

REAL Protocol – Express

Version
2018.Q3

Software User Guide



Last Update: 08-01-2018

TABLE OF CONTENTS

Introduction	3
Application Installation	4
Application Directory Structure	4
Starting the Application	5
Application Ribbon Bar	6
Protocol Data	8
Using the Program	9
Filters	10
Settings XML File	11
Template XML Files	12
Application Licensing	14
Debugging Protocol Maps	16

INTRODUCTION

The REAL Protocol - Express program is a simple-to-use application to display the protocol map from a device. The application uses an RS-232 port or an Ethernet port to connect to the device.

This application was designed to work on the following operating systems:

- Windows XP (32-bit)
- Windows 7 (32-bit, 64-bit)
- Windows 10 (64-bit)
- **NOTE: Requires .NET 4.0 or later**

The application uses XML files for configuration settings and protocol map description templates.

NOTE: *Each protocol is not fully implemented in REAL Protocol - Express, but rather a smaller subset of the most commonly used features. If any protocol feature is missing please send an email detailing the missing features and we will add it to the next version of the application.*

We will make every effort to promptly answer your questions and add new features as soon as possible.

APPLICATION INSTALLATION

The REAL Protocol - Express application does not require an installation program, but normally the application is deployed using one. Manual installation consists of simply copying all files included with the application to a folder on your computer.

NOTE: *The REAL Protocol - Express application uses Microsoft .NET Framework version 4.0 and has built-in security requirements. If the application is installed on a network server the application will assume the .NET components are not trusted and may not run.*

NOTE FOR WINDOWS 7: *In some cases the “Program Files” directory may require administration access to install an application. The installation program will default to the “C:\Users\Public” directory which does not require administration rights.*

APPLICATION DIRECTORY STRUCTURE

The REAL Protocol - Express application uses a specific directory structure to store system components and files. The list below describes the directories used by the system:

- **GUI** - Third-party components used by the graphical interface
- **Help** - Contains all system documentation in PDF format
- **DNP3 Templates** – User created XML template files defining the DNP3 map points
- **Modbus Templates** – User created XML template files defining the Modbus map points
- **Root Directory** - The start-up directory contains components for the visual part of the application and default system settings

STARTING THE APPLICATION

The application can be started by clicking on the “REAL Protocol - Express.exe” program. Since the application is a trial version by default, you will see the following dialog box when the application loads:

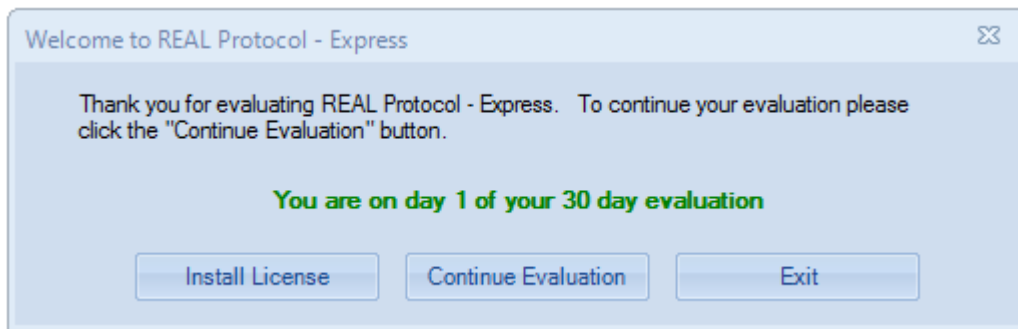


Figure 1

Clicking the “Continue Evaluation” button will start the application under evaluation mode. Evaluation mode is a full version of the application.

Clicking the “Install License” button will open the following dialog box where the user license key can be entered:

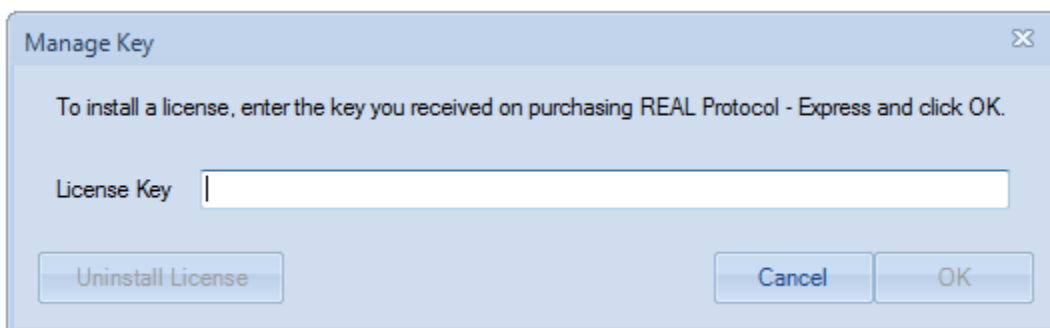


Figure 2

NOTE: The REAL Protocol - Express license key will be sent in an email message when a license is purchased.

APPLICATION RIBBON BAR

The ribbon bar at the top of the screen (shown in Figure 3) displays the main features of the application. Most of the protocol settings are handled internally by the application to simplify the process of displaying protocol maps.

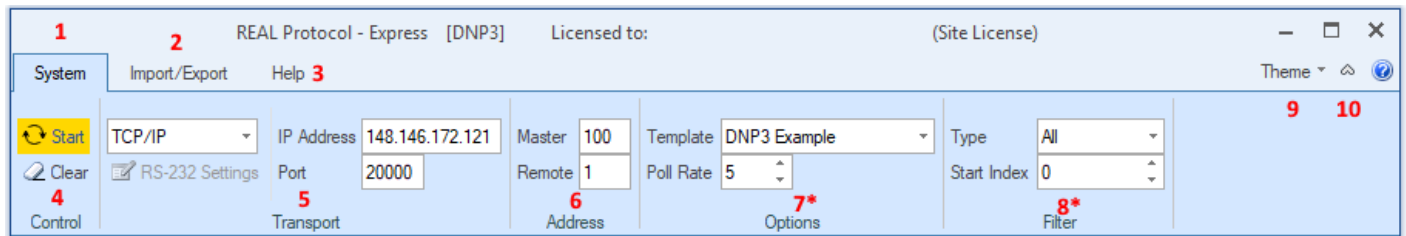


Figure 3 (DNP3)

System Ribbon Bar

- 1. System**
 - Main system ribbon bar
- 2. Import/Export**
 - Protocol packet import and export ribbon bar
- 3. Help**
 - Help ribbon bar
- 4. Control**
 - Start and stop polling
 - Clear Protocol data from tabs
- 5. Transport**
 - Set transport options: TCP/IP or RS-232 communication ports
 - Set RS-232 settings
 - Set IP address and port
- 6. Address**
 - Set master and remote addresses (Modbus only uses a remote address)
- 7. Options (Varies by protocol)**
 - Select optional template to translate protocol data to a human readable form
 - Set protocol device poll rate in seconds
 - Modbus option: 32-Bit Big-Endian or 32-Bit Little-Endian (changes 32-bit values based on how the data is stored in device)
- 8. Filter (Varies by protocol)**
 - Select a filter to display all or part of protocol data returned from device
 - Set start index of protocol data to be displayed
- 9. Theme**
- 10. Minimize or Maximize Ribbon Bar**

Figure 4 shows the help ribbon bar with the menu items for managing the license key and additional system information.

