

# Console Application User Guide Version 1.2

## System Requirements

- This application can only be installed on an IF61 release 2.
- If you want to be able to edit and recompile this application you must download and install the Intermec RFID Developer kit from [www.intermec.com](http://www.intermec.com) (go to Support and then Downloads).
- This application is written in C# using Visual Studio 2005.
- This sample project assumes you are familiar C# language and Intermec RFID products and features.

## Covered in this document

This document explains the examples provided in the sample code and covers how to install and run a console application on the IF61.

## Introduction

The purpose of this sample application is to provide examples of the common configuration options and RFID operations that most users will want to take advantage of. It is designed for you to play with the code. Please comment out code you do not want to use and uncomment code you do want to use. The program includes the ability to connect to the IF61 via a serial cable and send simple commands to control the application via Microsoft HyperTerminal software. The examples make use of the various IDL functions and features. The types of examples include the following:

- How to open a connection to the reader.
- How to enable IDL debugging logs.
- How to get and set attribute values.
- Different methods for reading tags.
- How to access tag data once read.
- How to write to tags.
- How to create GPIO triggers (for use with motion sensors, etc.).
- How to process GPIO events.
- How to create various IDL event handlers.

## How to use the sample application: VERY IMPORTANT READ THIS SECTION!

The sample application is designed so that it can be run on either your PC or on the IF61. To run the application on your PC, you will need to enter your IF61's IP address into the application and also set the following variable to TRUE:

```
oAlpha.bPCTestMode = true;
```

The IP address can be found in the file Program.cs in the function:

```
public void RunApp( ) {}
```

When the application is run on the IF61, it will use a local host network connection. You do not need to edit this value. It is very important to note that the sample code is setup for you to uncomment or comment out code and to play with it. It is not meant to be a functioning demo application.

### **Sample Functions List for Program.cs**

The purpose of this class contains the primary application calls. All of the primary functions are called from within the Program.cs file.

- `static void Main(string[] args)`
- `public class myAppsThread`
  - `public void RunApp()`
  - `public bool OpenRFIDReader(string sURL)`
  - `public bool OpenComPort(string sPort)`
  - `private void ProcessSerialCommand(string sCMD)`

### **Sample Functions List for SerialPortClass.cs**

The purpose of this class is to provide example serial port communications between the IF61 and the outside world.

- `public void CloseSerialPort()`
- `public bool OpenSerialPort(string sCOMPort, string sVersion)`
- `public bool IsBytesToRead()`
- `public string GetSerialData()`
- `public bool SendDataSerialPort(string sMsg)`

### **Sample Functions List RFIDReaderClass.cs**

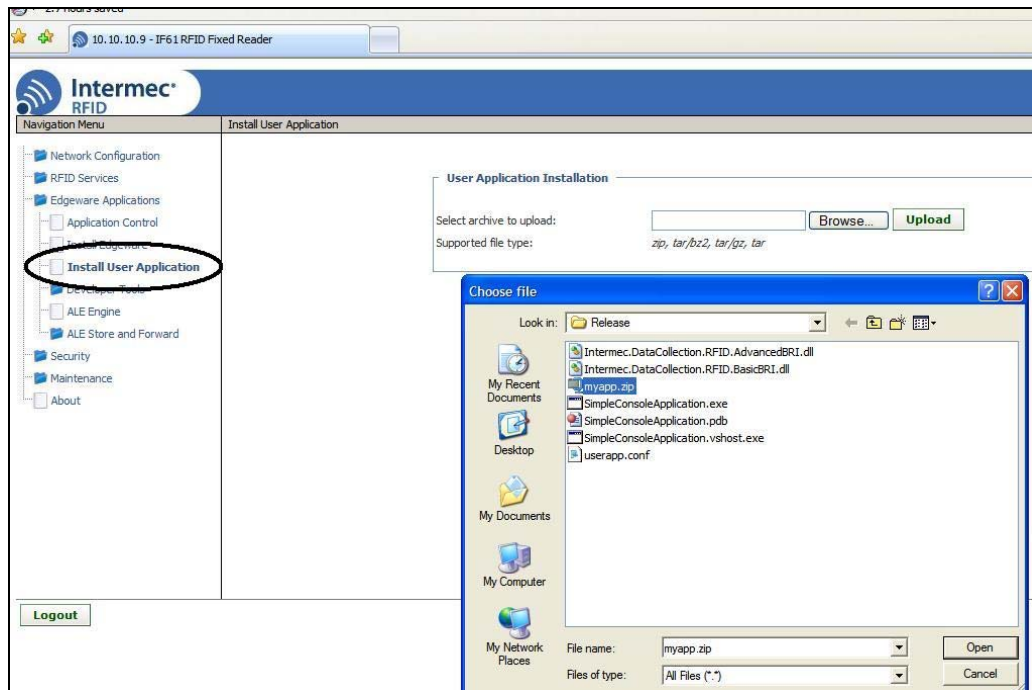
The purpose of this class is to provide examples of various RFID related operations

