
Banner PresencePLUS Sensor

Application Note

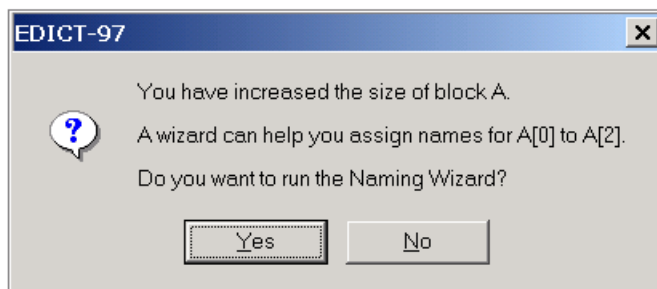
This document describes how to configure a Paradigm operator interface terminal to allow communications with a Banner PresencePLUS Sensor. The communications protocol allows the Paradigm HMI to store and download sensor inspections. Please read this document carefully before attempting to configure communications with these devices.

Overview

The driver described here allows the Banner PresencePLUS Sensor inspections to be loaded directly into the HMI, and downloaded from the HMI to the sensor. Hundreds of unique inspections can be stored in the HMI. A utility program, PPLoad.exe, is used to load all of the desired inspections directly into a comms block in the HMI. The desired inspection can then be selected, and downloaded to the sensor.

Instructions

1. Configure the sensor using Banner's PresencePLUS2 software package, and save the inspection files in an easily located folder on your PC, such as Banner's default folder C:\Program Files\Banner Engineering\PresencePLUS2\PPVSData Folder\. Repeat this for each unique inspection that will be stored in the HMI, saving each inspection under a unique file name in this folder.
2. Set up the comms blocks in Edict-97. One comms block will be used to store all of the inspections, and one comms block will be used to write the selected inspection to the sensor. NOTE: the actual sensor inspections will be loaded into the comms block after the Edict-97 database is downloaded to the HMI.
 - A. Inspection storage. In the example that follows, 3 inspections are stored in comms block A. Set the comms block A **Data Type** to "String[Size 128]", **Size** to "3", **Access** to "Both", and **Update** to "Manual". Use the Naming Wizard to name the comms blocks the same as the file names chosen in step 1 (this is important for "PPLoad.exe Utility Program" step 2C). For the example, comms block A[0] should be named "pass", A[1] should be named "pattern", and A[2] should be named "blank". After making these selections, the Device will show "Internal".



- B. Inspection write. Set the comms block B **Device** to the sensor's name, enter "SC00" for the **address**, set the **Size** to "1", and **Access** to "Write". Use the Naming Wizard to name comms block B[0] "config". NOTE: it is very important that the Access is set to "Write" only.
- C. Inspection read. Set the comms block C **Device** to the sensor's name, enter "SC00" for the **address**, set the **Size** to "1", **Access** to "Read", and **Update** to "Manual". Use the Naming Wizard to name comms block C[0] "readConfig". NOTE: it is very important that the Access is set to "Read" only.

- D. Download confirmation (v2.05+). Set the comms block D **Device** to the sensor's name, enter "DC" for the **address**, set the **Size** to "1", **Access** to "Both", and **Update** to "Auto".
- E. Upload confirmation (v2.05+). Set the comms block E **Device** to the sensor's name, enter "UC" for the **address**, set the **Size** to "1", **Access** to "Both", and **Update** to "Auto".

The screenshot shows the EDICT-97 - BANNER2.EDB application window. The 'Communications Blocks' table is displayed with the following data:

	Device	Address	Data Type	Size	Access	Update	Links	Enable
A	Internal	None	String [Size 128]	3	Both	Manual	None	Default
B	Dev01	SC00	String [Size 128]	1	Write	Auto	None	Default
C	Dev01	SC00	String [Size 128]	1	Read	Manual	None	Default
D	Dev01	DC	Individual Bytes	1	Both	Auto	None	Default
E	Dev01	UC	Individual Bytes	1	Both	Auto	None	Default
F	None	None	Individual Bits	0	Read	Auto	None	Default

Comms blocks A thru E are used for the example, but any available comms block could be used for any function. Due to Edict's nature of reading data "ahead", as opposed to "on demand", the inspections cannot be direct referenced. Rather, the Comms Block tables must be used.