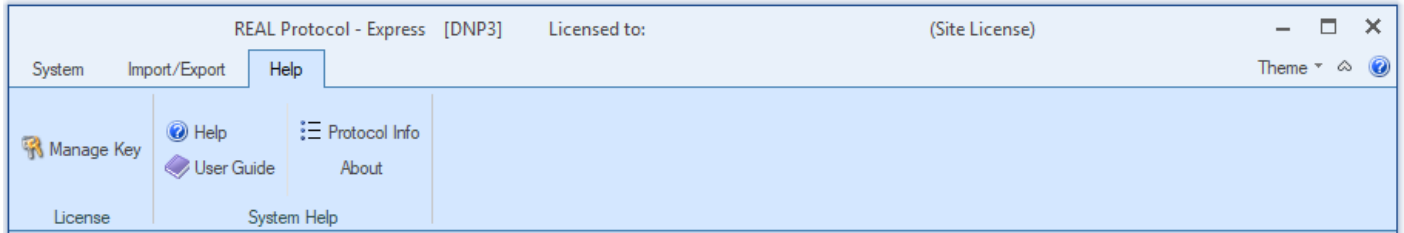


Figure 4

Help Ribbon Bar

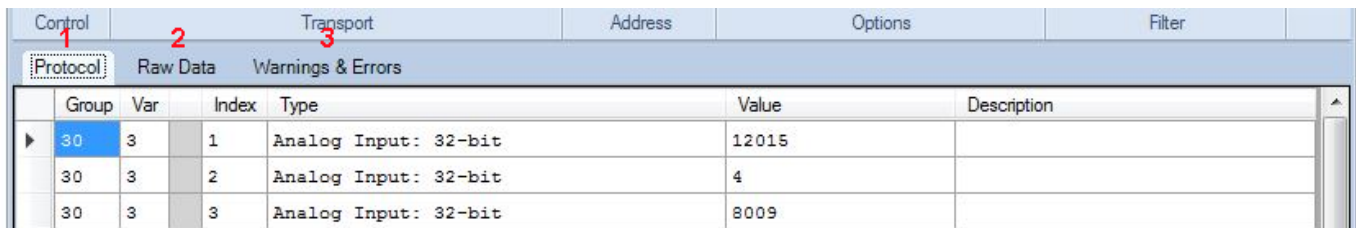
- **Manage Key** - Install/update or uninstall license key
- **Help** - Contact information
- **User Guide** - This document
- **Protocol Info** – Protocol specific information
- **About** - Version and copyright information for the application

NOTE: The “About” box contains a button called “Check Hash.” This button can be used to verify that the application has not been modified by a third-party after installation. If required we can send an email containing the current hash value for the application and you can check this value against the value calculated by the application.

The SHA-512 hash is used to calculate a unique digital signature for the application.

PROTOCOL DATA

Protocol data returned from the remote device is displayed in several tabbed windows. Each window displays the returned data in a slightly different way. The tabs for each window are shown and described below:



Group	Var	Index	Type	Value	Description
30	3	1	Analog Input: 32-bit	12015	
30	3	2	Analog Input: 32-bit	4	
30	3	3	Analog Input: 32-bit	8009	

Figure 5

Protocol Data Windows

1. **Protocol Tab** - Formatted protocol data
2. **Raw Data Tab** - Raw unformatted protocol bytes
3. **Warnings & Errors Tab** - Warning and error messages embedded in the protocol message (DNP3 protocol)

NOTE: The first protocol tab window will display the device data in a human readable form and the returned data is displayed in the exact order sent.

USING THE PROGRAM

The REAL Protocol - Express user interface is greatly simplified compared to other protocol polling applications. Every control and option is displayed on the ribbon bar at the top of the window. Most protocol settings will be the same for each protocol.

DNP3 Protocol

The system sends a class 0 scan to read all current data from the remote device. This request will return all static data points assigned to the device.

Modbus Protocol

The system sends a function 01, 02, 03 or 04 to retrieve 1-bit or 16-bit registers from the device. Since each vendor may implement different 16-bit or multiple 16-bit registers to represent 32-bit values, the protocol tab will display multiple variations of possible register values.

The 32-Bit Big-Endian or 32-Bit Little-Endian option will change the byte order of how 32-bit values are displayed. The system default value is “32-Bit Big-Endian” byte ordering.

A device can be polled by selecting the type of transport (RS-232 port or TCP/IP address) and then selecting the remote device’s address. Once the transport and remote addresses are selected click the “Start” button to begin polling the device.

NOTE: *If an error is encountered trying to connect to a device, a message will be displayed on the status bar at the bottom of the window as shown below.*

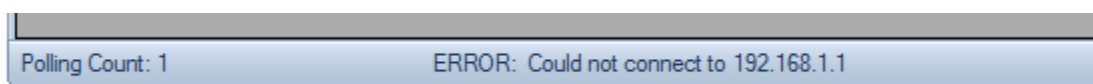


Figure 6

Once the polling starts and the data is flowing back to the application, you can click on the tabs in the protocol data window to see the data in more detail. Also notice that while polling a device many of the menu options are not available. The disabled menu options can be re-enabled once polling stops.

FILTERS

The “Filter” menu at the top of the screen can be used to display some or all of the returned protocol data. Using the “Filter” will help keep important data from scrolling out of view when viewing large protocol maps.

