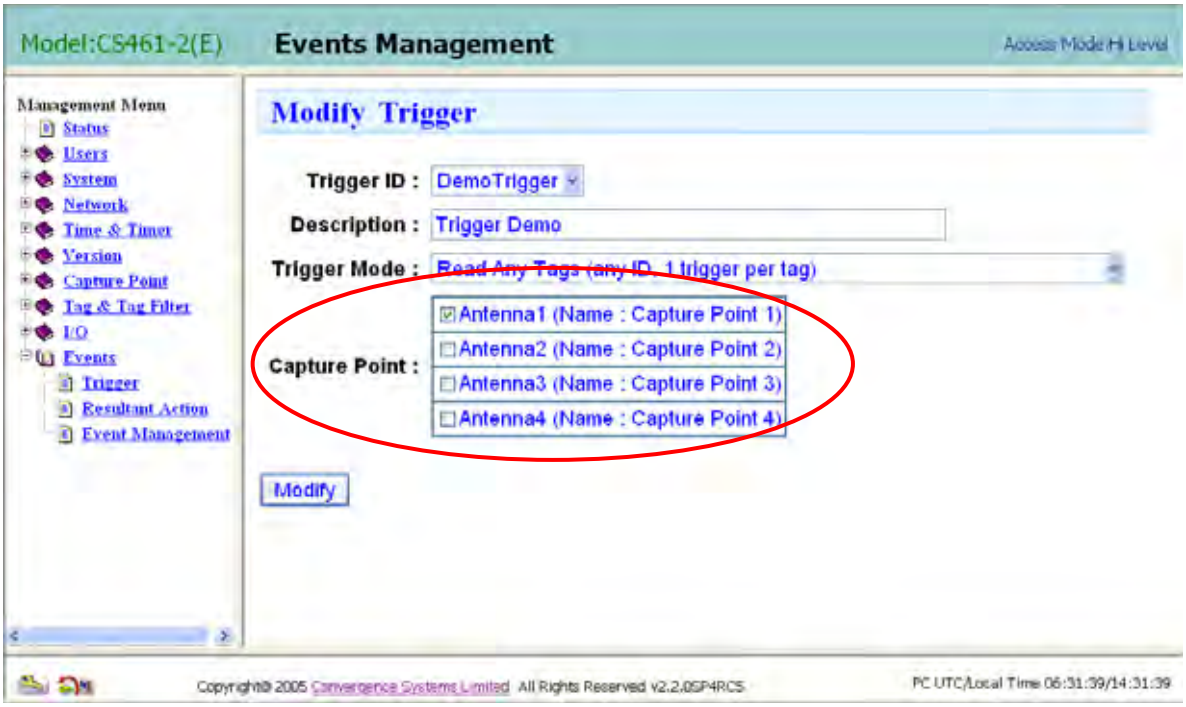


Figure 11-2

Cause: Selected Frequency Channel is jammed

For operation in Japan, if fixed channel is selected, “listen before talk” is employed. However, if the selected channel is jammed, the reader will not be able to use that channel to read tag. In this case, try to change the other channel in Operation Profile.

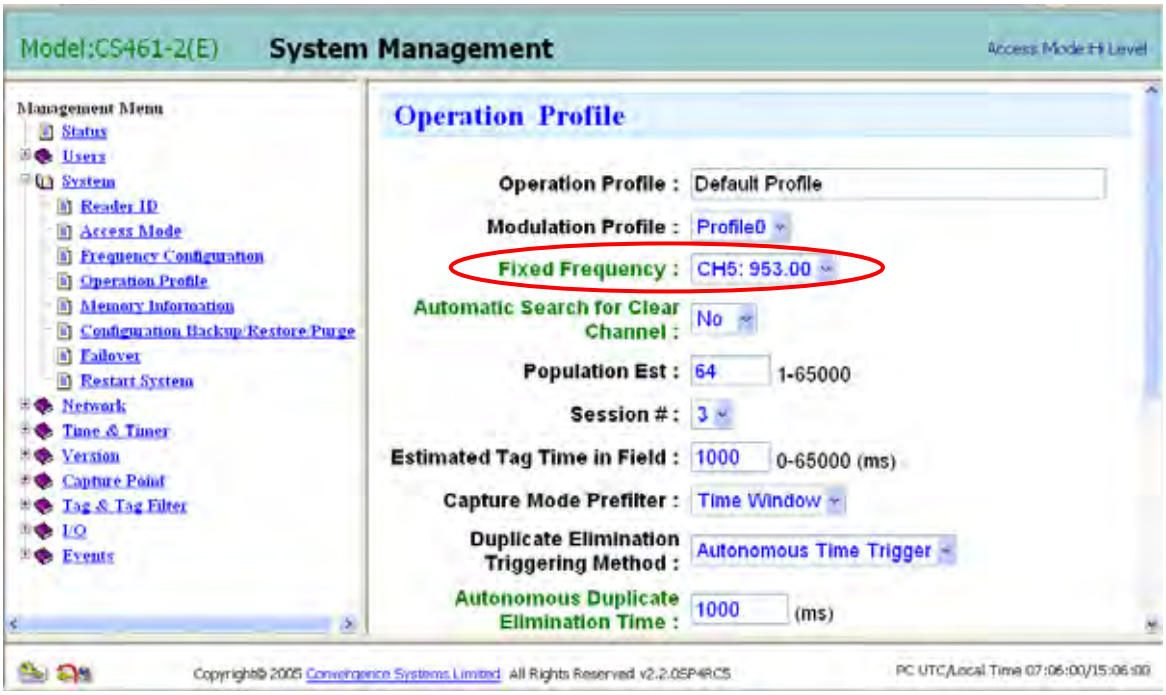


Figure 11-3

Cause: Population Estimation is not set properly

Make sure the Population Estimation is set properly in Operation Profile. It should be set to about 20% more than the maximum number of tags expected to be read at a time. Too large or too small value may degrade the reader performance.

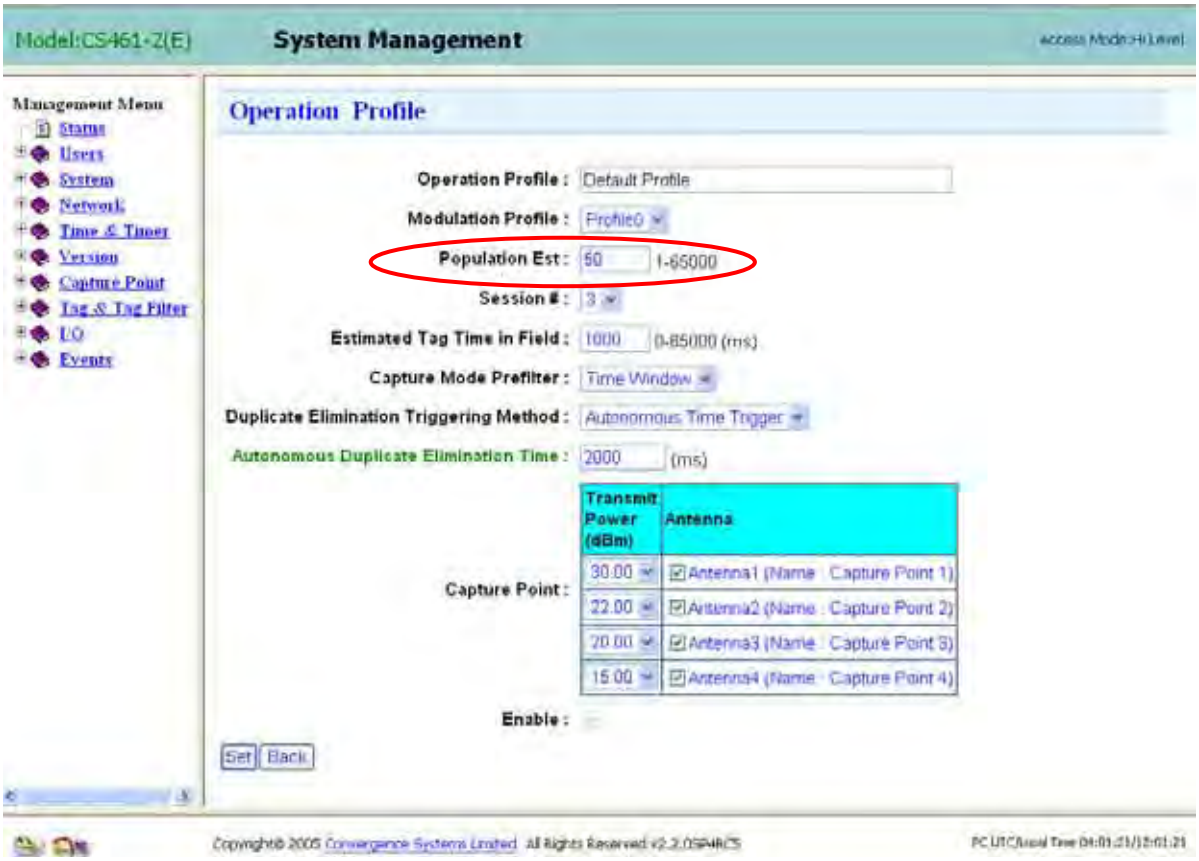


Figure 11-4

Cause: Estimated Tag Time In Field is not set properly

Make sure the Estimated Tag Time In Field is set properly in Operation Profile. It should be set to an estimated time that tags are expected to be in field. If it is set too large and tag appears only in field shortly, the tag may not be read.

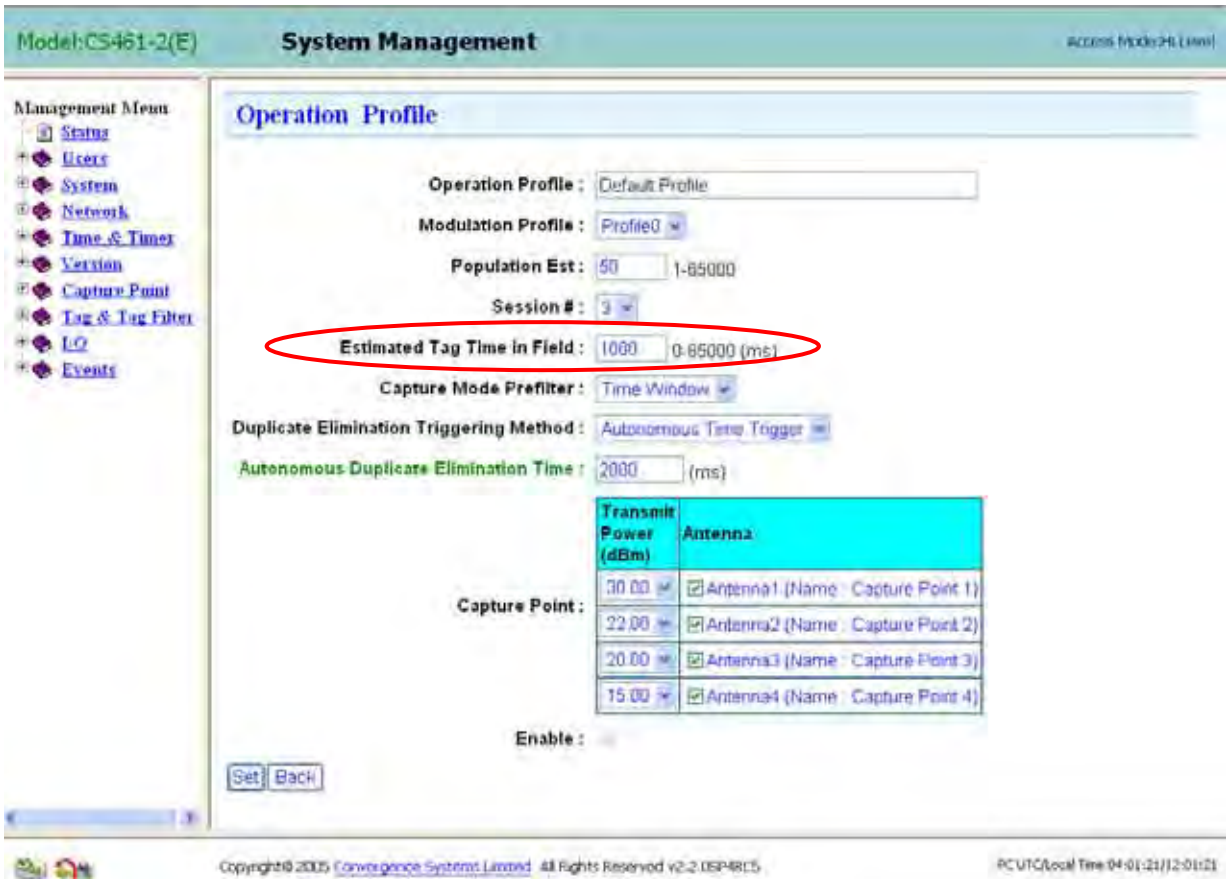


Figure 11-5

Cause: Selected Country is not the reader designed for

Make sure the Country selected in Frequency Configuration is the reader designed for. If the reader does not support operation for the selected country, it will not operate properly.

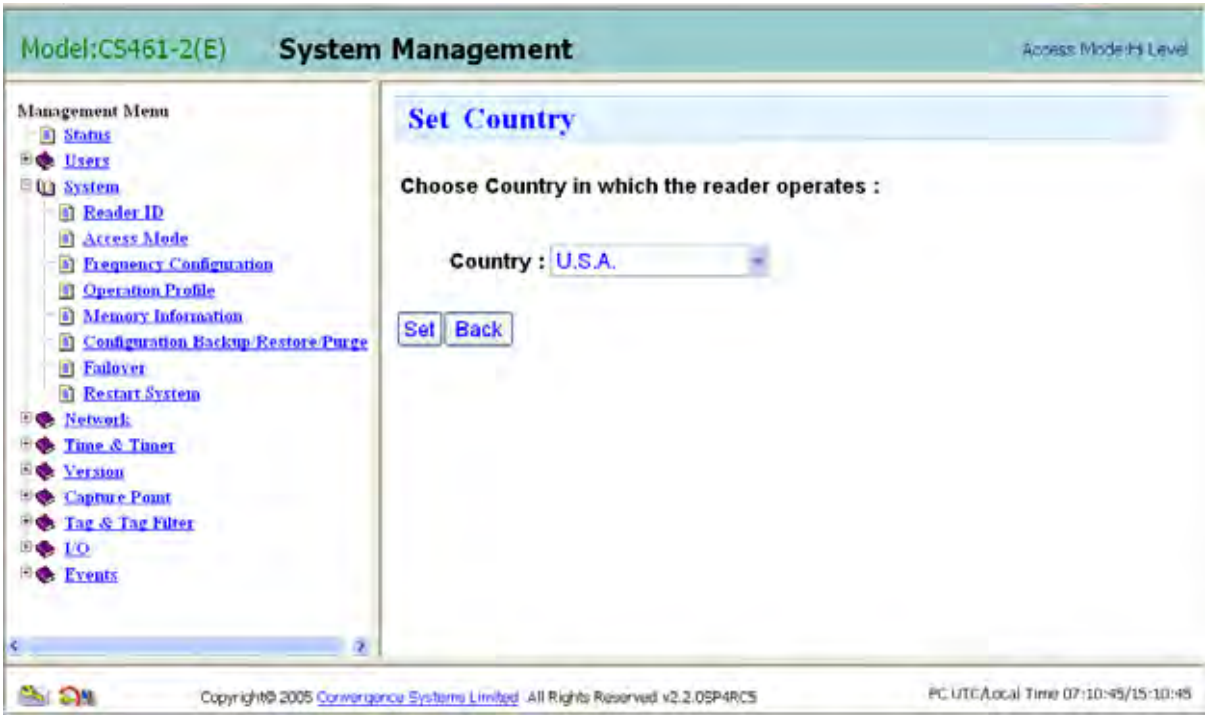


Figure 11-6

11.2.1.2 Short Read Range

Cause: Selected Channel is jammed

Avoid interference of noise.