3. Finish the Edict-97 database. To download an inspection to the sensor, set the Write comms block equal to the desired inspection. For the example, "config := pattern" would download the pattern.inp inspection to the sensor (this statement performs the same function as "B[0] := A[1]"). To upload an inspection to the sensor, implement an Event Action to call the method ReadBlock then set the data to the desired inspection storage area. For the example, to upload the sensor configuration and store to pattern, the Event Action should be "ReadBlock(C), pattern := readConfig" (this statement performs the same function as "ReadBlock(C), A[1] := C[0]"). When an inspection is downloaded/uploaded from the HMI to the sensor, the sensor is taken out of "run mode", the inspection is written to or read from the sensor, then the sensor is placed into "run mode". The sensor will perform a soft boot, which may take up to 20 seconds.

Inspection download/upload confirmations have been provided in driver revisions v2.03+. These confirmations verify a successful operation while also providing status and failure information. While confirmations can be set to any value from 0 to 255, meaningful value definitions are as follows:

Value	Definition
0 (0x00)	Operation in progress
1 (0x01)	Error Code 1
2 (0x02)	Error Code 2
3 (0x03)	Error Code 3
4 (0x04)	Error Code 4
253 (0xFD)	Operation Failed
254 (0xFE)	Operation Completed Successfully

Banner PresencePLUS Sensor

Decode text can be used to display the desired text based on the confirmation. By default, on power up of the terminal, confirmations are initialized to zero (Operation in progress). These confirmations must be initialized to any desired value (not listed in the table above) by using the "System initialized" global event.

Please note, when using an upload or download confirmation in a database, all uploads and downloads must be performed as shown below. When an error code or failure occurs, no retry is issued and the appropriate code is returned in the confirmation.

Inspection download/upload confirmations must be set to zero before performing an upload or download operation. In the following example, Event 1 sets the inspection download confirmation to zero, then performs the inspection download operation, while Event 2 sets the inspection upload confirmation to zero, then performs the inspection upload operation.

Event	Enable	Action	Route
1 Soft-key 1 pressed	Default	D[0]:=0, B[0]:=A[1]	Defat
2 Soft-key 3 pressed	Default	E[0]:=0, ReadBlock(C), A[1]:=C[0]	Defat
3 None	Default	None	Defat
4 None	Default	None	Defat

4. Individual parameters may also be accessed. Five parameter blocks are available.

Block	Address	Parameter	Access
Pixel Count	PC00	Current Pixel Count	Read Only
Percentage Match	PM00	Upper Match %	Read/Write
	PM01	Lower Match %	Read/Write
	PM02	Pixel Count (based on Match	Read/Write
	PM03	Pass Count (High)	Read Only
	PM04	Pass Count (Low)	Read Only
Pixel Count Statistics	PS00	Max Pixel Count	Read/Reset*
	PS01	Min Pixel Count	Read/Reset*
	PS02	Total Targets Inspected	Read/Reset*
	PS03	Average Target Pixel Count	Read/Reset*
Pass/Fail Counts** (v2.05+)	PF00	Pass Count	Read/Reset*
	PF01	Fail Count	Read/Reset*
Trigger Image Capture Mechanism (v2.05+)	TI00	Trigger : Any change in value will activate trigger	Write Only
Sensor Status (v2.06+)	SS00	Sensor Status: 0 = No Error, 1 = ERROR	Read/Clear Error
Run Mode (v2.06+)	RM00	Start/Stop Run Mode: 0 = Stop, 1 = Start	Read/Write