- `public void CloseReader()`
- `public bool CreateReader(string sURL)`
- `private void CreateTriggers()`
- `private void ConfigGen2Settings()`
- `private bool GetAttributeList()`
- `private void ReadTagsReportEvent()`
- `public void ReadTagsReportEventAll()`
- `private void ReadTagsReportDirect()`
- `private void ReadTagsReportNo()`
- `private void WriteTag()`
- `private void LoadTags(BRIRReader tRdr, int iRdrIndex)`
- `private bool AddEventHandlers()`
- `void brdr_EventHandlerRadio(object sender, EVTADV_Radio_EventArgs EvtArgs)`
- `void brdr_EventHandlerGPIO(object sender, EVTADV_GPIO_EventArgs EvtArgs)`
- `private void LogTags(string sTagData)`

- `private Int64 CheckTagList(string sID)`
- `void brdr_EventHandlerTag(object sender, EVTADV_Tag_EventArgs EvtArgs)`

### How to install the application on the IF61

- Open the web browser interface on the IF61.
- Go to *Edgware Applications* and select the *Install User Application* link.



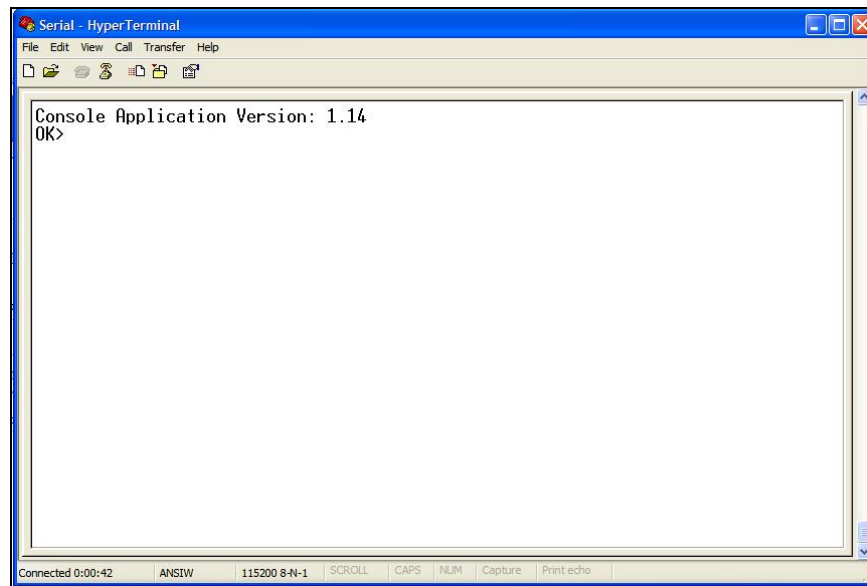
**Figure 1: Installing the application on an IF61**

- Click on Browse and select *myapp.zip* file (see figure 1).
- Click on the *Upload* button. The file will be loaded.
- After the application is loaded you should automatically see the *Application Control* page which is located under *Edgware Applications* (see figure 2).
- Before starting the application open HyperTerminal on your PC or some other serial port monitoring application (115200, N 8, 1 no flow control).
- Click on the green arrow button to start the application (figure 2). It will switch to a red box when the application is running. Click on the red box to stop the application. When the application starts you will see a version message appear in HyperTerminal (see figure 3). The application is now ready. If you trigger the motion sensor it will attempt to read tags and return them to HyperTerminal (or what ever serial port monitoring application you have connected).



**Figure 2: Application Control Page**

- If you want to upgrade or reinstall the version of the application that is running on the IF61, you must first remove the current version. Click on the *Uninstall* button on the far right of the screen to remove the current version.