Cause: Transmit power is not enough

Make sure the Transmit Power is large enough. Increase the power in Operation Profile if necessary:

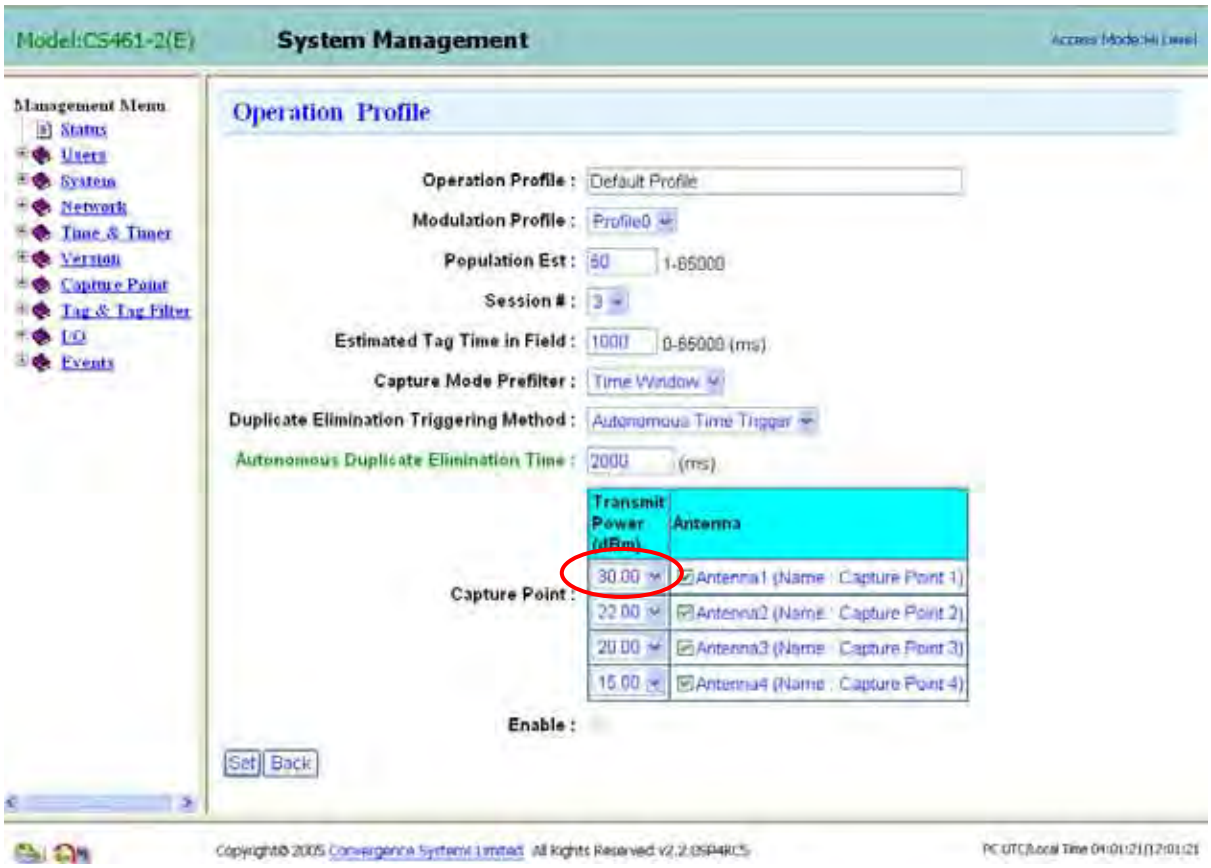


Figure 11-7

11.2.1.3 No Read From Dense Readers

Cause: Inappropriate Modulation Profile is used

When more than one reader are operating in close separation, interference may occur, to overcome, Modulation Profile 2 or 3 should be used. These two profiles are specifically used in dense reader environment.

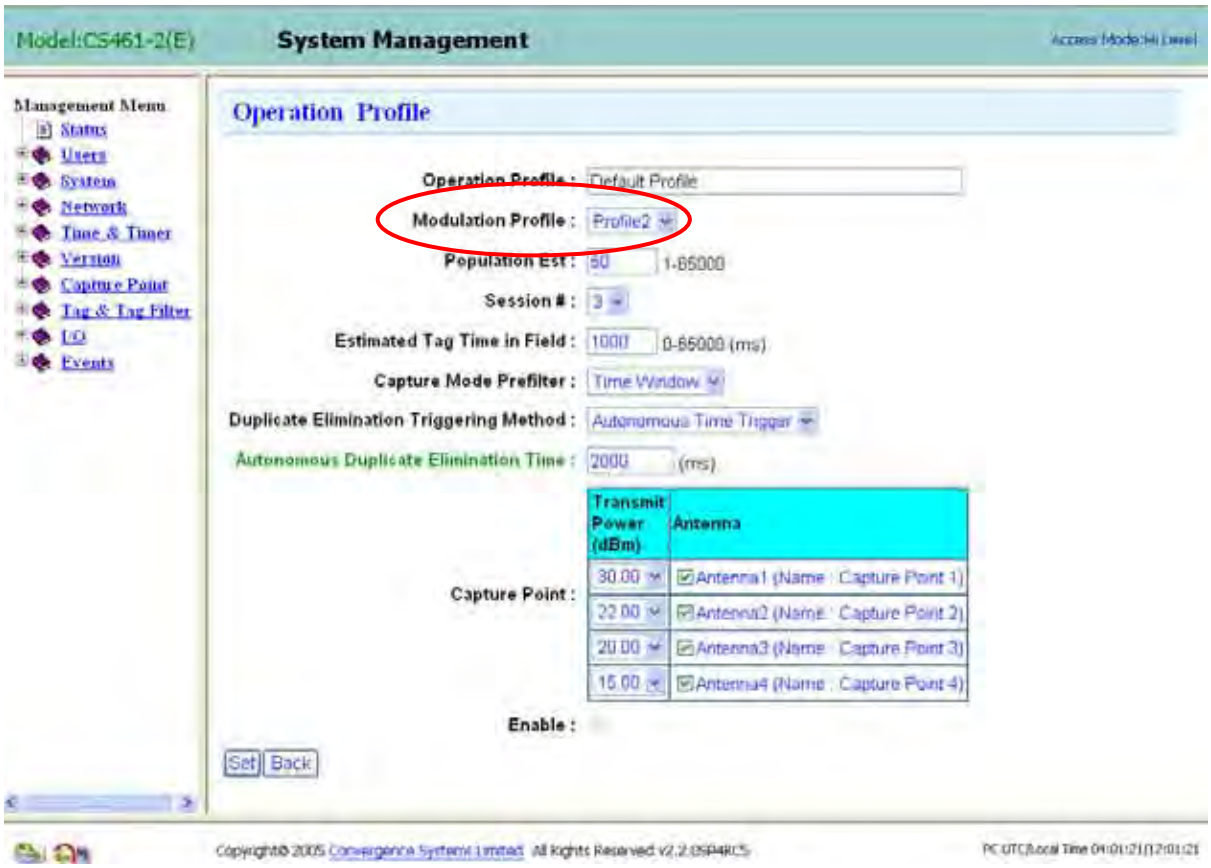


Figure 11-8

Cause: Same session number is used

Select different session numbers for different readers in Operation Profile.

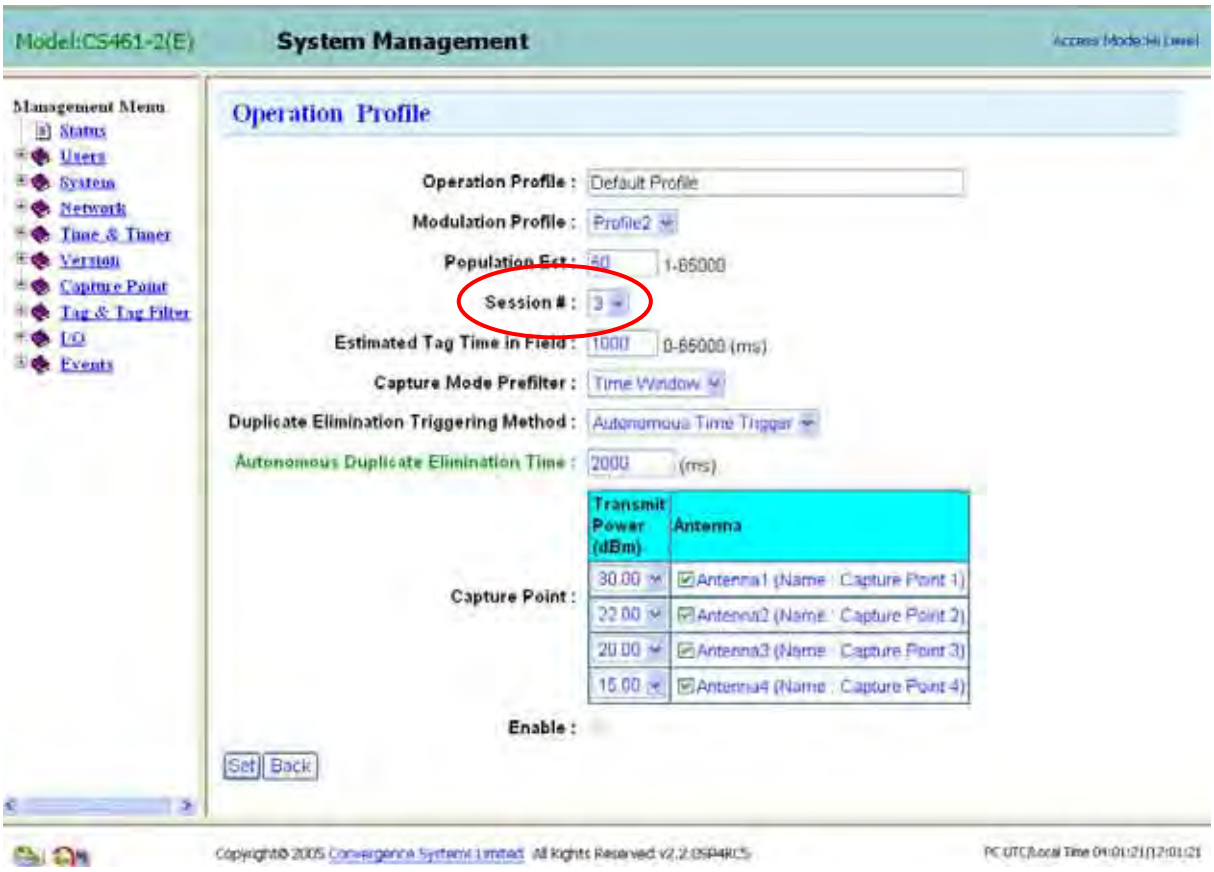


Figure 11-9

11.2.1.4 I/O Device Not Work

Cause: Device connection problem

Test the functionality of I/O port in the “I/O Port Testing” page. The login name and password for this page are as follows:

Login: test engineer
Password: cnernd

Figure 11-10

After login, the 4 input sensors and 8 output controls can be tested.

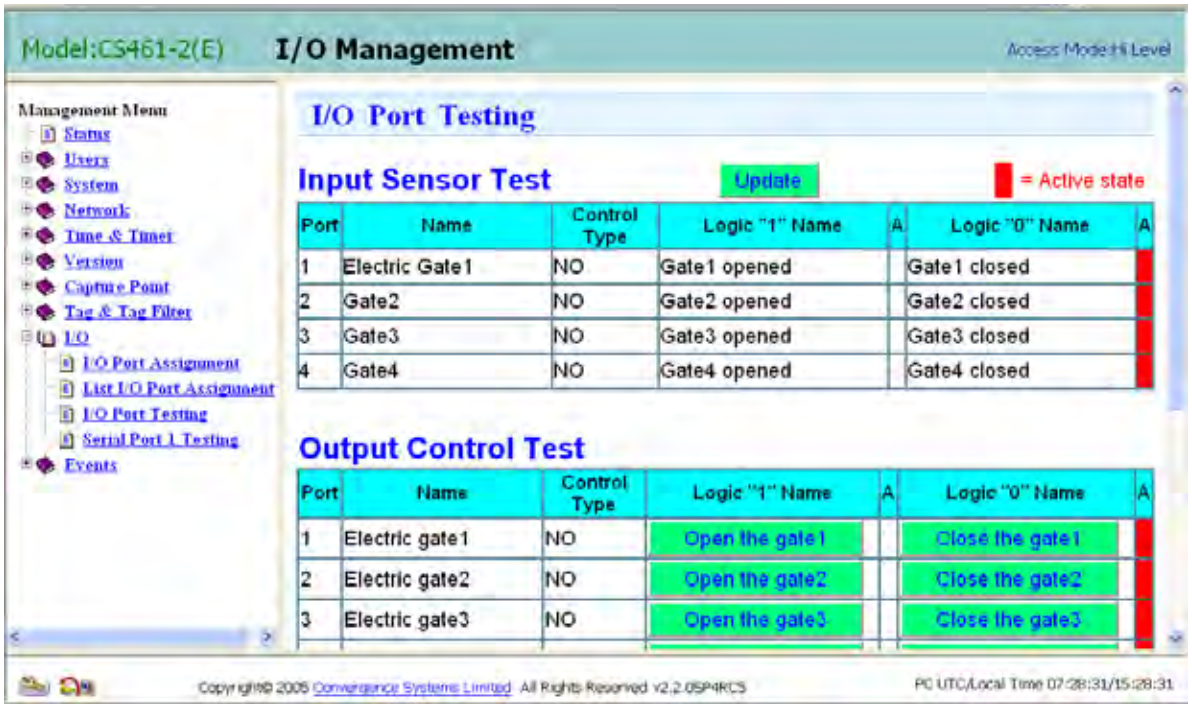


Figure 11-11

11.2.2 Web Browser Interface

11.2.2.1 Cannot Access Browser Interface

Cause: Incorrect IP address or port number is used

Make sure the IP address and port number in the URI is correct. The IP address and port number of the reader can be checked with a console.