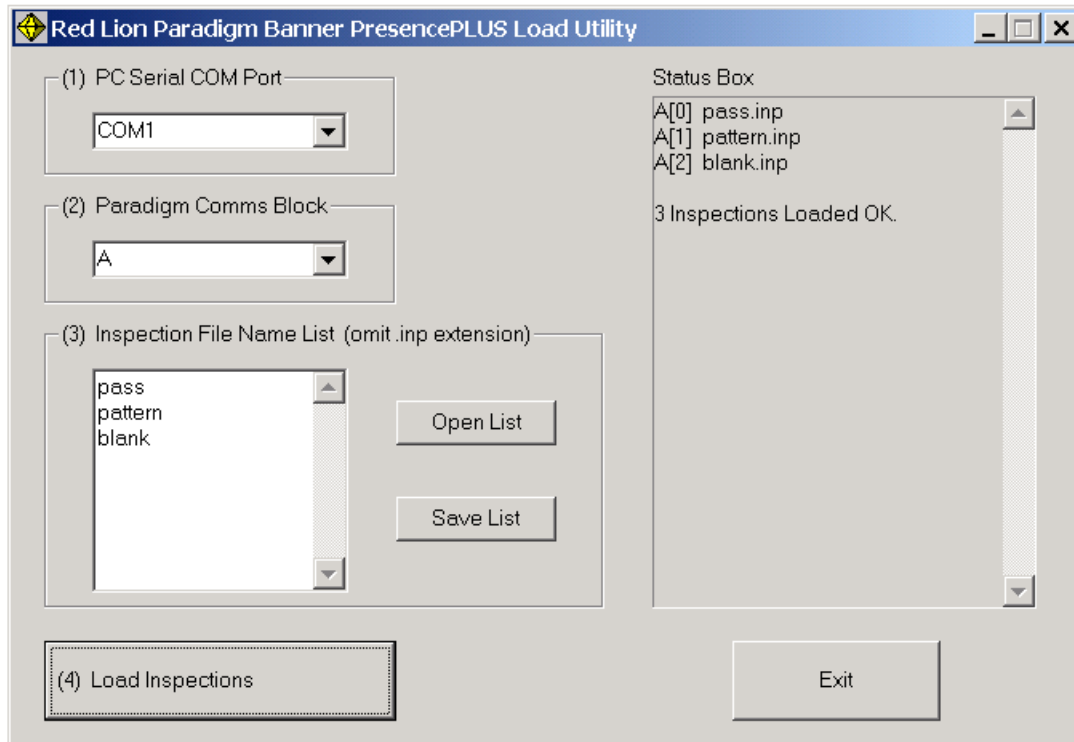
Banner PresencePLUS Sensor

*Parameter writes of a value of 0 will result in all parameters in the block to be reset to a value of 0. Parameter writes of a non-zero value will result in no action.

**Current value is retained in Red Lion HMI upon power down/up of sensor. Subsequent readings will be accumulated. A reset will set all pass/fail values in the Red Lion HMI and the Banner PresencePLUS sensor to 0.

5. After the Edict database is downloaded to the HMI, use the PPLoad.exe utility program to load the inspections into the HMI "storage" comms block. This step can be repeated at any time, to update any of the inspections. NOTE: If the Edict-97 database is subsequently changed, the inspections should be re-loaded into the comms blocks, UNLESS the following are ALL true: A. An incremental download was request (ie. F9 or the toolbar button). B. Incremental download was possible (ie. it didn't do a full download anyway.) C. No changes were made to the "storage" comms block properties.