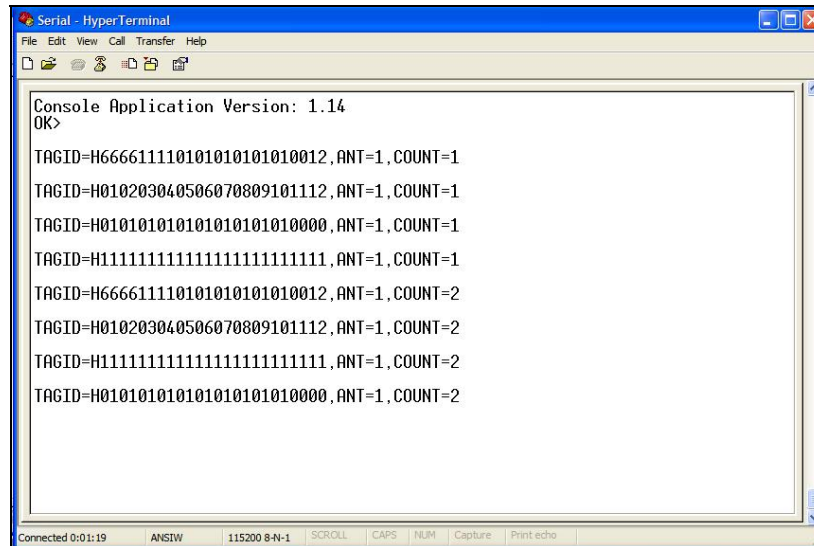
**Figure 3: Application version message**

## Tag Data

When tags are read the EPC ID, the antenna the tag was read on, and the number of times the tag was read will be sent out the serial port of the IF61 (see figure 4). Since the application uses EVENT mode, tags will be read only one timer per read. If tags remain in the read zone their count will be increment each time the motion sensor is triggered. The application keeps an internal list of tags its read. It stores the EPC ID, antenna, and count. The application stores the tag information based on which antenna it was read on. So if a tag is read on all four antennas, it will appear four times in the tag list. It will keep a separate count for the number of times it was read on each antenna. The data format sent to HyperTerminal is as follows:

**TAGID=XXXXXXXXXXXXXXXXXXXXXXXXXX,ANT=X,COUNT=XCRLF**

Where CRLF = carriage return line feed. There are no spaces inserted into the data string. A comma is used to separate the fields. Each EPC ID will be on a new line IF61 (see figure 4).



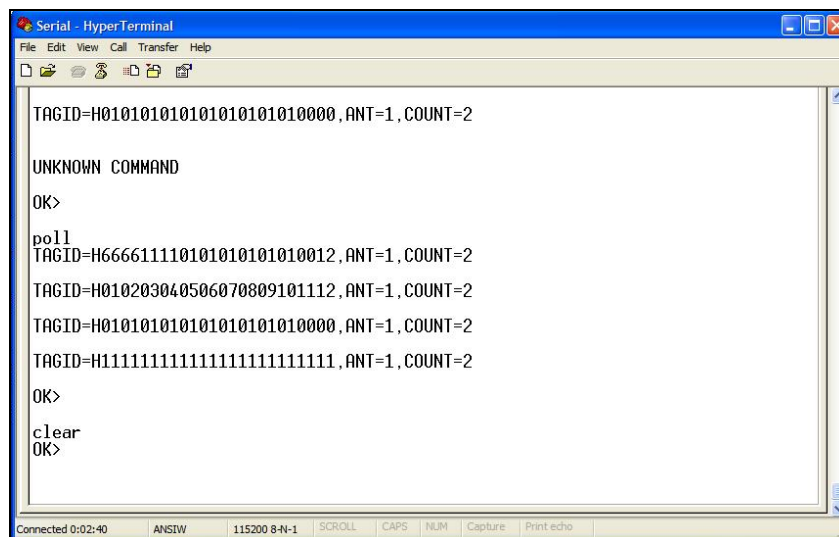
```
Serial - HyperTerminal
File Edit View Call Transfer Help
Console Application Version: 1.14
OK>
TAGID=H6666111101010101010012,ANT=1,COUNT=1
TAGID=H010203040506070809101112,ANT=1,COUNT=1
TAGID=H0101010101010101010000,ANT=1,COUNT=1
TAGID=H111111111111111111111111,ANT=1,COUNT=1
TAGID=H6666111101010101010012,ANT=1,COUNT=2
TAGID=H010203040506070809101112,ANT=1,COUNT=2
TAGID=H111111111111111111111111,ANT=1,COUNT=2
TAGID=H0101010101010101010000,ANT=1,COUNT=2
Connected 0:01:19  ANSIW  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

Figure 4: Tag data sent to HyperTerminal

### Serial Interface Commands

There are a number of commands that can be sent to the console application via the serial interface. These commands are as follows (also see figures 4-5):

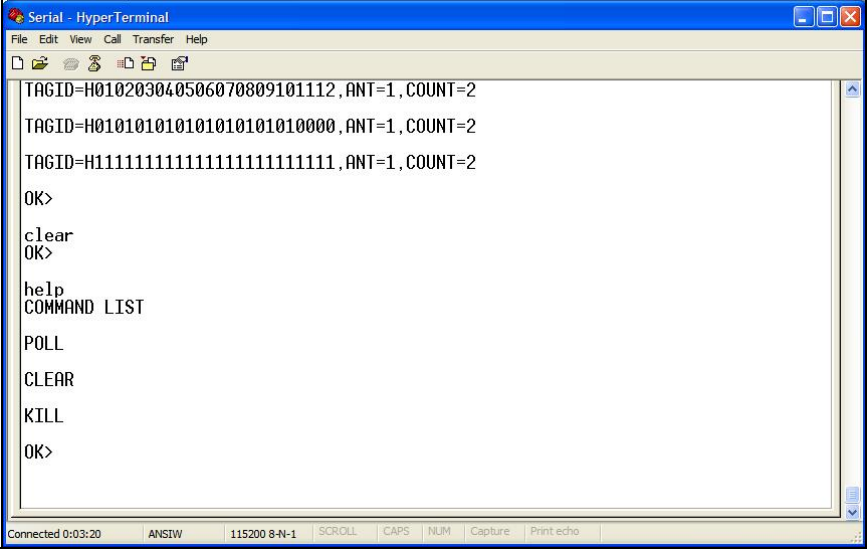
- HELP – will return a list of commands that you can use via the serial interface.
- POLL – will return the entire tag list.
- CLEAR – will clear the tag list.
- KILL – will stop the console application. You should also stop it via the web browser interface. When you issue the KILL command both the TRIGGERS and macros will be deleted. They are created each time you start the application.