i) Connect a PC with the reader using the serial port. Then, open a console and connect to the reader using the following settings:

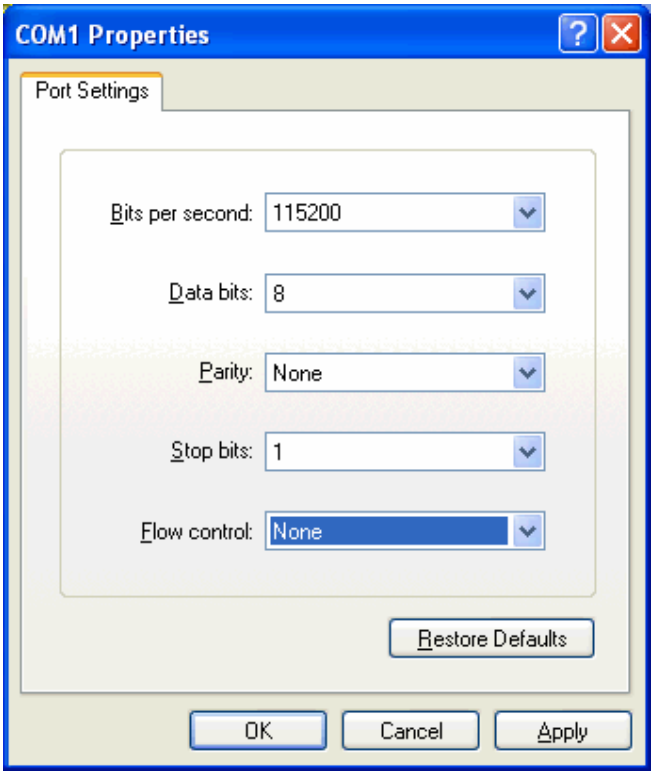


Figure 11-12

ii) Press enter after connected, the login prompt will be shown:

Copyright © Convergence Systems Limited, All Rights Reserved

Figure 11-13

The login name and password are as follows:

Login: root

Password: csl1

The following screen is shown after login:

Figure 11-14

iii) The IP address can be viewed by the `ifconfig` command as follows:

Copyright © Convergence Systems Limited, All Rights Reserved

```
CSL login: root
Password:
root@CSL:~# ifconfig
ixp0      Link encap:Ethernet  HWaddr 00:05:7B:22:01:38
          inet addr:10.8.123.228  Bcast:10.8.123.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25416 errors:0 dropped:18 overruns:0 frame:0
          TX packets:400 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:256
          RX bytes:3210977 (3.0 Mb)  TX bytes:111537 (108.9 Kb)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:112548 errors:0 dropped:0 overruns:0 frame:0
          TX packets:112548 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7032954 (6.7 Mb)  TX bytes:7032954 (6.7 Mb)

root@CSL:~# _
```

Figure 11-15

iv) To view the port number, type the following command:

```
cat /tmp/usb_websvr/websvr/nevow-tmp/cgi/NetworkConfig.xml
```

The port number and other network settings are shown:

```
root@CSL:~# cat /tmp/usb_websvr/websvr/nevow-tmp/cgi/NetworkConfig.xml
<CSL>
<Command>getNetworkConfig</Command>
<NetworkConfig ap_name="DemoAP" backup_ap_name="DemoAP_Backup" dhcpmode="0" gate
way="10.8.123.1" ip="10.8.123.228" mask="255.255.255.0" networktype="Wired" port
="80" ecutype="None" value="" />
</CSL>root@CSL:~#
```

Figure 11-16

Cause: Configuration file is corrupted

If the IP address and port number is correct but still cannot access the browser interface, the configuration file of the reader maybe corrupted. Restore factory default setting by the following commands in the console:

```
> cp –f /tmp/usb_websvr/websvr/nevow-tmp/default/* /tmp/usb_websvr/websvr/nevow-tmp/cgi
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
> cp –f /tmp/usb_websvr/websvr/nevow-tmp/default/run.pyc /tmp/usb_websvr/websvr/nevow-tmp
> cp –f /tmp/usb_websvr/websvr/nevow-tmp/default/rfid_opera_config.txt /tmp/usb_main
```

Then, restart the reader. Note that the IP address and port number of the reader will be changed to 192.168.25.248 and 80 respectively.

11.2.2.2 Health Check Failed

The “Status” page of web browser interface provides health check of the reader. Normally, the health check result for Modem Controller, Middleware and Edge Server should all be “PASS” as shown:

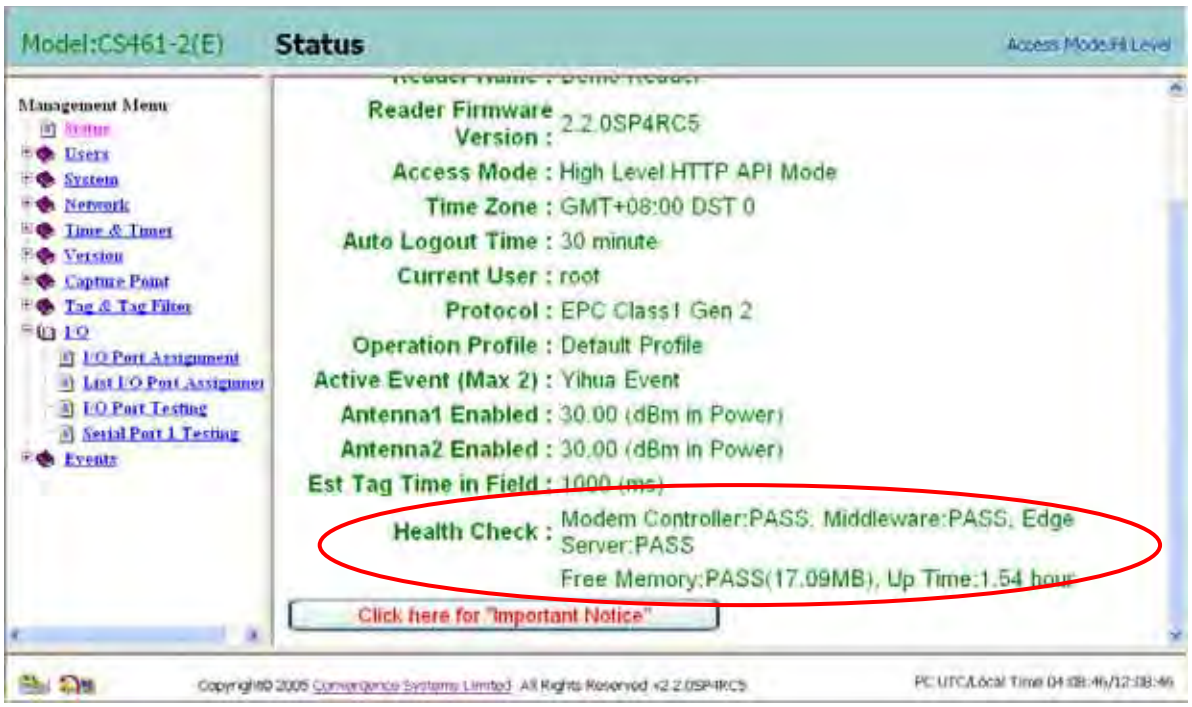


Figure 11-17

If any of the results is failed, try to restart the reader. If the problem persists, restore factory default setting by following the instruction in chapter 11.2.2.1 step 0.

11.2.2.3 Write Tag Fail

Make sure the reader configuration of write tag is set correctly. The configuration for write tag and read tag are different. Check that the antenna is selected and the Transmit Power is set large enough (at least 22 dBm) in the “Write Tag Testing” page.

Figure 11-18

11.2.3 Low Level API Demo Program

11.2.3.1 Cannot Connect to Reader

If the reader is not connected successfully, the following screen is shown when the “START Inventory Run” button is pressed:

Figure 11-19

Cause: Reader is not set to Low Level Mach1 API Mode

Make sure the access mode of the reader is set to Low Level Mach1 API Mode. The access mode can be set in web browser interface:

Copyright © Convergence Systems Limited, All Rights Reserved

Figure 11-20

Cause: Incompatible version of demo program is used with the reader's firmware

Make sure the demo program version is compatible with the reader's firmware. For reader firmware 2.2.0 or later, demo program 4.0.5 or later should be used.

11.2.3.2 Cannot Read Tags

Cause: Population Estimation is not set properly

Make sure the Population Estimation is set properly. It should be set to about 20% more than the maximum number of tags expected to be read at a time. Too large or too small value may degrade the reader performance.

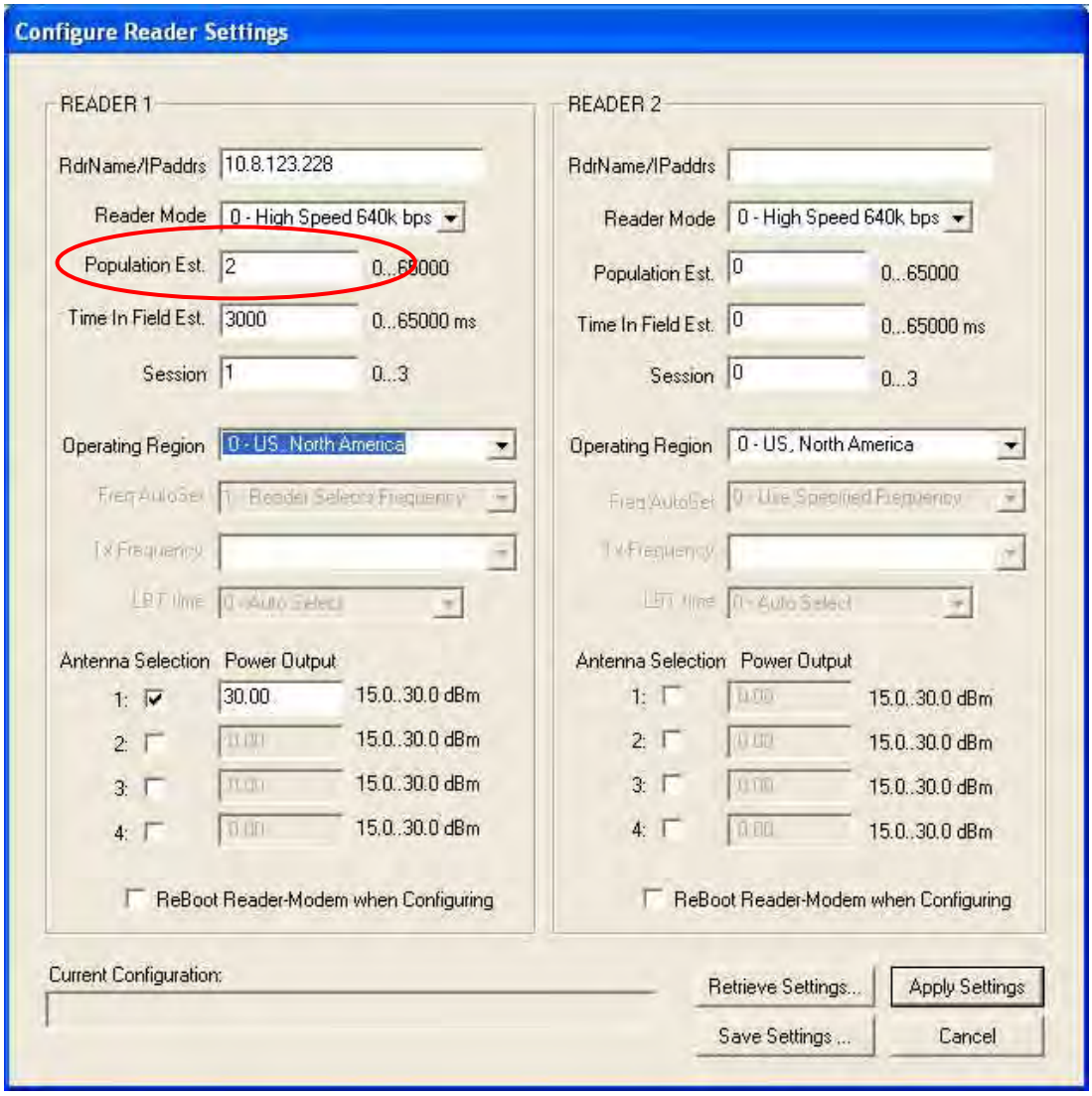


Figure 11-21

Cause: Estimated Tag Time In Field is not set properly

Make sure the Time In Field Estimation is set properly. It should be set to an estimated time that tags are expected to be in field. If it is set too large and tag appears only in field shortly, the tag may not be read.

Copyright © Convergence Systems Limited, All Rights Reserved

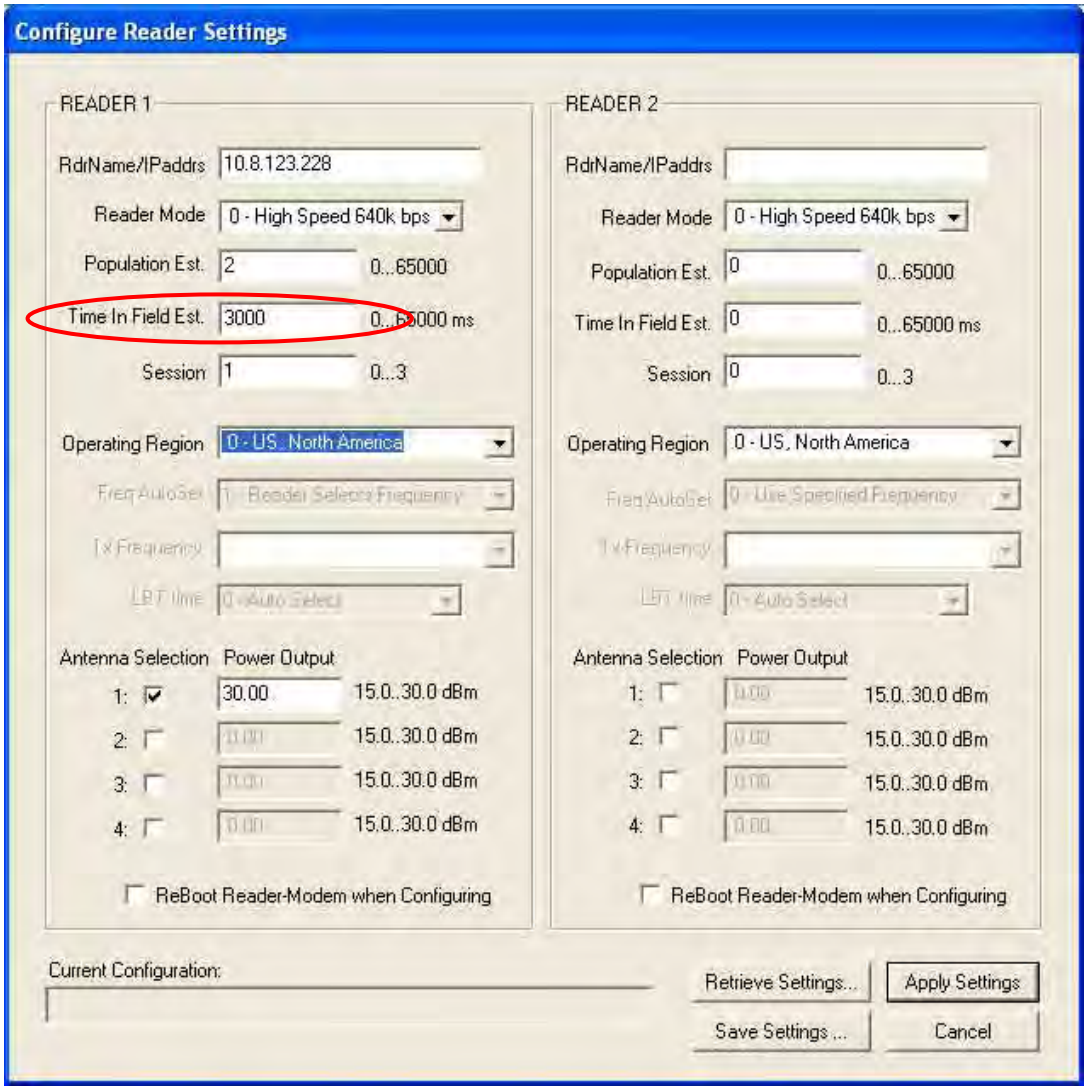


Figure 11-22

Cause: Selected Country is not the reader designed for

Make sure the Operating Region selected in reader configuration is the reader designed for. If the reader does not support operation for the selected region, it will not operate properly.