PPLoad.exe Utility Program



PPLoad

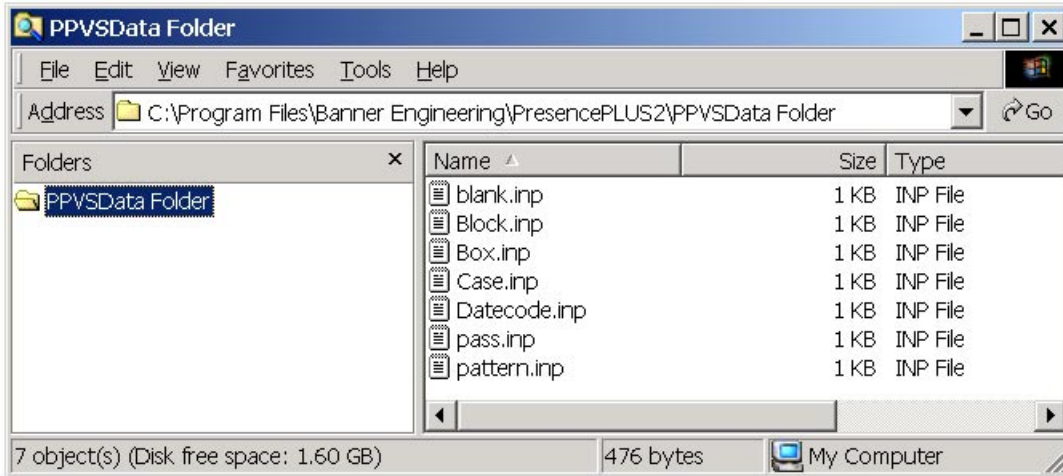
1. Install the PPLoad.exe utility program on your PC.
 - A. From the PPLoad.ZIP file, extract PPLoad.exe to the folder on your PC, where the inspections are stored (such as Banner's default folder C:\Program Files\Banner Engineering\PresencePLUS2\PPVSData Folder\ from "Instructions" Step 1). Note: the PPLoad.exe file MUST be in the same folder as Banner's *.inp inspection files.
 - B. Create a shortcut to the PPLoad.exe program, and place the shortcut where you can easily locate it, such as on your desktop, or in your Start Menu.
2. Run the PPLoad.exe program shortcut. NOTE: the HMI's programming port must be connected to the PC's COM port (via programming cable P890301Z).
 - A. Select the PC Serial COM Port. "COM1" for the example.
 - B. Select the Paradigm Comms Block that will store the inspections. "A" for the example.
 - C. The Inspection File Name List text box should contain a list of the inspection files that are to be loaded into the HMI. This list should contain one inspection file name per line, and should not contain the ".inp" file name extension, any comments, or extra characters. If you've previously saved a list of file names, open the list, and then go to Step D. For new applications, the easiest and most accurate way to load this list, is to copy it from Edict-97 Named Data (this assumes that you used the Naming Wizard in "Instructions" Step 2A):
 1. In Edict-97, open your database, then jump to Named Data.
 2. Select all of the inspection names, then right-click, and Copy.

	Name	Maps To	Data Type
1	pass	A[0]	String [Size 256]
2	pattern	A[1]	String [Size 256]
3	blank	A[2]	String [Size 256]
4	config	B[0]	String [Size 256]
5	None	Internal	16-bit Signed
6	None	Internal	16-bit Signed

3. Go back to the PPLoad utility, right-click in the File Name List box, and Paste the file names.
 4. Save the file name list for future use by clicking "Save List".
- D. Click "Load Inspections". The Status Box will show the comms block, and file name for each inspection loaded into the HMI, and will report the number of inspections that were loaded OK. Typical PPLoad faults reported in the Status Box:
- "Error reading <filename>.inp". Verify the file name and location (the inspection files must be in the same folder as the PPLoad.exe utility, and should not contain any extra characters or descriptors).
 - "Error Writing <filename>.inp". Verify that the HMI programming port is connected to the proper PC Serial COM Port (via P890301Z programming cable).