The DNP3 filter provides the following view types:

1. **All:** Display all protocol data
2. **All, No Binary:** Do not display binary data
3. **All, No Binary or Frozen:** Do not display binary or frozen data
4. **Analog:** Display analog data
5. **Analog – Poll:** Send a special analog poll command to a device that doesn’t respond to the normal “Class 0” poll.
6. **Binary:** Display binary data
7. **Counter:** Display counter data
8. **Start Index:** The filter can also display data starting from any index. Normally the index starts at zero (0) but can be set to start from any value to keep data at the end of a long protocol map from scrolling off the screen.

The Modbus filter provides the following view types:

1. **Function** – 01, 02, 03 or 04 (RTU or TCP formats)
2. **Start Register** – Modbus registers start at 1
3. **Total Registers** – Number of registers to display from start register

NOTE: *The filter can be changed at any time while polling a device.*

SETTINGS AND SYSTEMSETTINGS XML FILES

Systems.xml File

The REAL Protocol - Express application uses an XML file in the root directory called “Settings.xml” to store default system settings. An example settings file is shown below:

```
<?xml version="1.0"?>
<SETTINGS>
  <DNP3>
    <RS-232>
      <BaudRate>9600</BaudRate>
      <DataBits>8</DataBits>
      <Parity>None</Parity>
      <StopBits>1</StopBits>
      <FlowControl>None</FlowControl>
    </RS-232>

    <TCPIP>
      <IPAddress>192.168.1.1</IPAddress>
      <Port>20000</Port>
    </TCPIP>

    <OPTIONS>
      <MasterAddress>100</MasterAddress>
      <RemoteAddress>1</RemoteAddress>
      <PollRate>5</PollRate>
      <Template>None</Template>
      <Filter>All</Filter>
    </OPTIONS>
  </DNP3>
  <MODBUS>
    <RS-232>
      <BaudRate>9600</BaudRate>
      <DataBits>8</DataBits>
      <Parity>None</Parity>
      <StopBits>1</StopBits>
      <FlowControl>None</FlowControl>
    </RS-232>

    <TCPIP>
      <IPAddress>192.168.1.1</IPAddress>
      <Port>502</Port>
    </TCPIP>

    <OPTIONS>
      <RemoteAddress>1</RemoteAddress>
      <PollRate>5</PollRate>
      <Template>None</Template>
      <Table>04 TCP</Table>
      <StartRegister>1</StartRegister>
    </OPTIONS>
  </MODBUS>
</SETTINGS>
```

NOTE: The “Settings.xml” file can be edited to change the default settings used by the application when it starts.

SystemSettings.xml File

The SystemSettings.xml file contains the name of the current theme for the application.

TEMPLATE XML FILES

Viewing a large protocol map can be difficult and error-prone, so the template feature can help by placing a human readable description next to the data points. The template files are created by the end user and are placed in the templates directory.

When the application starts, the application loads the names of all templates based on the protocol type. DNP3 templates are stored in the “DNP3 Templates” directory and the Modbus templates are stored in the “Modbus Templates” directory.

Once a template is selected, the system will fill the “Description” column with the user created names for the different data points.

DNP3 Template Example

An example template file is shown below:

```
<?xml version="1.0"?>
<Groups>
  <Group>
    <Index>20, 6, 0, VAR_Delivered</Index>
    <Index>20, 6, 1, VAR_Received</Index>
  </Group>
  <Group>
    <Index>30, 3, 0, Phase1_Volts</Index>
    <Index>30, 3, 1, Phase2_Volts</Index>
    <Index>30, 3, 2, Phase3_Volts</Index>
  </Group>
</Groups>
```

The DNP3 template file consists of a series of groups (one or more) that define the protocol maps for the returned values. Let's look at the example template file above.

The example template file consists of two protocol groups:

1. **Group 20, Variance 6** – Counter 16-bit
2. **Group 30, Variance 3** – Analog Input 32-bit

The next two values in the template file after the group and variance values are the index number and the description. The index is the number that represents the unique row from a particular protocol group. Normal index values start at 0 and increment by one for each row. The description is a user defined value for a data point.