Copyright © Convergence Systems Limited, All Rights Reserved

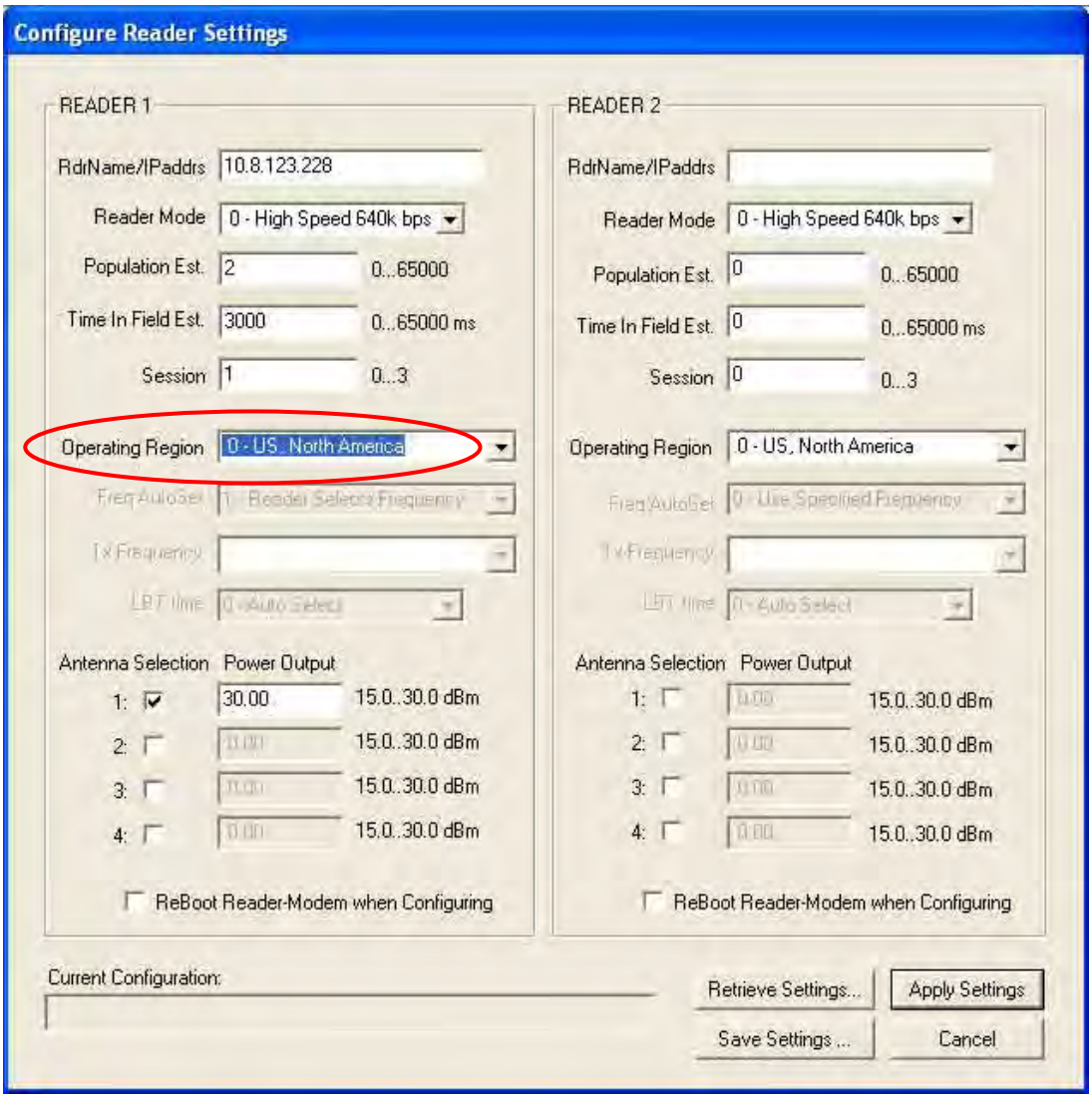


Figure 11-23

11.2.4 Programming Interface

11.2.4.1 getCaptureTagsRaw Cannot Get Newly Captured Tag

When the Trigger Mode is “Polling Trigger by Client”, “startInventory” command must be called before calling the “getCaptureTagsRaw” command to acquire new tag data. Repeat the following commands every time new tag data are required:

Command 1:

session_id=<login_session_id>&command=startInventory&mode=pollingTrigger

Copyright © Convergence Systems Limited, All Rights Reserved

Command 2:

session_id=<login_session_id>&command=getCaptureTagsRaw&mode=getEPC

Copyright © Convergence Systems Limited, All Rights Reserved

11.3 Bug Reporting: Format & Information Required

When the user reports a bug, he/she can dramatically improve return time for bug fix by submitting various information that can be captured from the reader and from a PC connected to the reader.

The reader needs to be debugged from both the reader side and the PC side. On the reader side, the entry point is the DB9 connector. The DB9 connector actually is a command console that displays the Linux OS screen outputs. One can use this port to obtain some important information that will help the debugging process. From the PC side, there are multiple means, including the browser, the command prompt and also specific windows software. In order to ease the debug and report the problem effectively, the following procedure should be followed to check if the reader's system is working properly. In addition, the screen shot and log capture is valuable for further analysis and in-depth debugging. PLEASE SEND THEM TO CSL TECHNICAL SUPPORT ENGINEERS AT THE MOMENT YOU ENCOUNTER PROBLEMS.

11.3.1 Prerequisite

Connect RS232 serial cable to the UART port (DB9 connector on the rear panel of the Reader) and PC.

Serial Port configuration:
Baud rate : 115200
Date : 8 bit
Parity : none
Stop : 1 bit
Flow control : none

Tera Term or equivalent terminal program is recommended since it is easier to capture the log data. Hyperterm has a bug such that if the message is too long then some of the older logs are scrolled to a point they are lost. Tera Term does not have this problem.

Copyright © Convergence Systems Limited, All Rights Reserved

11.3.2 Bug Reporting Procedure

Procedure 1 – reader console: ps

Type "ps" command to check process status. The processes marked in red color must be present (basically modem_ctrl, websvr and rfid_app). If they are not in the list, the processes have died.

EXAMPLE SCREEN CAPTURE:

```
root@CSL:/# ps
  PID  Uid     VmSize Stat Command
    1 root       1672 S    init
    2 root            S    [keventd]
    3 root            R    [ksoftirqd_CPU0]
    4 root            S    [kswapd]
    5 root            S    [bdflush]
    6 root            S    [kupdated]
    7 root            S    [mtdblockd]
   29 root       1604 S    /sbin/syslogd -p /var/logs
   32 root       1428 S    /sbin/klogd
   41 root            D    [ixp425_csr]
   42 root            S    [ixp425 ixp0]
   50 root            S    [jffs2_gcd_mtd0]
   51 root            S    [jffs2_gcd_mtd1]
   52 root            S    [jffs2_gcd_mtd2]
   53 root            S    [jffs2_gcd_mtd3]
  119 root       2268 S    /bin/sh /usr/sbin/logrotate_wrapper
  121 root       2284 S    /bin/sh /usr/sbin/sshd_wrapper
  123 root       1432 S    /usr/sbin/inetd
  125 root       2464 S    /usr/sbin/ntpd
  127 root       1380 S    /usr/mpr/xscale_modem_ctrl
  129 root       2388 S    -bash
  132 root       1652 S    sleep 86400
  136 root       3332 S    /usr/sbin/sshd -D
  147 root      12088 S    /tmp/usb_rfid/rfid_app
  148 root      13096 S    /tmp/usb_websvr/websvr/python /tmp/usb_websvr/websvr
  172 root      12088 S    /tmp/usb_rfid/rfid_app
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
  173 root     12088 S    /tmp/usb_rfid/rfid_app
  174 root     12088 S    /tmp/usb_rfid/rfid_app
  175 root     12088 S    /tmp/usb_rfid/rfid_app
  176 root     12088 S    /tmp/usb_rfid/rfid_app
  204 root      6580 S    sshd: root@pts/0
  209 root      2396 S    -bash
  217 root           S   [rpciod]
  276 root      1728 R    ps
root@CSL:/#
```

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 2 – reader console: free

Type “free” command to check memory usage. The free memory should be greater than 8000 (Kbytes in unit).

```
root@CSL:/# free
             total        used        free      shared     buffers
  Mem:       63124       48112       15012           0         344
  Swap:          0           0           0
  Total:     63124       48112       15012
root@CSL:/#
```

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 3 – reader console: top

Type "top" command to check CPU utilization. This will give us an idea which processes are taking up what amount of resources and whether they are taking up more or less than they should. (Note that "twistd" is actually the websvr process)

EXAMPLE SCREEN CAPTURE:

```
top - 09:11:29 up 45 min,  2 users,  load average: 1.28, 1.15, 1.01
Tasks:  34 total,   2 running,  32 sleeping,   0 stopped,   0 zombie
Cpu(s):   1.0% user,   7.2% system,   0.0% nice,  91.8% idle
Mem:     63124k total,    48372k used,    14752k free,      352k buffers
Swap:       0k total,       0k used,       0k free,    26540k cached

end capWin:5030 PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
  127 root     15   0   460  460  400 S  3.6  0.7   1:07.90 xscale_modem_ct
  284 root     16   0   984  984  824 R  1.6  1.6   0:02.62 top
  176 root     15   0  1728 1728 1404 R  0.3  2.7   0:00.99 rfid_app
    3 root     34  19     0    0    0 R  0.3  0.0   0:01.44 ksoftirqd_CPU0
  148 root     15   0 10564  10m 2560 S  0.3 16.7   0:33.09 twistd
    1 root     15   0   576  576  504 S  0.0  0.9   0:03.56 init
    2 root     RT   0     0    0    0 S  0.0  0.0   0:00.00 keventd
    4 root     25   0     0    0    0 S  0.0  0.0   0:00.00 kswapd
    5 root     25   0     0    0    0 S  0.0  0.0   0:00.00 bdflush
    6 root     15   0     0    0    0 S  0.0  0.0   0:00.09 kupdated
    7 root     15   0     0    0    0 S  0.0  0.0   0:00.00 mtdblockd
   29 root     15   0   688  684  584 S  0.0  1.1   0:00.04 syslogd
   32 root     15   0   560  560  428 S  0.0  0.9   0:00.04 klogd
   41 root     15   0     0    0    0 D  0.0  0.0   0:00.00 ixp425_csr
   42 root     15   0     0    0    0 S  0.0  0.0   0:00.00 ixp425 ixp0
   50 root     35  10     0    0    0 S  0.0  0.0   0:02.25 jffs2_gcd_mtd0
   51 root     30  10     0    0    0 S  0.0  0.0   0:00.00 jffs2_gcd_mtd1
   52 root     30  10     0    0    0 S  0.0  0.0   0:00.00 jffs2_gcd_mtd2
```

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 4 – reader console: ifconfig

Type “ifconfig” command to get network configuration (IP, mask, etc.). The IP address should be what is set before.

EXAMPLE SCREEN CAPTURE:

```
root@CSL:/# ifconfig
ixp0      Link encap:Ethernet  HWaddr 00:05:7B:22:00:19
          inet addr:192.168.25.245  Bcast:192.168.25.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6102 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1043 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:256
          RX bytes:1090054 (1.0 Mb)  TX bytes:462787 (451.9 Kb)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:311680 errors:0 dropped:0 overruns:0 frame:0
          TX packets:311680 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:41369618 (39.4 Mb)  TX bytes:41369618 (39.4 Mb)

root@CSL:/#
```

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 5 – reader console: ping

Type “ping” command to check network integrity. To test whether a reader can reach out to the IP network and the round-trip time is reasonable.
e.g. several ms for LAN environment

EXAMPLE SCREEN CAPTURE:

```
root@CSL:/# ping 192.168.25.1
PING 192.168.25.1 (192.168.25.1): 56 octets data
64 octets from 192.168.25.1: icmp_seq=0 ttl=64 time=1.2 ms
64 octets from 192.168.25.1: icmp_seq=1 ttl=64 time=1.1 ms
64 octets from 192.168.25.1: icmp_seq=2 ttl=64 time=1.3 ms
64 octets from 192.168.25.1: icmp_seq=3 ttl=64 time=1.4 ms
```

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 6 – reader console & PC command prompt: netstat

Netstat is a most useful function call to check the sockets being opened up on the reader side and on the PC side.

EXAMPLE SCREEN CAPTURE:

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 7 – PC browser: login

Login the reader from the internet browser. Check if you can still login. If not, it may due to the websvr process is down, the IP address is changed, or the password has been changed.