A Graphical Example

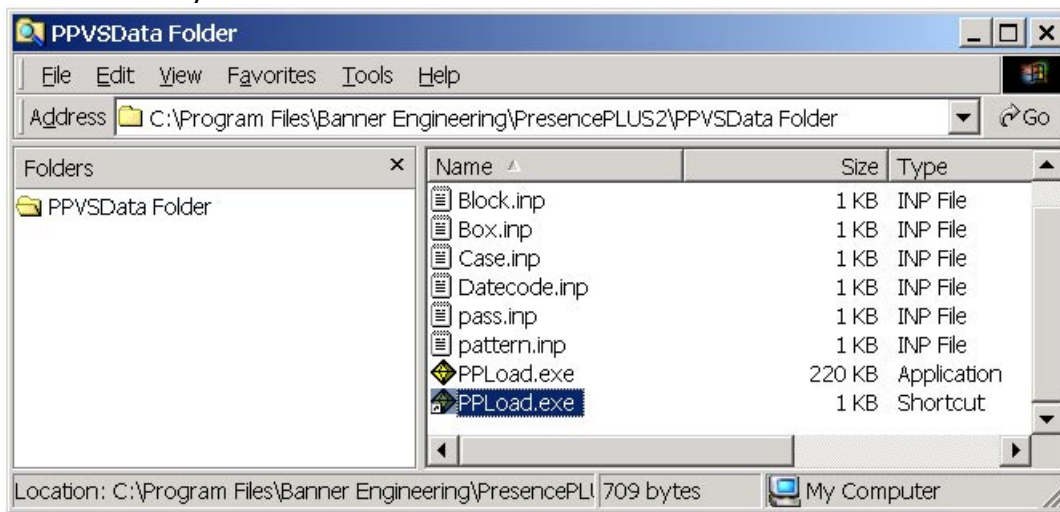
1. Configure each of the inspections using Banner's PresencePLUS2 software package, and save the inspection files in Banner's default folder. For this example, the inspections will be Block.inp, Box.inp, Case.inp, and Datecode.inp.



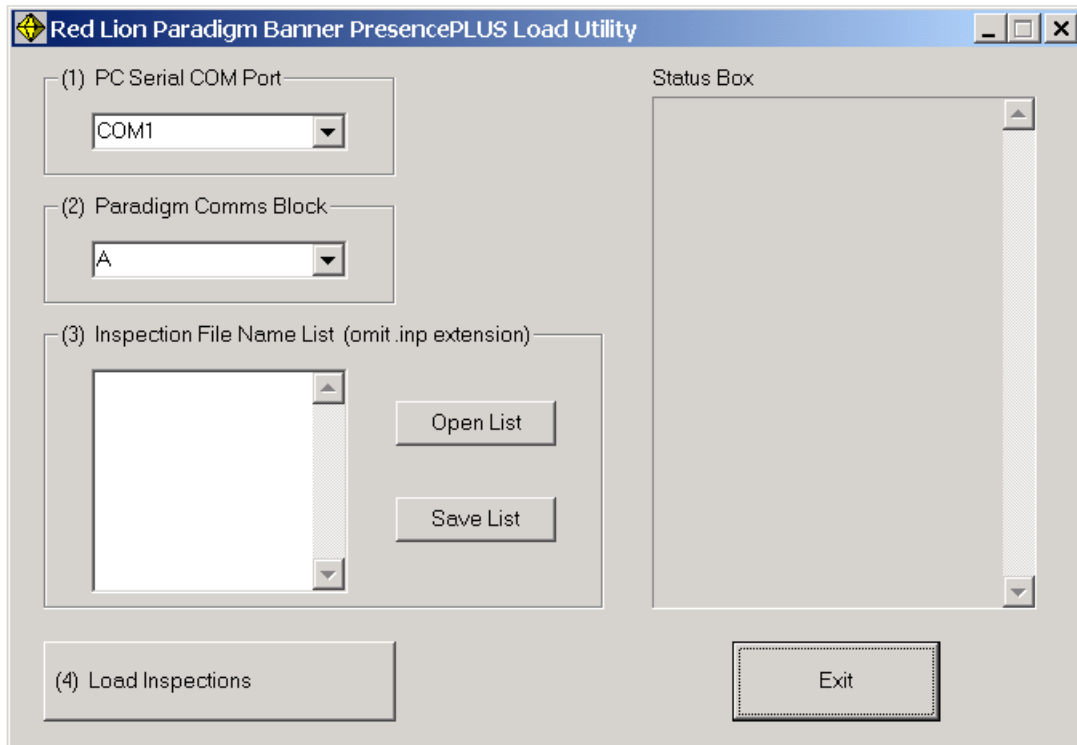
2. Set up the comms blocks in Edict-97. Comms block A will store the 4 inspections, and comms block B will write the selected inspection to the sensor. Use the Naming Wizard to name A[0] Block, A[1] Box, A[2] Case, A[3] Datecode, and B[0] Sensor.