```
Serial - HyperTerminal
File Edit View Call Transfer Help
TAGID=H0101010101010101010000,ANT=1,COUNT=2
UNKNOWN COMMAND
OK>
poll
TAGID=H6666111101010101010012,ANT=1,COUNT=2
TAGID=H010203040506070809101112,ANT=1,COUNT=2
TAGID=H0101010101010101010000,ANT=1,COUNT=2
TAGID=H111111111111111111111111,ANT=1,COUNT=2
OK>
clear
OK>
Connected 0:02:40  ANSIW  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

Figure 5: POLL and CLEAR commands

All command responses are terminated with an OK>CRLF. If you issue invalid command the application will return an UNKNOWN COMMAND message.



The screenshot shows a HyperTerminal window titled "Serial - HyperTerminal". The window has a menu bar (File, Edit, View, Call, Transfer, Help) and a toolbar. The main text area displays the following output:

```
TAGID=H010203040506070809101112,ANT=1,COUNT=2
TAGID=H0101010101010101010000,ANT=1,COUNT=2
TAGID=H11111111111111111111111111,ANT=1,COUNT=2
OK>
clear
OK>
help
COMMAND LIST
POLL
CLEAR
KILL
OK>
```

At the bottom of the window, a status bar shows "Connected 0:03:20", "ANSIW", "115200 8-N-1", and several checkboxes for "SCROLL", "CAPS", "NUM", "Capture", and "Print echo".

### How to create an install file for the IF61

If you wish to modify the console application you will need to create a new install file. This is very simple to do. After you make your changes to the source code, compile it. Then copy the .exe file to the *myapp.zip* file. You can rename the zip file if you want to. If you want to change the name of the .exe file, you will have to edit the *userapp.conf* file as well. The *myapp.zip* file contains four files. There are two RFID dll files, the application .exe file, and the *userapp.conf* file. The *userapp.conf* file contains the installation information that the IF61 needs to install and configure the application in the web browser interface. The file contains three lines which are self explanatory:

- AUTOSTART=false
- RUNAFTERINSTALL=false
- CMDLINE=./SimpleConsoleApplication.exe

Make any changes you need to and then save the file. Make sure that Microsoft's Notepad editor does not append a .txt to the end of the file name. It has a bad habit of doing this. Copy the new *userapp.conf* file into the *myapp.zip* and load it on to the IF61.