EXAMPLE SCREEN CAPTURE:

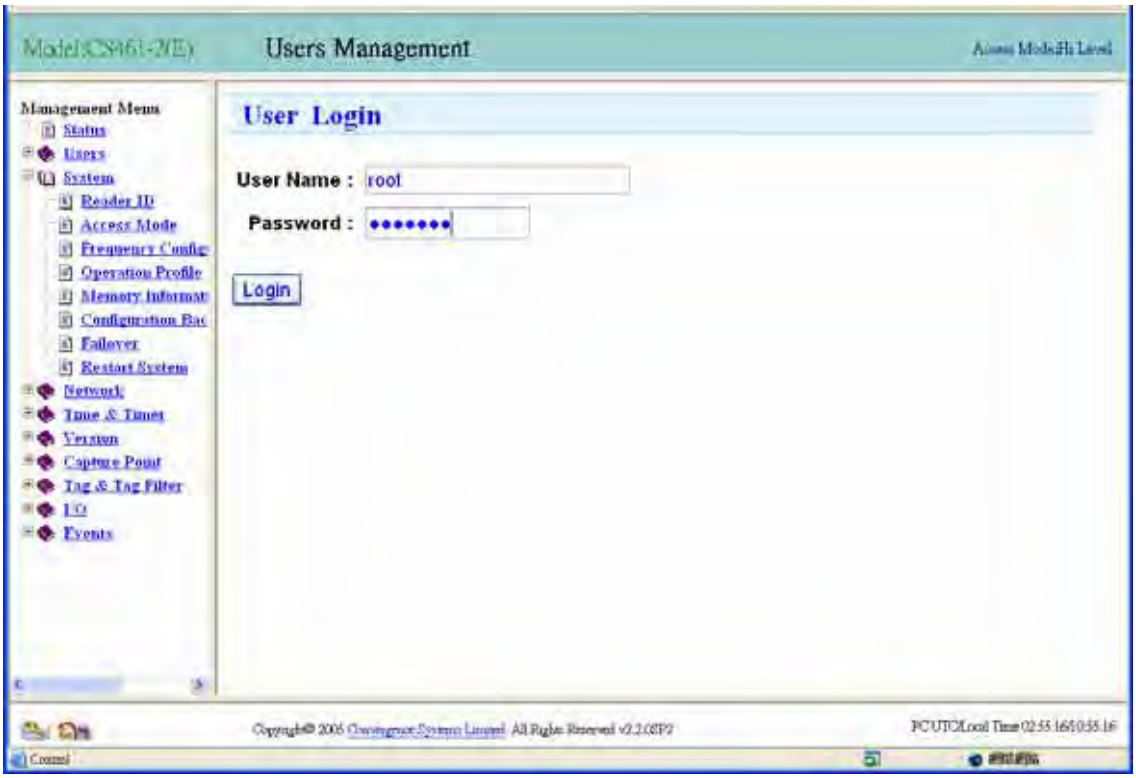


Figure 11-24

Procedure 8 – PC browser: get status

Click "Status" under "Management Menu" to check the system status. There should be no message in red color in Health Check. Moreover, free memory should be high value.

EXAMPLE SCREEN CAPTURE:

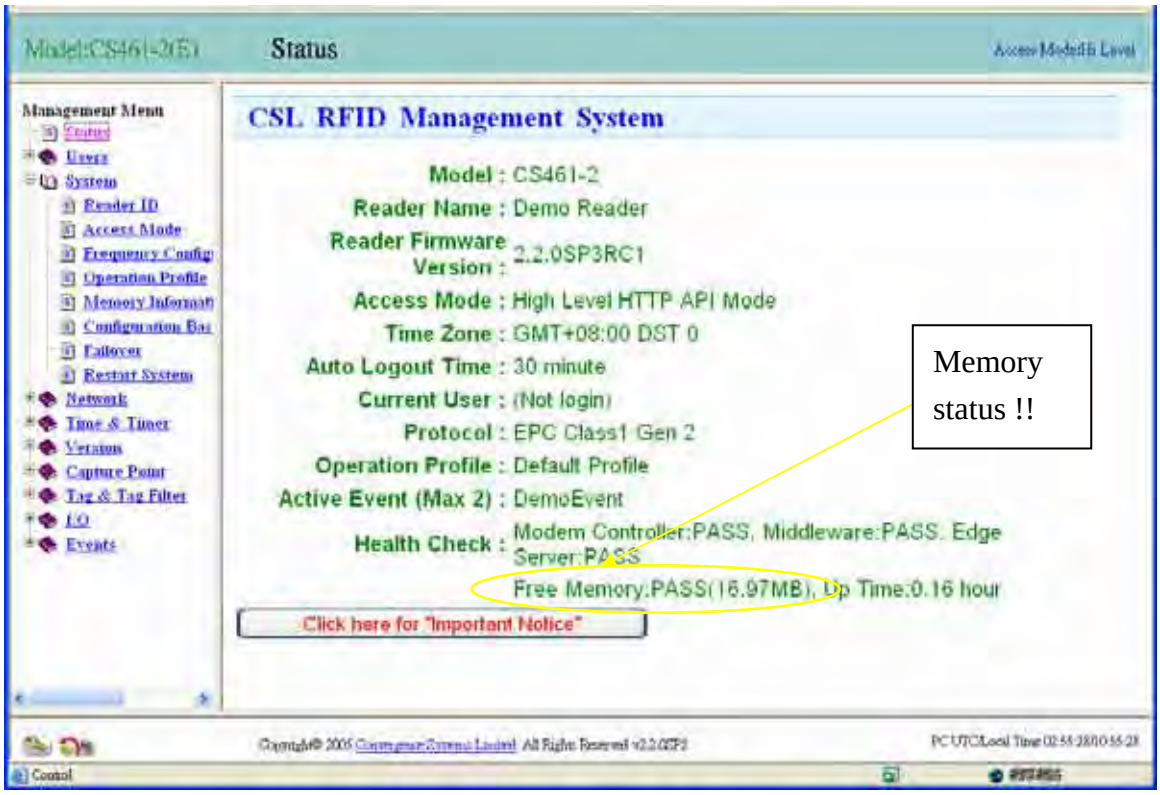
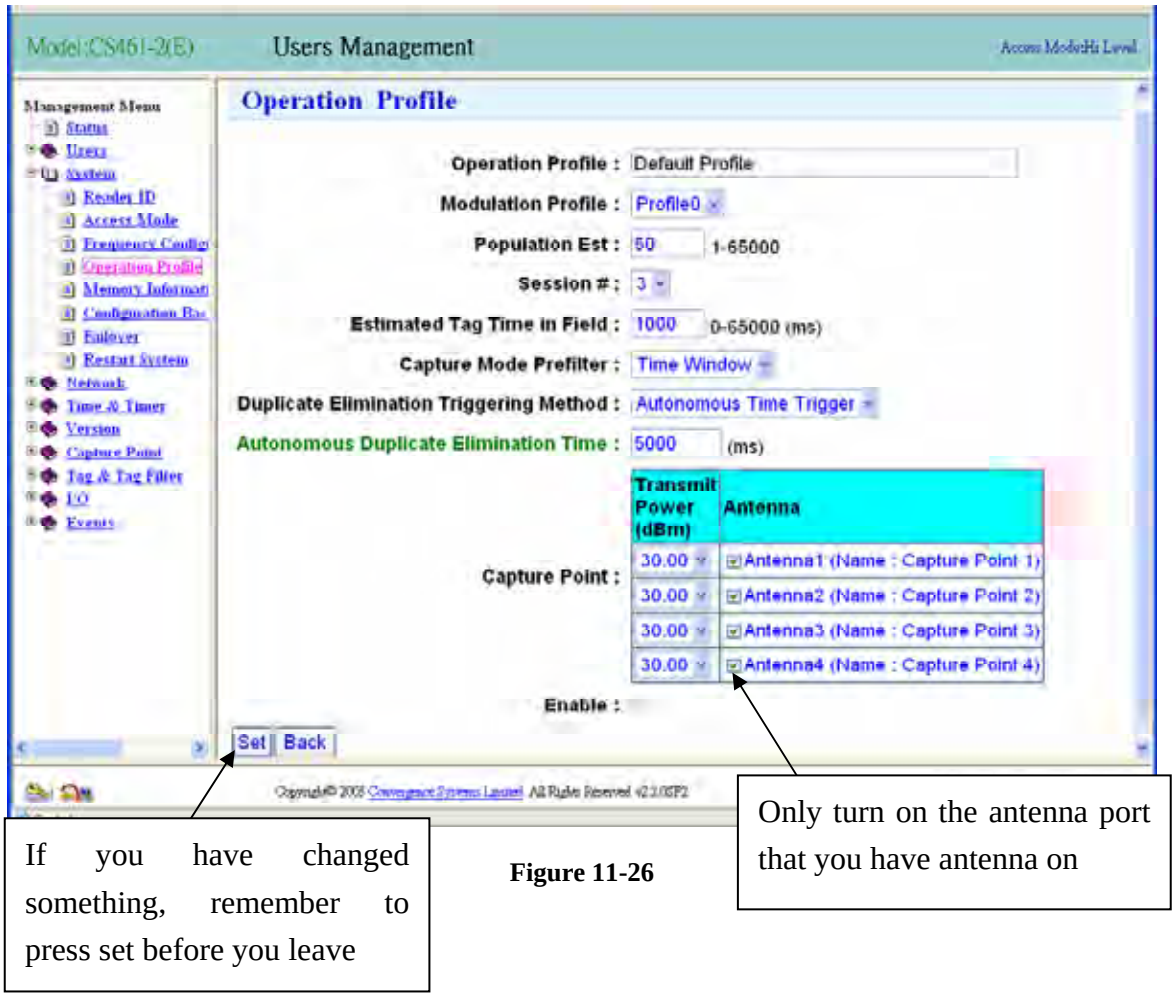


Figure 11-25

Procedure 9 – PC browser: operation profile

Click "Operation Profile" under "System Submenu" to check the Operation Profile setting. Make sure all settings are set properly.

EXAMPLE SCREEN CAPTURE:



If you have changed something, remember to press set before you leave

Only turn on the antenna port that you have antenna on

Figure 11-26

Procedure 10 – PC browser: failover

Click “Failover” under “System Submenu” to check if backlog is enabled.

EXAMPLE SCREEN CAPTURE:

Figure 11-27

Procedure 11 – PC browser: network configuration

Click "Configuration" under "Network Submenu" to check if the network configuration is set properly.

EXAMPLE SCREEN CAPTURE:

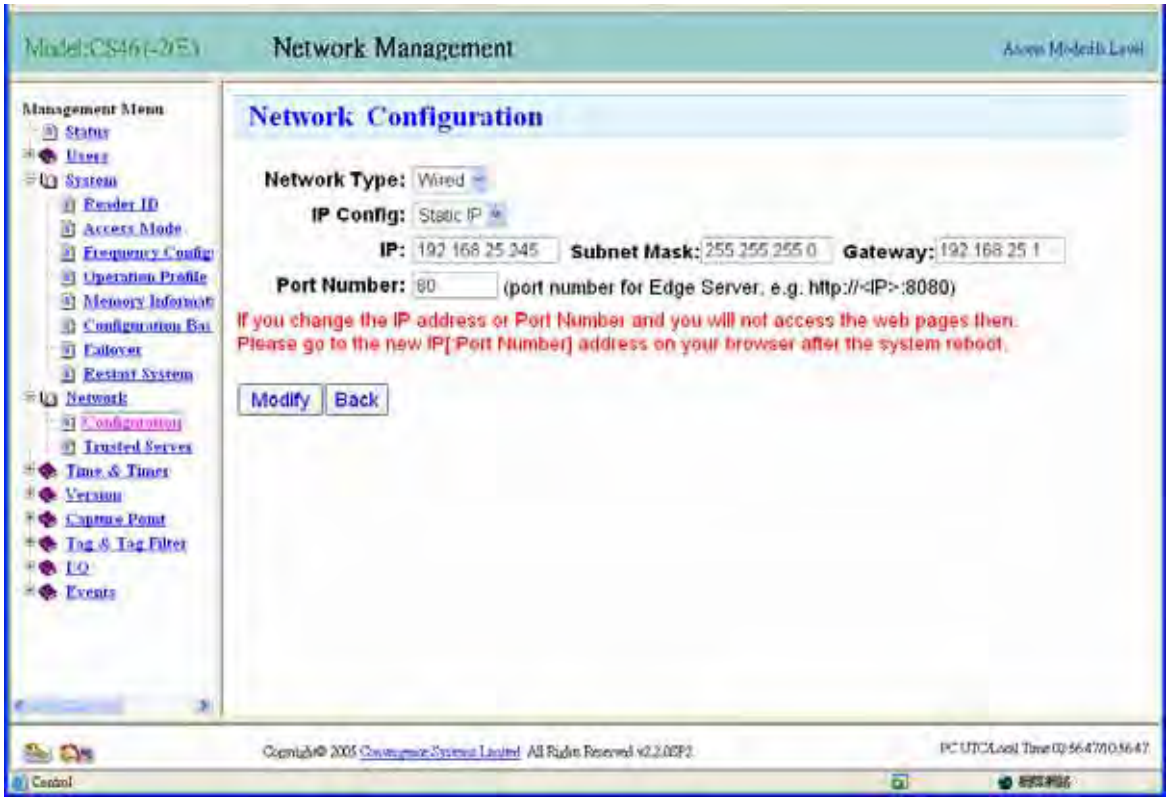


Figure 11-28

Procedure 12 – PC browser: date time

Click “Date/Time” under “Time and Timer Submenu” to check if system date/time is correct.

EXAMPLE SCREEN CAPTURE:

Figure 11-29

Procedure 13 – PC browser: version control

Click “Version Control” under “Version Submenu” to check firmware version information.

EXAMPLE SCREEN CAPTURE:

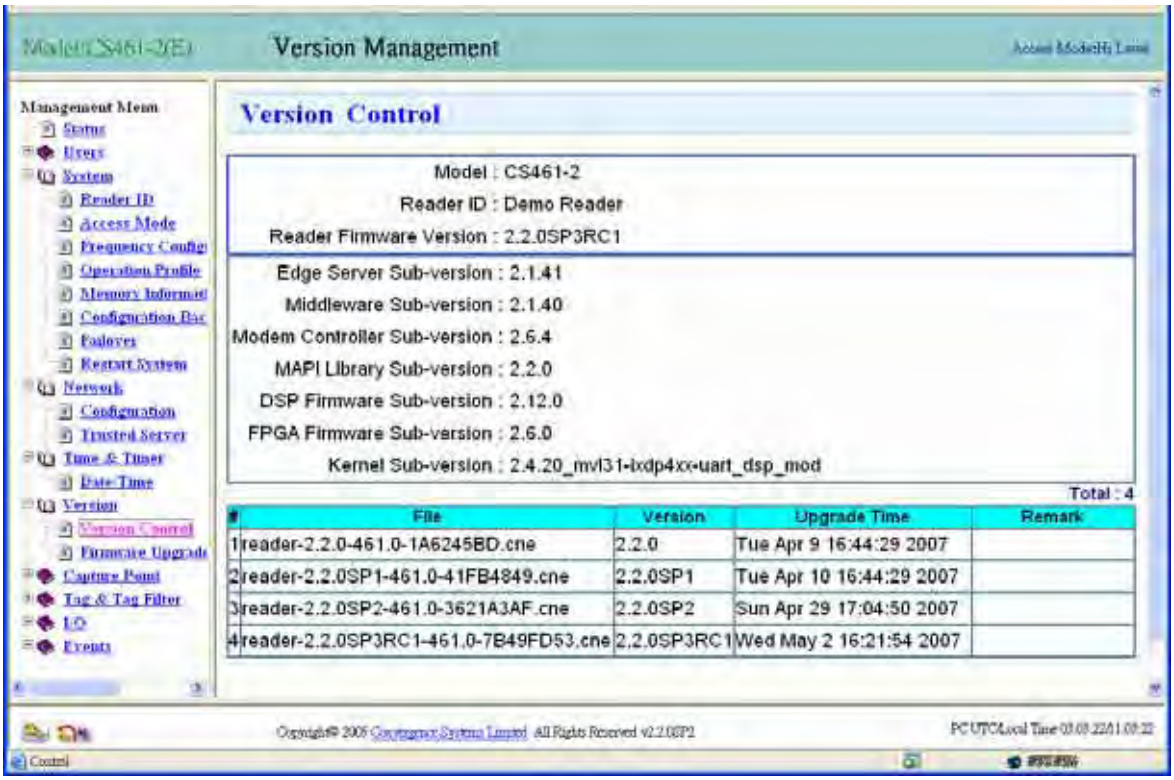


Figure 11-30

Procedure 14 PC browser: capture tags

Click “Capture Tags (Time Window Mode, Event Driven) - EPC” under “Capture Tags Testing Submenu”.