	Device	Address	Data Type	Size	Access	Update
A	Internal	None	String [Size 128]	4	Both	Manual
B	P02	SC00	String [Size 128]	1	Write	Auto
C	None	None	16-bit Signed	0	Read	Auto

3. Finish the Edict-97 database, and download it to the HMI. Leave Edict-97 open for Step 6.
4. Install the PPLoad Program, create a shortcut to it, and then move the shortcut to a location that will be easy to locate.



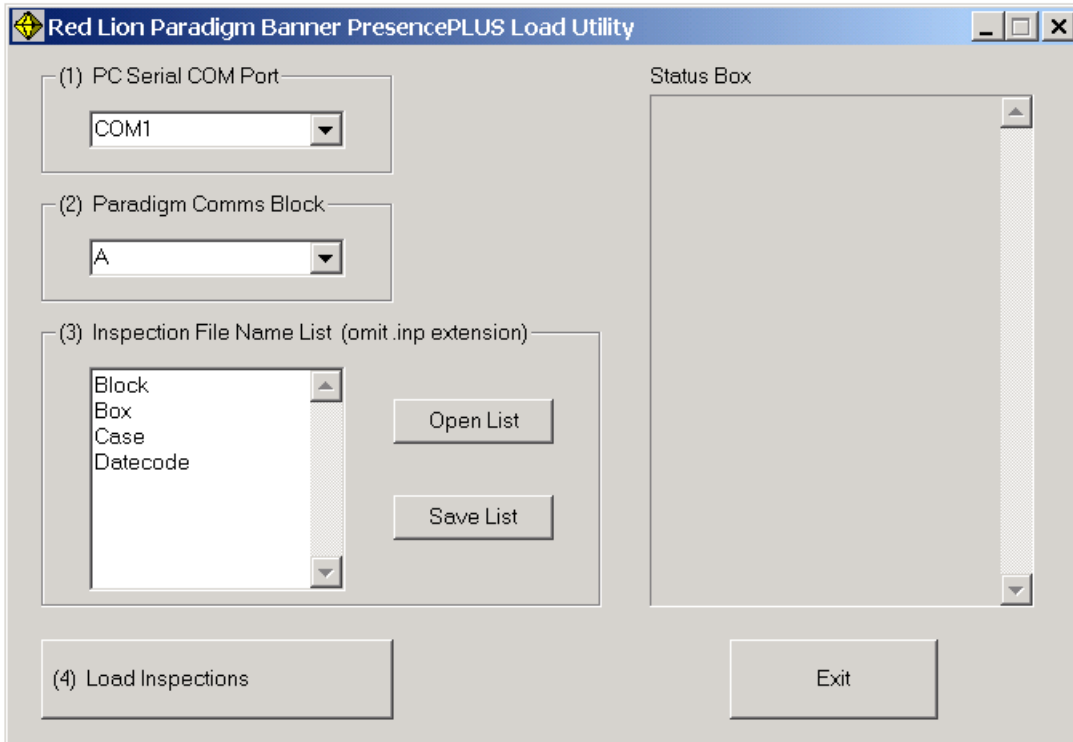
5. Run the PPLoad Program. Select the PC COM port, and the comms block that will store the inspections.