Modbus Template Example

An example template file is shown below:

```
<?xml version="1.0"?>
<Functions>
  <Function>
    <Index>3, 1, Phase1_Volts</Index>
    <Index>3, 2, Phase3_Volts</Index>
    <Index>3, 3, Phase3_Volts</Index>
  </Function>
  <Function>
    <Index>4, 1, VAR_Delivered</Index>
    <Index>4, 2, VAR_Received</Index>
  </Function>
</Functions>
```

The Modbus template file consists of a series of functions (one or more) that define the protocol maps for the returned values. Let's look at the example template file above.

The example template file consists of two protocol functions:

1. **Function 03** – 16-bit
2. **Function 04** – 16-bit

The next two values in the template file after the function number are the index number and the description. The index is the number that represents the unique row from a particular protocol function. Index values start at 1 and increment by one for each row. The description is a user defined value for a data point.

NOTE: The DNP3 protocol group and variance values are always displayed in the first two columns on the “Protocol” tab window when a device is being polled. Use these values when creating a template file.

The Modbus function value is always displayed in the first column of the “Protocol” tab window when a device is being polled. Use this value when creating a template file.

If you need help building a template please send us an email message describing the protocol map and we will help you build the template.

APPLICATION LICENSING

The REAL Protocol - Express application initially runs as an evaluation version when first installed. After a license is purchased an email will be sent with the license key.

Once the license key is entered into the system a “License.lic” file is created in the root directory. This license key is valid for the version of the application that you purchased.

If a license needs to be moved to a new computer or laptop, then uninstall the license key from the current computer and install this same license on a new computer. The steps for moving and updating a license key are shown below:

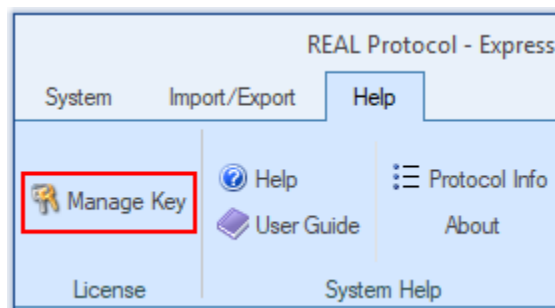


Figure 7

Moving a License Key

1. Go to Help on the ribbon bar
2. Click “Manage Key” to open a license dialog box
3. Copy the key and save the license for use on the new computer
4. Click the “Uninstall License” button to remove the license

Updating or Installing a License Key

1. Go to Help on the ribbon bar
2. Click “Manage Key” to open a license dialog box
3. Enter license key and click the “OK” button to create the license file
4. Restart the application to load the new license key

DEBUGGING PROTOCOL MAPS

The REAL Protocol - Express application displays a subset of the possible values that can be stored in a protocol map. If you have a device where the data can't be displayed, the application can capture the packet and that packet can be sent to us for analysis.

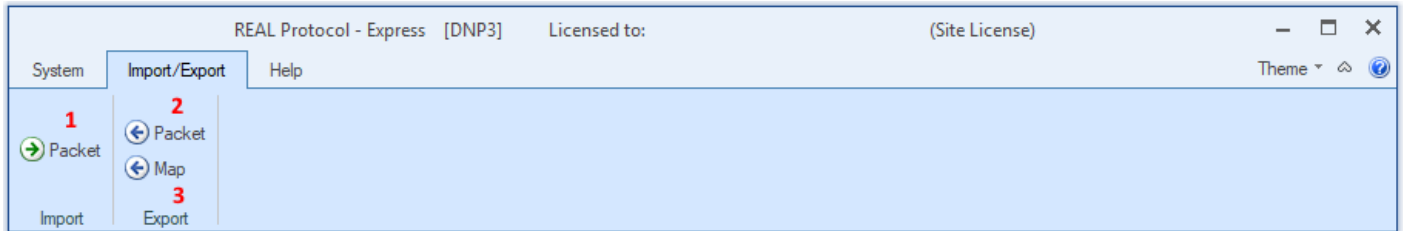


Figure 8

On the “Import/Export” ribbon bar the following options are available:

1. **Import Packet** - Clicking this button opens a window where a “Hex” packet can be imported and parsed by the system. This option is used by internal software developers to add new protocol features to the system.
2. **Export Packet** - Clicking this button will display the last packet received from the device. The data in the packet can be copied to an email message and sent to us for analysis.
3. **Export Map** - Clicking this button opens a window displaying the last formatted packet from the device. This formatted packet can be copied from the window and saved to a text file or sent in an email message.