EXAMPLE SCREEN CAPTURE:

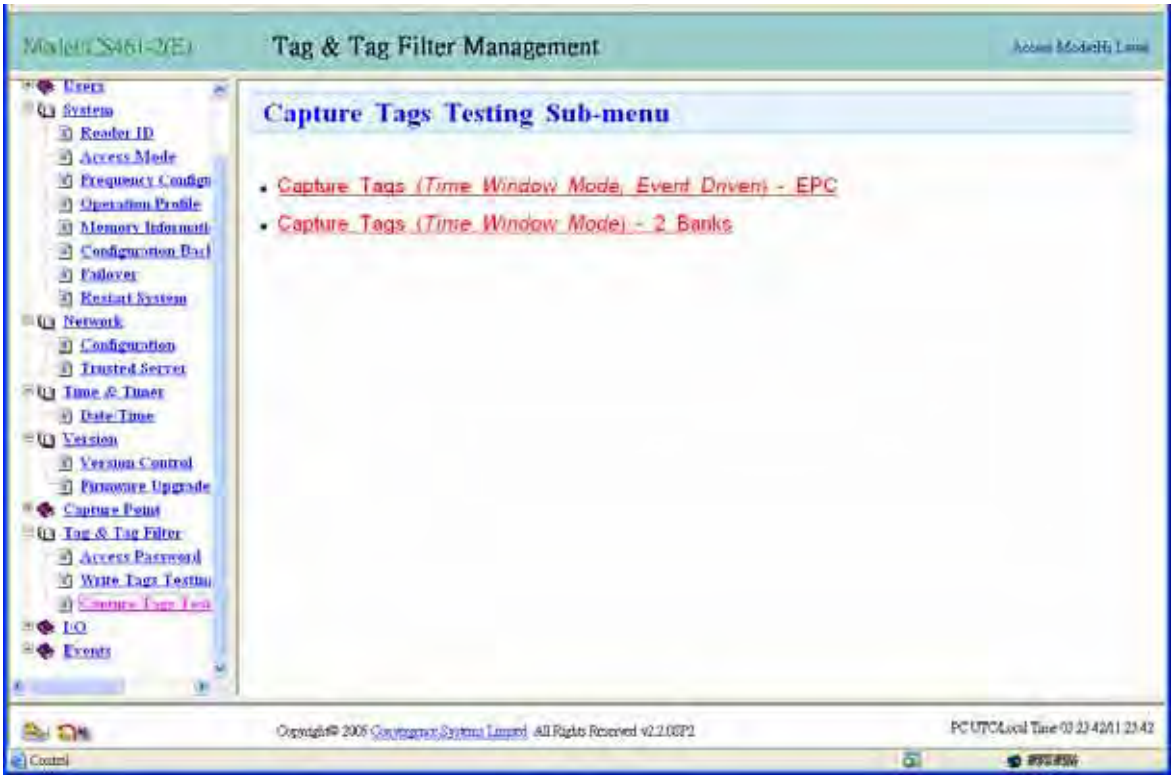
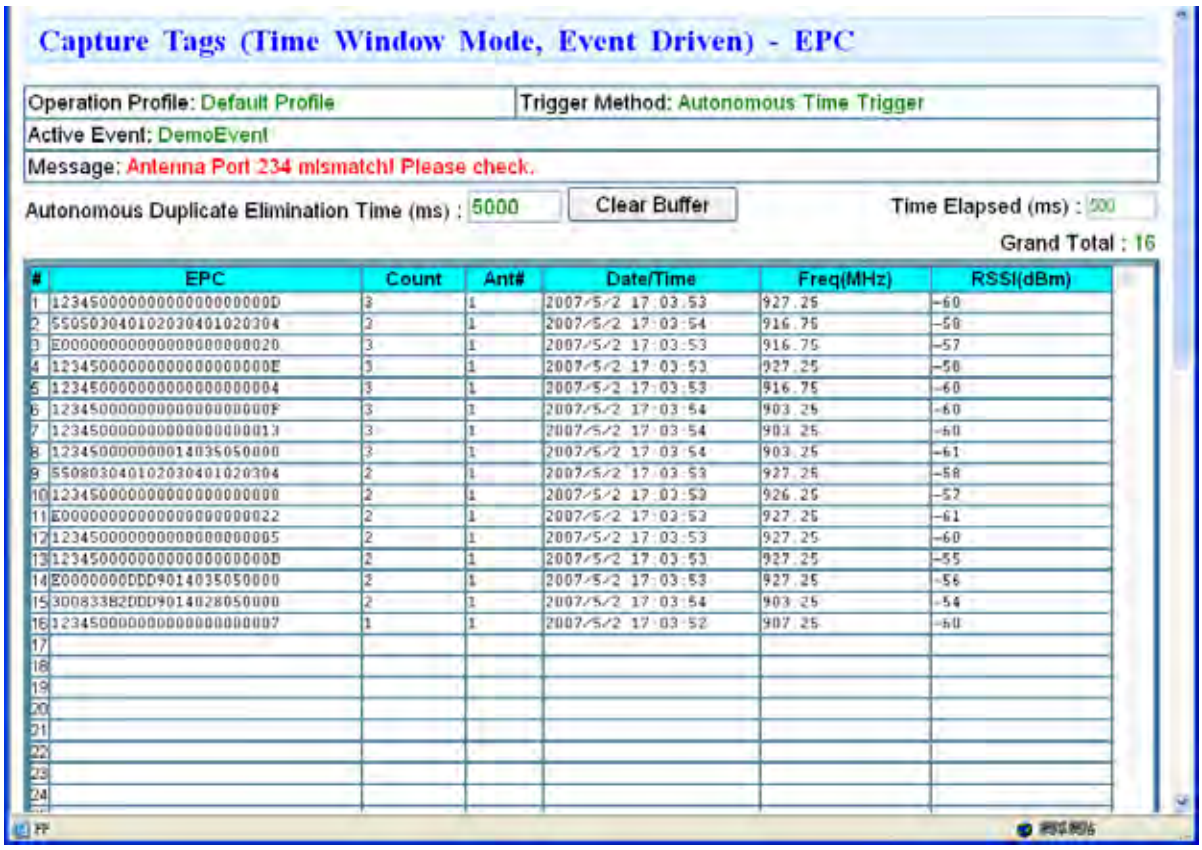


Figure 11-31

Tags should be read on the screen as shown below:



#	EPC	Count	Ant#	Date/Time	Freq(MHz)	RSSI(dBm)
1	12345000000000000000000D	3	1	2007/5/2 17:03:53	927.25	-60
2	550503040102030401020304	2	1	2007/5/2 17:03:54	916.75	-50
3	E0000000000000000000020	3	1	2007/5/2 17:03:53	916.75	-57
4	12345000000000000000000E	3	1	2007/5/2 17:03:53	927.25	-50
5	123450000000000000000004	3	1	2007/5/2 17:03:53	916.75	-60
6	12345000000000000000000F	3	1	2007/5/2 17:03:54	903.25	-60
7	123450000000000000000013	3	1	2007/5/2 17:03:54	903.25	-60
8	123450000000014035050000	3	1	2007/5/2 17:03:54	903.25	-61
9	550803040102030401020304	2	1	2007/5/2 17:03:53	927.25	-58
10	123450000000000000000000	2	1	2007/5/2 17:03:53	926.25	-57
11	E0000000000000000000022	2	1	2007/5/2 17:03:53	927.25	-61
12	123450000000000000000005	2	1	2007/5/2 17:03:53	927.25	-60
13	12345000000000000000000B	2	1	2007/5/2 17:03:53	927.25	-55
14	E0000000DDD9014035050000	2	1	2007/5/2 17:03:53	927.25	-58
15	300833B2DDD9014028050000	2	1	2007/5/2 17:03:54	903.25	-54
16	123450000000000000000007	1	1	2007/5/2 17:03:52	907.25	-60
17						
18						
19						
20						
21						
22						
23						
24						

Figure 11-32

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 15 – reader console: system log file

Look at System Log file (either from RS-232 console terminal, or you can telnet into the reader).

Type "cat system.log" under folder "/tmp/usb_main" to get the content of **system.log**, which is log info used by Edge Server. Here the time is Local Time.

EXAMPLE SCREEN CAPTURE:

```
root@CSL:/tmp/usb_main# cat system.log
2007/05/02, 16:23:30 first login
2007/05/02, 16:26:47 first login
2007/05/02, 18:17:42 first login
root@CSL:/tmp/usb_main#
```

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 16 – reader console: RFID log file

Look at RFID Log file (either from RS-232 console terminal, or you can telnet or SSH into the reader).

Type "cat rfid_app.log" under folder "/tmp/usb_main" to get the content of rfid_app.log, which is log info used by Middleware program. Here the time is UTC time.

EXAMPLE SCREEN CAPTURE:

```
root@CSL:/tmp/usb_main# cat rfid_app.log
  rfid_app starts...   Sun Apr 29 08:08:06 2007
  rfid_app starts...   Sun Apr 29 09:01:17 2007
  rfid_app starts...   Sun Apr 29 09:06:03 2007
  rfid_app starts...   Wed May  2 07:33:19 2007
  rfid_app starts...   Wed May  2 08:23:08 2007
  rfid_app starts...   Wed May  2 08:26:25 2007
  rfid_app starts...   Wed May  2 10:17:20 2007
root@CSL:/tmp/usb_main#
```

Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 17 – reader console: boot up history

Copy the boot up history log from the RS-232 console terminal program as shown below for example. It is valuable information for internal debugging.