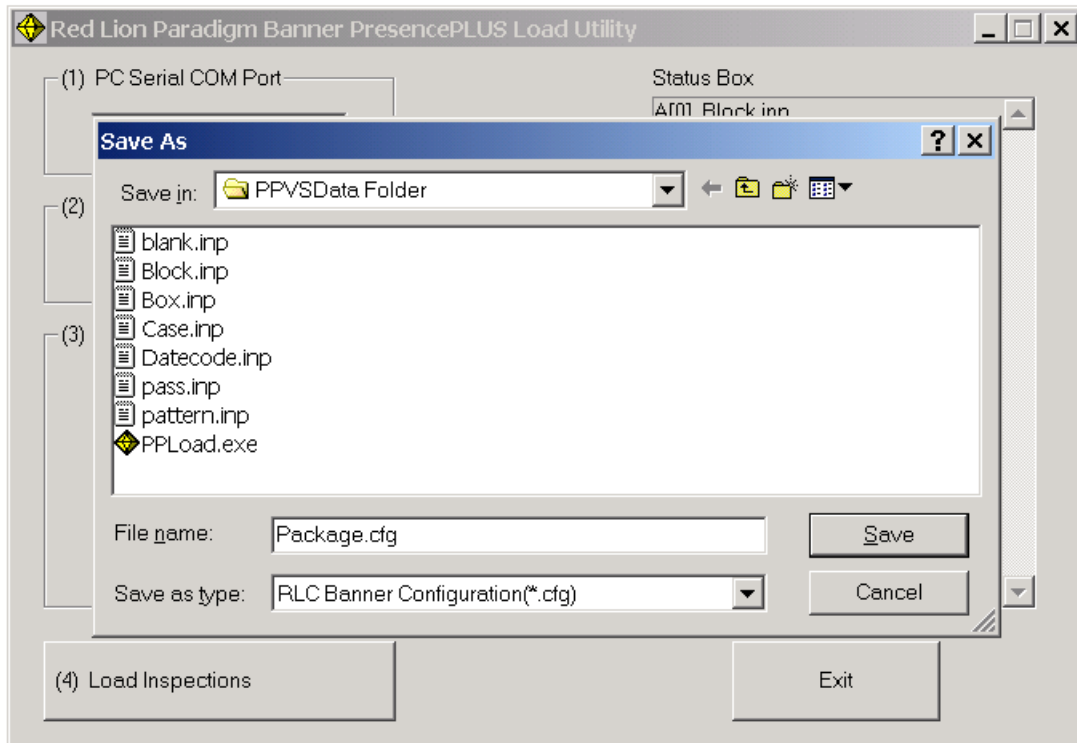
6. In your Edict-97 database, jump to Named Data. Select all of the inspection names, then right-click, and Copy.

	Name	Maps To	Data Type	Transform
1	Block	A [0]	String [Size 256]	None
2	Box	A [1]	String [Size 256]	None
3	Case	A [2]	String [Size 256]	None
4	Datecode		String [Size 256]	None
5	Sensor		String [Size 256]	None
6	None	nal	16-bit Signed	None
7	None	nal	16-bit Signed	None