EXAMPLE PRINTOUT:

```
+
Trying NPE-B...success. Using NPE-B with PHY 0.
Ethernet eth0: MAC address 00:05:7b:22:00:19
IP: 192.168.25.150/255.255.255.0, Gateway: 192.168.25.1
Default server: 0.0.0.0, DNS server IP: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version CSL Rev C Version 1.0.5 F - built 12:10:57, Dec
7 2006

Platform: Impinj Speedway Rev C (IXP42X 266MHz) BE
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x00000000-0x04000000, 0x000679c8-0x03fc1000 available
FLASH: 0x50000000 - 0x54000000, 512 blocks of 0x00020000 bytes each.
FLASH ID: 8919

Current time: 05/02/2007 8:25:39

Primary image (active): 0x51000000
No valid secondary image present: 0x50200000
== Executing boot script in 3.000 seconds - enter ^C to abort
RedBoot> fis load -b 0x02000000 Image_1
RedBoot> fis unlock -f 0x51a00000 -l 0x400000
... Unlock from 0x51a00000-0x51e00000: ................................
RedBoot> fis unlock -f 0x52020000 -l 0x1e00000
... Unlock from 0x52020000-0x53e20000: ........................................
.............................................................................
.............................................................................
.....................................
RedBoot> exec -b 0x02000000 -l 0xc0000 -c "console=ttyS0,115200 root=/dev/ram in
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
itrd=0x02200000,8000K"
Uncompressing Linux..................................................... done, b
ooting the kernel.
Linux version 2.4.20_mvl31-ixdp4xx-uart_dsp_mod (root@ericpc) (gcc version 3.3.1
 (MontaVista 3.3.1-3.0.10.0300532 2003-12-24)) #33 Thu Nov 23 10:39:19 HKT 2006
CPU: XScale-IXP4xx/IXC11xx [690541f1] revision 1 (ARMv5TE)
CPU: D undefined 5 cache
CPU: I cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
CPU: D cache: 32768 bytes, associativity 32, 32 byte lines, 32 sets
Machine: Impinj SPEEDWAY C
On node 0 totalpages: 16384
zone(0): 16384 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: console=ttyS0,115200 root=/dev/ram initrd=0x02200000,8000K
Calibrating delay loop... 266.24 BogoMIPS
Memory: 64MB = 64MB total
Memory: 55048KB available (1419K code, 248K data, 76K init)
XScale Cache/TLB Locking Copyright(c) 2001 MontaVista Software, Inc.
XScale cache_lock_init called
   Calling consistent alloc
   low_level_page initialized
   low_level_page @ 0xc4800000
      icache_lock_fn @ 0xc4800080
      dcache_lock_fn @ 0xc48000a0
      icache_unlock_fn @ 0xc4800098
      dcache_unlock_fn @ 0xc48000f0
Initializing TLB locking
TLB locking initialized
Dentry cache hash table entries: 8192 (order: 4, 65536 bytes)
Inode cache hash table entries: 4096 (order: 3, 32768 bytes)
Mount-cache hash table entries: 1024 (order: 1, 8192 bytes)
Buffer-cache hash table entries: 4096 (order: 2, 16384 bytes)
Page-cache hash table entries: 16384 (order: 4, 65536 bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
LSP Revision 1
ikconfig 0.5 with /proc/ikconfig
Starting kswapd
Disabling the Out Of Memory Killer
Journalled Block Device driver loaded
JFFS2 version 2.1. (C) 2001, 2002 Red Hat, Inc., designed by Axis Communications
 AB.
i2c-core.o: i2c core module version 2.6.2 (20011118)
i2c-dev.o: i2c /dev entries driver module version 2.6.2 (20011118)
i2c-algo-bit.o: i2c bit algorithm module version 2.6.2 (20011118)
i2c-proc.o version 2.6.2 (20011118)
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with MANY_PORTS SHARE_IRQ SERIAL_PCI en
abled
ttyS00 at 0xff000003 (irq = 15) is a XSCALE UART
ttyS01 at 0xff001003 (irq = 13) is a XSCALE UART
Hello World
RAMDISK driver initialized: 16 RAM disks of 12288K size 1024 blocksize
loop: loaded (max 8 devices)
NO QRY response
IXP425_Flash: Found 1 x16 devices at 0x2000000 in 16-bit mode
 Intel/Sharp Extended Query Table v1.4 at 0x010A
cfi_cmdset_0001: Erase suspend on write enabled
number of CFI chips: 2
Using buffer write method
kmod: failed to exec /sbin/modprobe -s -k RedBoot, errno = 2
RedBoot partition parsing not available
Using static MTD partitions.
Creating 5 MTD partitions on "IXP425_Flash":
0x010e0000-0x01a00000 : "SOP"
0x01a00000-0x01c00000 : "SCP"
0x01c00000-0x01e00000 : "CAP"
0x02020000-0x03e20000 : "CSL"
0x03f7f000-0x03f80000 : ""RedBoot config""
mtd: partition ""RedBoot config"" doesn't start on an erase block boundary -- fo
rce read-only
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 4096 bind 8192)
IP-Config: No network devices available.
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.95 (c) 1998-1999 Rebel.com
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 8000K
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 76K
serial console detected.  Disabling virtual terminals.
init started:  BusyBox v0.60.3 (2005.07.21-03:41+0000) multi-call binary
Starting system log daemon: syslogd.
Starting kernel log daemon: klogd.
Loading modules: /lib/modules/ixp400.o Using /lib/modules/ixp400.o
Module init.
/lib/modules/ixp425_eth.o Using /lib/modules/ixp425_eth.o
ixp425_eth:
Initializing IXP425 NPE Ethernet driver software v. 1.1
ixp425_eth: CPU clock speed (approx) = 0 MHz
ixp425_eth: ixp0 is using the PHY at address 0
ixp425_eth: ixp1 is using the PHY at address 1
/lib/modules/ds1302.o Using /lib/modules/ds1302.o
/lib/modules/drv_mdm.o ixp425_eth: ixEthMiiLinkStatus failed on PHY1.
       Can't determine
the auto negotiated parameters. Using default values.
Using /lib/modules/drv_mdm.o

Mounting local filesystems...
Hostname: CSL.
Cleaning: /etc/network/ifstate.
Setting up IP spoofing protection: rp_filter.
Disable TCP/IP Explicit Congestion Notification: done.
Configuring network interfaces: done.

Setting the System Clock using the Hardware Clock as reference...
System Clock set. Local time: Wed May  2 08:26:20 UTC 2007

ntpdate -b -s 140.142.16.34
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
Cleaning: /tmp /var/lock /var/run.
redirect to csl edge server
boot script for cust
Using /lib/modules/gpout.o
@hf101.GPOUT: <6>HPI: hpi_base=0x57000000, tx_buffer_size=1042, rx_buffer_size=4
096, hpi_major=16
Version: $Id: 0x020604F1 MAPI 2 Apr 18 2007 15:40:29 $
/tmp/modem_app.fifo creation success
Using /lib/modules/hpi.o
HPI: hpi_base=0x54000000, tx_buffer_size=1042, rx_buffer_size=4096, hpi_major=15
==== Firmware running ... ===
Starting OpenBSD Secure Shell server: sshd.
cp gpout.o
rmmod gpout
insmod gpout.o
Using /lib/modules/gpout.o
@hf101.GPOUT: <6>HPI: hpi_base=0x57000000, tx_buffer_size=1042, rx_buffer_size=4
096, hpi_major=16
Model.xml exist
FactoryCal.xml exist
=== Middleware running ... ===
=== Edge Server running ... ===
=== Up and running ... ===
Middleware Version: Gen2 RFID Middleware 2.1.40
libMAPI Version: 2.2.0

MAIN thread pid=147 ppid=1
Hardware Version:Rev E

CSL login: xml failOver setting=0
High level access
accessMode=1
Default Menu=7
DHCP is 0
IP Addr is 192.168.25.245
Mask is 255.255.255.0
GW Addr is 192.168.25.1
get_APconfig done
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
get_TagBaudRate done
read_config_file end
finished reading operating config...
main: read_opera_config done
finished loading config...
main: load_opera_config done
/tmp/websvricmd_to_reader.fifo creation successful
/tmp/reader_to_websvr.fifo creation successful
/tmp/command_in.fifo creation successful
/tmp/command_out.fifo creation successful
/tmp/CaptureTagsControl.fifo creation successful
/tmp/tagData.fifo creation successful
/tmp/EventParams.fifo creation successful
/tmp/rdr2Event.fifo creation successful
/tmp/Event2MA.fifo creation successful
/tmp/Event2rdr.fifo creation successful
/tmp/ntf2rdr.fifo creation successful
/tmp/modem_app.fifo already exists
/tmp/ntfData.fifo creation successful
killall: dhclient: no process killed
Cleaning: /etc/network/ifstate.
Reconfiguring network interfaces: SIOCADDRT: File exists
done.
eth0 Local ip addr: 192.168.25.245
eth0 MAC addr: 00.05.7b.22.00.19
eth0 netmask : 255.255.255.0
capture_mode=2
  <AvailableFreqList><Total country="U.S.A." fixed="50" hopping="50" regulatoryR
egion="0: FCC part 15.247" />
regulatoryRegion=FCC
Antenna=15
xmlget capmode=2
xmlget duptime=5000
modProfile=0
xmlget popEst=50
xmlget session=3
  txPowerPerAntenna[1]=188
  txPowerPerAntenna[2]=188
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
  txPowerPerAntenna[3]=188
  txPowerPerAntenna[4]=188
xmlget triggerMethod=1
  estimatedTagTimeInField=1000 ms
Read_tag thread pid=173 ppid=172
TagsCumulative.txt exists!
create lstn_skt
lstn setsockopt OK
ntf handler thread pid=174 ppid=172
Event thread started pid=175 ppid=172
Cannot find legal list file!
MAPI connected!
   cancel notifyThread
get_eventparams
invenEnableTrig=Always On
invenDisableTrig=Never Stop
triglog_mode=1
get_action1_param
   action_id1=DemoAction action_type=0
  act1:evtBatchNtfFlag=1075125732
get_action2_param
action_id2= action_type=0
size of action=2
   Number of event(s)=1 backLogSize=10000 blog bytes=28
In event_create_thread
    connMode=-1086334600
Done get evtparam!
Number 1 Event params:
    event_id=DemoEvent
    There are 1 triggers(s)
    trig_id=DemoTrigger
    trig_mode=1
      CP:0
      CP:1
      CP:2
      CP:3
    action_id=DemoAction
    action_type=0
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
    action_id=
    action_type=0
Connected to modem
bootmodem cmd sent
ntf:MAPI_DATA_TYPE_SOCKET_CONNECTION_STATUS_NTF status:0
  verXml=E-FCC
  - versionCode=1E0001
ntf:BOOT_MODEM_NTF result: successful

Modem state: INIT
temperature alarm set
  <AvailableFreqList><Total country="U.S.A." fixed="50" hopping="50" regulatoryR
egion="0: FCC part 15.247" />
regulatoryRegion=FCC
set to FCC  regResult=0
2007/05/02 16:26 LCT [-] Log opened.
******ntf:SET_REGULATORY_REGION_NTF result: successful

modem state: IDLE
evt start inven
  <AvailableFreqList><Total country="U.S.A." fixed="50" hopping="50" regulatoryR
egion="0: FCC part 15.247" />
regulatoryRegion=FCC
Antenna=15
xmlget capmode=2
xmlget duptime=5000
modProfile=0
xmlget popEst=50
xmlget session=3
  txPowerPerAntenna[1]=188
  txPowerPerAntenna[2]=188
  txPowerPerAntenna[3]=188
  txPowerPerAntenna[4]=188
xmlget triggerMethod=1
  estimatedTagTimeInField=1000 ms
   start purge 1178094391
evt:DemoEvent purged buffer 10
   end purge 1178094392
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
inven rsp=0
2007/05/02 16:26 LCT [-] twistd 2.4.0 (/tmp/usb_websvr/websvr/python 2.4.3) star
ting up
2007/05/02 16:26 LCT [-] reactor class: <class 'twisted.internet.selectreactor.S
electReactor'>
2007/05/02 16:26 LCT [-] Loading run.py...
2007/05/02 16:26 LCT [-] @hf.ElementTree.101
2007/05/02 16:26 LCT [-]
2007/05/02 16:26 LCT [-] accessmode 1
2007/05/02 16:26 LCT [-] country U.S.A.
2007/05/02 16:26 LCT [-] setCountry: U.S.A. Freq_USA.txt

end capWin:5019
2007/05/02 16:26 LCT [-] country= US_50
2007/05/02 16:26 LCT [-]
2007/05/02 16:26 LCT [-] lenFix= 50
2007/05/02 16:26 LCT [-] lenHop= 50
2007/05/02 16:26 LCT [-] name= U.S.A.
2007/05/02 16:26 LCT [-] len 220
2007/05/02 16:26 LCT [-] ============getRegionNumber U.S.A. 0
2007/05/02 16:26 LCT [-] start,end 902.75 927.25
2007/05/02 16:26 LCT [-] event convert to mw
2007/05/02 16:26 LCT [-] record= {'enable': 'true', 'triggering_logic': 'DemoTri
gger', 'event_id': 'DemoEvent', 'operProfile_id': 'Default Profile', 'resultant_
action': 'DemoAction', 'inventoryEnablingTrigger': 'Always On', 'event_log': 'fa
lse', 'inventoryDisablingTrigger': 'Never Stop', 'desc': 'Event Demo'}
2007/05/02 16:26 LCT [-] event_id= DemoEvent logic_id= DemoTrigger
2007/05/02 16:26 LCT [-] action_id_list= DemoAction
2007/05/02 16:26 LCT [-] action_id1= DemoAction action_id2=
2007/05/02 16:26 LCT [-] logic_id= DemoTrigger
2007/05/02 16:26 LCT [-] len logic= 1
2007/05/02 16:26 LCT [-] len action= 1
2007/05/02 16:26 LCT [-] action_mode Do Nothing (Only Show on Screen)
2007/05/02 16:26 LCT [-] act= DemoAction
2007/05/02 16:26 LCT [-] server_id=
2007/05/02 16:26 LCT [-] action_id2 is null!
2007/05/02 16:26 LCT [-] target group list= []
2007/05/02 16:26 LCT [-] convGrp
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
2007/05/02 16:26 LCT [-] There are 0 group ids
2007/05/02 16:26 LCT [-]
2007/05/02 16:26 LCT [-] There is 1 group ids
2007/05/02 16:26 LCT [-]
2007/05/02 16:26 LCT [-] command=setEventParams&update=yes
2007/05/02 16:26 LCT [-]
Cannot find legal list file!
   cancel notifyThread
get_eventparams
invenEnableTrig=Always On
invenDisableTrig=Never Stop
triglog_mode=1
get_action1_param
   action_id1=DemoAction action_type=0
  act1:evtBatchNtfFlag=1075125732
get_action2_param
action_id2= action_type=0
size of action=2
   Number of event(s)=1 backLogSize=10000 blog bytes=28
In event_create_thread
    connMode=-1086334600
Done get evtparam!
Number 1 Event params:
    event_id=DemoEvent
    There are 1 triggers(s)
    trig_id=DemoTrigger
    trig_mode=1
      CP:0
      CP:1
      CP:2
      CP:3
    action_id=DemoAction
    action_type=0
    action_id=
    action_type=0
fevt param
2007/05/02 16:26 LCT [-] fifo thread active...
Antenna 2 disconnected
```

Copyright © Convergence Systems Limited, All Rights Reserved

```
Antenna 3 disconnected
Antenna 4 disconnected

end capWin:5019

end capWin:5010
2007/05/02 16:26 LCT [-] twisted.internet.protocol.Factory starting on 8000
2007/05/02 16:26 LCT [-] Starting factory <twisted.internet.protocol.Factory ins
tance at 0x40867d4c>
2007/05/02 16:26 LCT [-] Loaded.
2007/05/02 16:26 LCT [-] nevow.appserver.NevowSite starting on 80
2007/05/02 16:26 LCT [-] Starting factory <nevow.appserver.NevowSite instance at
 0x40867e2c>
2007/05/02 16:26 LCT [-] logout automatically Thu Jan  1 00:00:00 1970
2007/05/02 16:26 LCT [-] normal logout

end capWin:5060

end capWin:5015

end capWin:5018

end capWin:5063
recv command=getReaderInfo

readerInfo:command=endGetReaderInfo&mw=2.1.40&libmapi=2.2.0&mc=2.6.4&dsp=2.12.0&
fpga=2.6.0

2007/05/02 16:27 LCT [-] !2.1.40=2.2.0=2.6.4=2.12.0=2.6.0
2007/05/02 16:27 LCT [-]
```

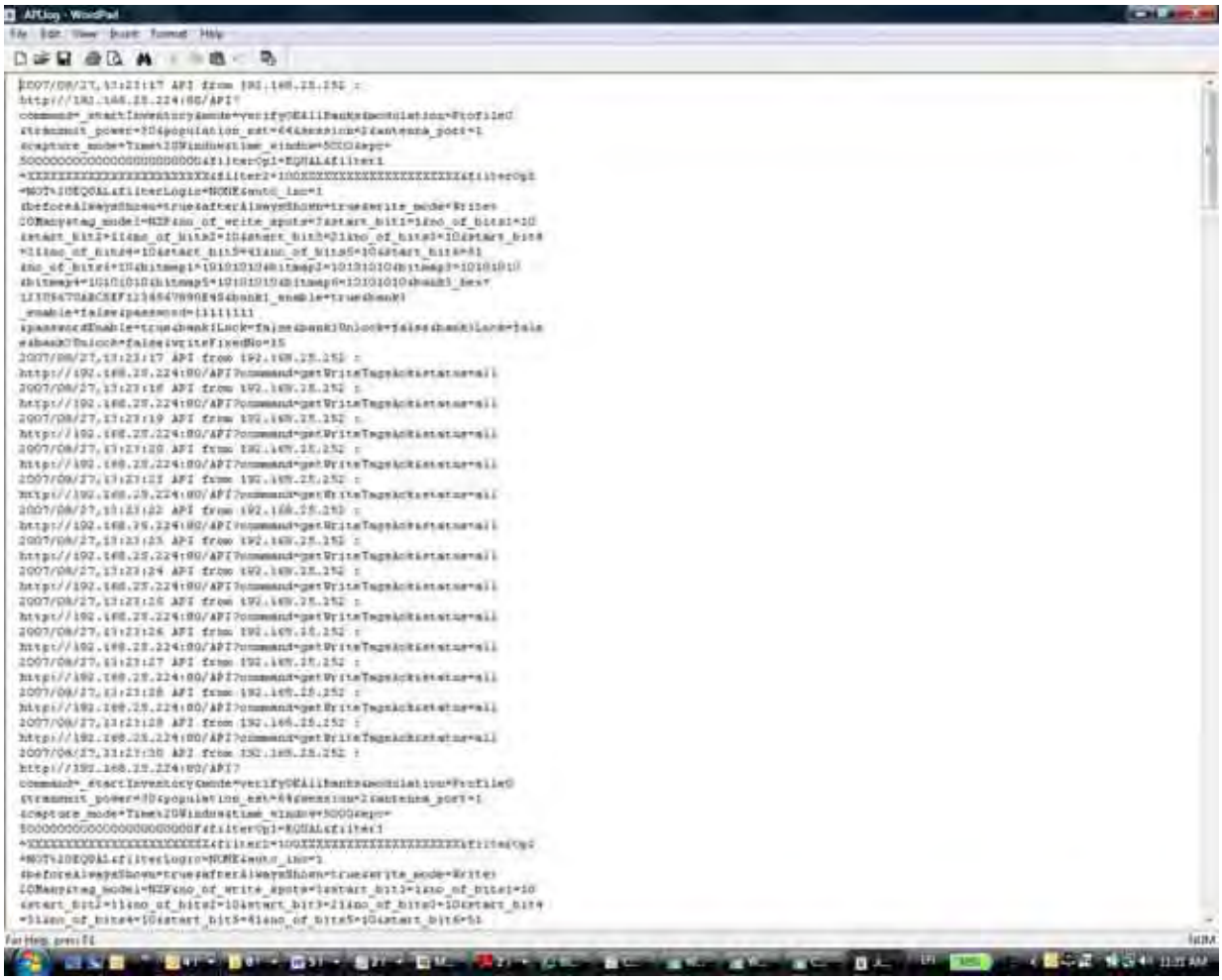
Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 18 – reader console: API Log

API log contains all the API calls that have been sent from a PC application to the reader that is being debugged. This is an excellent tool to make sure indeed the reader has received what the programmer wants it to receive.

Type “cat API.log” under folder “/tmp/usb_main” to get the content of API.log, which is log info of the API sent to the reader. If there are too many APIs sent, the file is also backed up to API_backup.log, also in the same directory. Every API code record comes with a time stamp that is based on Local Time of the reader.

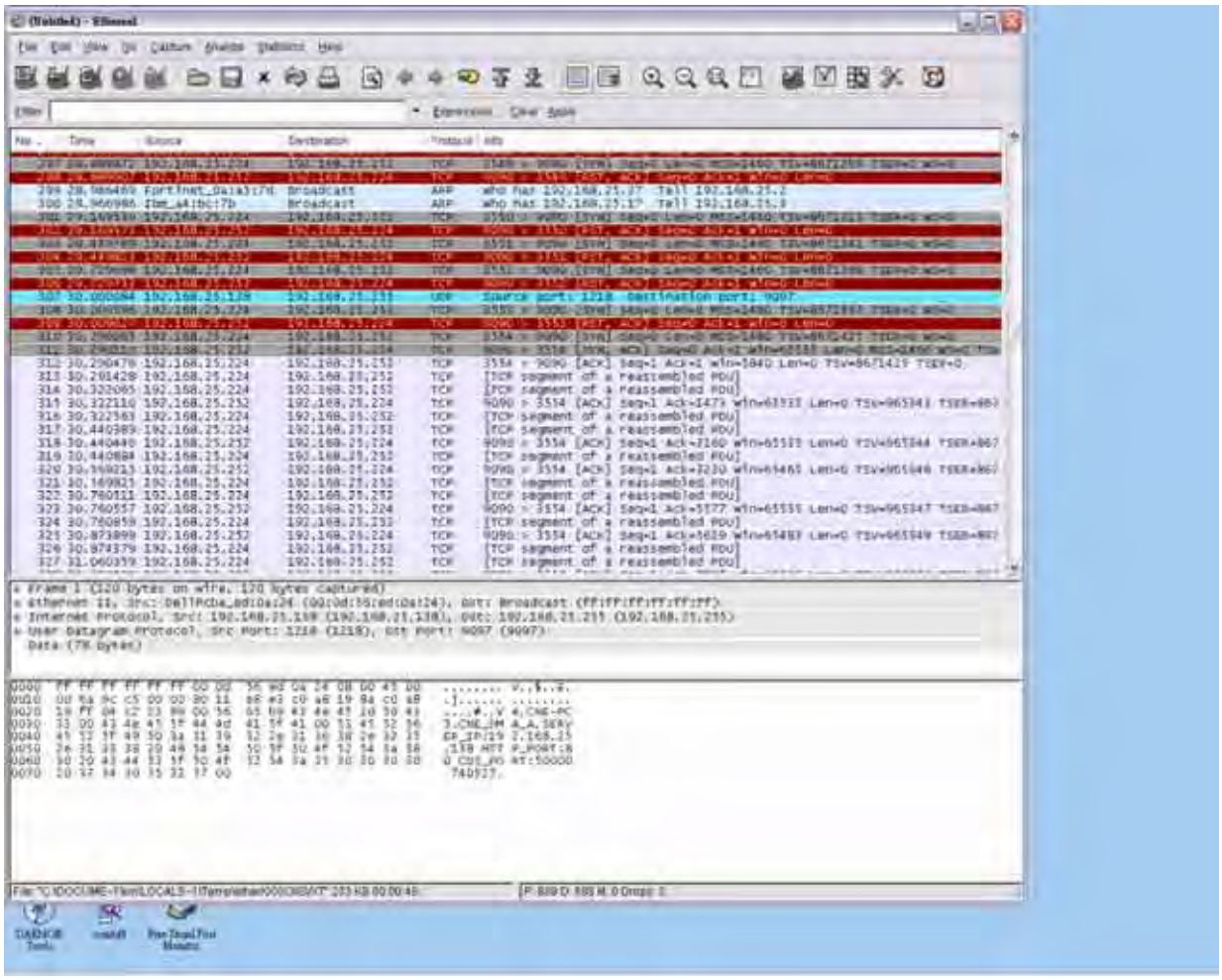
EXAMPLE SCREEN CAPTURE:



Procedure 19 – PC: Ethereal on PC

A most useful tool, this time on the PC side, is the Ethereal program (or any other available sniffer program in the market). With the Ethereal program, one can look at whether the reader has sent out tag data or API responses to the PC (not necessarily yet to the SI's program, but at least to the PC).

EXAMPLE SCREEN CAPTURE:



Copyright © Convergence Systems Limited, All Rights Reserved

Procedure 20 – PC: SSH from PC

Remote login the reader from PC by ssh or telnet with ID = root, password = csl (please wait about 20 seconds to login for ssh, 60 seconds for telnet).

e.g.

```
ssh 192.168.25.246
```

or

```
telnet 192.168.25.246
```

Repeat procedure 1 – 15 and procedure 14 – 15 to get the information.

Remark: For procedure 1 – 15 and procedure 14 – 15, when you use the RS-232 console terminal you would notice that the reader is constantly printing out some messages, and thus messing up with you input of commands. It does not happen if you use remote login method such as telnet or SSH to get the information. Of course with remote login method you cannot do procedure 16 which is the bootup log. Procedure 16 bootup log can only be seen on RS-232 console terminal.

Copyright © Convergence Systems Limited, All Rights Reserved

Appendix A. RFID Basics

Passive tag RFID technology involves the reader, the antenna and the tag.

The reader sends out energy in the relevant frequency band to the antenna via RF cables, and the antenna radiates the energy out. This energy impinges on an RFID tag.

The RFID tag consists of an antenna coupled to an RFID IC. This IC converts the AC voltage it receives at the antenna port to DC voltage that in turn is used to empower the digital circuit inside.

The digital circuit then turns on and off some components connected to the antenna port, thereby changing its scattering behavior, in a pre-designed clock rate.

This changing of antenna port parameters then causes a “modulation” of the back-scattered RF energy.

This modulated back-scattered energy is detected by the reader and the modulation is captured and analyzed.

Copyright © Convergence Systems Limited, All Rights Reserved

Appendix B. Glossary

Air interface
The complete communication link between an Interrogator and a Tag including the physical layer, collision arbitration algorithm, command and response structure, and data-coding methodology.

Autonomous time trigger
Each tag will only be reported once within a duplicate elimination time. See also duplicate elimination time.

Batch alert to server
Collected tag information are sent to server at the end of each duplicate elimination cycle (Time Window)

Capture point
Unique name corresponding to each of the four antennas

Command set
The set of commands used to explore and modify a Tag population.

Continuous wave
Typically a sinusoid at a given frequency, but more generally any Interrogator waveform suitable for powering a passive Tag without amplitude and/or phase modulation of sufficient magnitude to be interpreted by a Tag as transmitted data.

Cover-coding
A method by which an Interrogator obscures information that it is transmitting to a Tag. To cover-code data or a password, an Interrogator first requests a random number from the Tag. The Interrogator then performs a bit-wise EXOR of the data or password with this random number, and transmits the cover-coded (also called
ciphertext) string to the Tag. The Tag uncovers the data or password by performing a bit-wise EXOR of the received cover-coded string with the original random number.

Dense-Interrogator environment
An operating environment (defined below) within which the number of simultaneously active

Copyright © Convergence Systems Limited, All Rights Reserved

Interrogators is large relative to the number of available channels (for example, 50 active Interrogators operating in 50 available channels).

Duplicate elimination time

Time span of a duplicate elimination cycle, within which duplicate tags will be removed.

Duplicate Elimination Triggering Method

The method used to trigger inventory with duplicate elimination. See also autonomous time trigger and polling trigger by client.

Estimated tag time in field

An estimation of how long a tag will remain within the read zone of antenna

Event

An event defines action to be performed for a specific triggering logic. See also inventory enabling trigger, trigger, inventory disabling trigger, and resultant action.

Extended temperature range

–40 °C to +65 °C (see nominal temperature range).

Full-duplex communications

A communications channel that carries data in both directions at once. See also half-duplex communications.

Half-duplex communications

A communications channel that carries data in one direction at a time rather than in both directions at once. See also full-duplex communications.

Instant alert to server

Collected tag information are sent to server immediately as it is read

Inventoried flag

A flag that indicates whether a Tag may respond to an Interrogator. Tags maintain a separate inventoried flag for each of four sessions; each flag has symmetric A and B values. Within any given session, Interrogators typically inventory Tags from A to B followed by a re-inventory of Tags from B back to A (or vice versa).

Inventory enabling trigger

Copyright © Convergence Systems Limited, All Rights Reserved

The initial trigger that turns on the RF power of the reader to start doing inventory

Inventory Enabling Cycle

Time between an inventory enabling trigger and inventory disabling trigger.

Inventory disabling trigger

The trigger that turns off the RF power of the reader to stop doing inventory

Inventory round

The period between successive Query commands.

Inventory Search Mode

Method of reading tags by antenna. See also Single Target Large Population Inventory.

Modulation Profile

Way of transmitting information between tags and reader.

Multiple-Interrogator environment

An operating environment (defined below) within which the number of simultaneously active Interrogators is modest relative to the number of available channels (for example, 10 active Interrogators operating in 50 available channels).

Network failure data backlog

Tag data buffered in reader memory during network failure. Buffered tags are sent to trusted server when network is restored.

Nominal temperature range

–25 °C to +40 °C (see extended temperature range).

Operating environment

A region within which an Interrogator's RF transmissions are attenuated by less than 90dB. In free space, the operating environment is a sphere whose radius is approximately 1000m, with the Interrogator located at the © 2004, EPCglobal Inc. Page 13 of 94 31 January 2005 center. In a building or other enclosure, the size and shape of the operating environment depends on factors such as the material properties and shape of the building, and may be less than 1000m in certain directions and greater than 1000m in other directions.

Operating procedure

Copyright © Convergence Systems Limited, All Rights Reserved

Collectively, the set of functions and commands used by an Interrogator to identify and modify Tags. (Also known as the Tag-identification layer.)

Passive Tag (or passive Label)

A Tag (or Label) whose transceiver is powered by the RF field.

Permalock or Permalocked

A memory location whose lock status is unchangeable (i.e. the memory location is permanently locked or permanently unlocked) is said to be permalocked.

Persistent memory or persistent flag

A memory or flag value whose state is maintained during a brief loss of Tag power.

Physical layer

The data coding and modulation waveforms used in Interrogator-to-Tag and Tag-to-Interrogator signaling.

Polling Trigger by Client

Tags read are buffered in reader until client application polls the read result. A tag will only be reported once in each polling trigger.

Protocol

Collectively, a physical layer and a Tag-identification layer specification.

Q

A parameter that an Interrogator uses to regulate the probability of Tag response. An Interrogator commands Tags in an inventory round to load a Q-bit random (or pseudo-random) number into their slot counter; the Interrogator may also command Tags to decrement their slot counter. Tags reply when the value in their slot counter (i.e. their slot – see below) is zero. Q is an integer in the range (0,15); the corresponding Tagresponse probabilities range from 20 = 1 to 2–15 = 0.000031.

Resultant Action

Resultant action that will be enforced when an event logic is established

Single Target Large Population Inventory

A mode for reading a large number of tags at a time accurately. When this mode is used, tags that are read already will not respond to the reader for a short period of time. This can avoid the

Copyright © Convergence Systems Limited, All Rights Reserved

strong tags from dominating the week ones.

Session

An inventory process comprising an Interrogator and an associated Tag population. An Interrogator chooses one of four sessions and inventories Tags within that session. The Interrogator and associated Tag population operate in one and only one session for the duration of an inventory round (defined above). For each session, Tags maintain a corresponding inventoried flag. Sessions allow Tags to keep track of their inventoried status separately for each of four possible time-interleaved inventory processes, using an independent inventoried flag for each process.

Single-Interrogator environment

An operating environment (defined above) within which there is a single active Interrogator at any given time.

Singulation

Identifying an individual Tag in a multiple-Tag environment.

Slot

Slot corresponds to the point in an inventory round at which a Tag may respond. Slot is the value output by a Tag's slot counter; Tags reply when their slot (i.e. the value in their slot counter) is zero. See also Q (above).

Slotted random anticollision

An anticollision algorithm where Tags load a random (or pseudo-random) number into a slot counter, decrement this slot counter based on Interrogator commands, and reply to the Interrogator when their slot counter reaches zero.

Tag-identification layer

Collectively, the set of functions and commands used by an Interrogator to identify and modify Tags (also known as the operating procedure).

Tari

Reference time interval for a data-0 in Interrogator-to-Tag signaling. The mnemonic "Tari" derives from the ISO/IEC 18000-6 (part A) specification, in which Tari is an abbreviation for Type A Reference Interval.

Trigger

Copyright © Convergence Systems Limited, All Rights Reserved

A stimulus that causes the reader to recognize it and do something about it.

Trusted Server

Server for automatic data submission by the reader using the event engine.

Copyright © Convergence Systems Limited, All Rights Reserved

Appendix C. API Table

Copyright © Convergence Systems Limited, All Rights Reserved