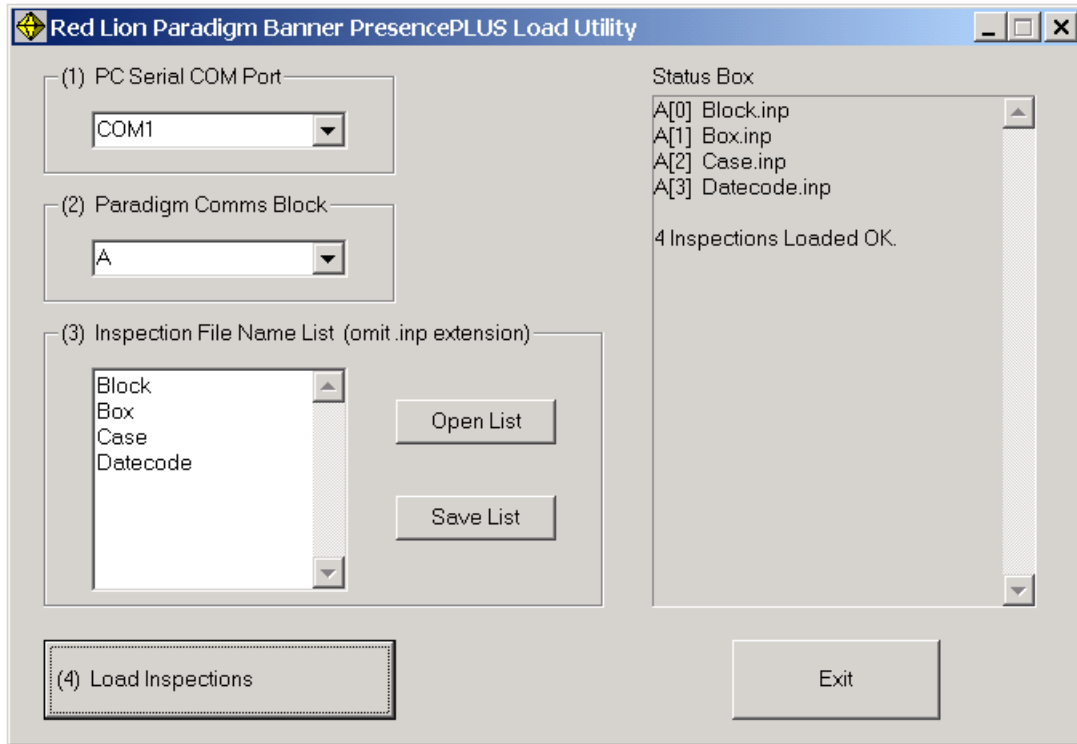
7. Right-click in the File Name List box, and Paste the file names.



8. Click "Save List" for future use.



- Click "Load Inspections". The 4 inspection files will be loaded from the PC into the comms blocks of the HMI.



Communications

Communications with the Banner PresencePLUS Sensor is via an RS-232 point to point link. Default serial communications format is 115,200 baud rate, 8 data bits, No parity, and 2 stop bits. Edict-97's Comms Port, Data Format will show the baud rate as 19,200, but is actually 115,200 when the Banner PresencePLUS Sensor is selected. The connection details are described in the tables below.

Via PresencePLUS PC cable (RLC Cable # P895047Z)

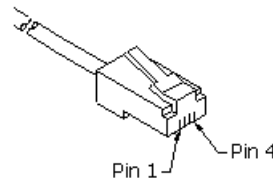
Paradigm RS232 Port	Banner PresencePLUS PC Cable, DB-9
Pin 1 (Tx)	Pin 3 (Rx)
Pin 2 (Rx)	Pin 2 (Tx)
Pin 3 (RTS)	
Pin 4 (CTS)	
Pin 5 (Comm.)	Pin 5 (Comm.)

In addition a link must be fitted between Pin 3 (RTS) and Pin 4 (CTS) on the Paradigm.

 Direct connection to PresencePLUS (RLC Cable # P895064Z)

Paradigm RS232 Port	Banner PresencePLUS RS232 Port, RJ-11
Pin 1 (Tx)	Pin 3 (Rx)
Pin 2 (Rx)	Pin 2 (Tx)
Pin 3 (RTS)	
Pin 4 (CTS)	
Pin 5 (Comm.)	Pin 4 (Comm.)

In addition a link must be fitted between Pin 3 (RTS) and Pin 4 (CTS) on the Paradigm.



Knowledge of Unit Operation Is Assumed

In all cases, the simple principle of 'pass-through' is maintained: there is no attempt to validate an inspection in terms of the end use of the unit: both familiarity with the PresencePLUS sensor and knowledge of system operation are assumed.

Ordering Information

MODEL NO.	DESCRIPTION	PART NO.
CL01	LCD, 2 X 20, 3 Soft keys, 128 K memory, 115.2K Baudrate Capable	CL010050
	LCD, 2 X 20, 3 Soft keys, 512 K memory, 115.2K Baudrate Capable	CL010060
	EDICT-97 Development Kit. Includes Software, Manual and DB-9 to RJ-11 HMI Programming Cable	SFEDT
	Battery Replacement	BAL3R004