

Exhibit G: User Manual

FCC ID: QYT-4120

Installation Notes and MSSAssist Users Guide

IDmicro 4120 Interrogator

(Draft version - 10June2003)

1.0 INSTALLATION NOTES

In accordance with FCC regulations the following information is provided to all users:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

All installations of the IDmicro 4120 must be accomplished by professional installers who are knowledgeable of the FCC rules and regulations. When designing the transmit antenna placement, particular emphasis on location and stand offs must be accorded to ensure people can not be exposed to levels above the permissible limits. The IDmicro 4120 is used with 8 dB linearly polarized antennas (from Arc Wireless) or 4 dB circularly polarized antennas (from MaCom). These antennas have been approved for use by the FCC. No other antennas may be used without written approval.

IDmicro warrants that with the approved antennas, the output power of the 4120 will not exceed the FCC permissible levels. Use of additional amplifiers or higher gain antennas is not permitted. Deviation from or usage of non-IDmicro supplied parts may void the users authority to use the equipment.

2.0 MsAssist Reference Guide



This section is an overview of the MicroStamp Assist (MsAssist.exe) application.¹ The MsAssist.exe program is a development tool created for engineers to experiment and gain familiarization with the RFID system.

The MsAssist program's usefulness is enhanced with features such as IDmicro-supplied software libraries, on-line bit references, calculators for determining register values, tools for manipulating command windows, and the ability to record and play back command sequences.

0.1 MsAssist Navigation

The following section describes important navigation techniques for the MsAssist.exe program.

0.1.1 Displaying MsAssist Windows

The MsAssist.exe software application was developed using Visual Basic (VB) 6.0 (32-bit) and is limited to any constraints placed on VB.

0.1.2 Navigating With Hot Keys

If the **MsAssist** main window is the active window, any of the drop-down menus may be selected by using the <ALT> key plus the underlined character from the menu choice. For example, to select the **Tools** drop-down menu, press the key sequence <ALT><O>.

If a command window is the active window, the hot keys (instead of the mouse) may be used to select any of the buttons. These hot keys are not case sensitive.

Send	<ALT><S>
Minimize	<ALT><M>
Close	<ALT><C>

0.1.3 Minimizing a Group Box

A few of the tag-specific functions, such as **SetDataLogger**, have multiple parameters which require a 1024x768 display resolution. However, a minimizing feature, available on all command windows containing multiple group boxes, allows MsAssist to run on monitors using a 640x480 resolution.

To minimize a group box, place the mouse pointer on the group box caption to be minimized and click the right mouse button. Repeat the process to restore the group box to its original size.

1. IDmicro also provides a Readme.txt file (included with the development system software) that documents changes and updates to the MSL or the *MsAssist Reference Guide*.

0.2 Command List

The command list is an internal structure in the MsAssist program that is used to track command windows. A command window is defined as any window selectable from **Tag Cmnds**, **Interr Cmnds**, or **Conven Cmnds** drop-down menus. Tools that position and manipulate the command windows such as **Cascade Cmnds** and **Reposition Cmnds**, as well as the **Macro Play/Record** use the command list to accomplish this.

The command list maintains the command window's sizing information which makes it possible to minimize and maximize the group boxes in the command windows.

When the **Macro Play/Record** is used to play the commands, the command list executes beginning with the first command in the list. The **Load** button from the **Macro Play/Record** window uses the command list to destroy existing command windows. Command list information, stored at the end of the macro file, is used to build the new command list.

0.3 MsAssist Window Layout

The following section describes the layout of command windows in the MsAssist.exe program.

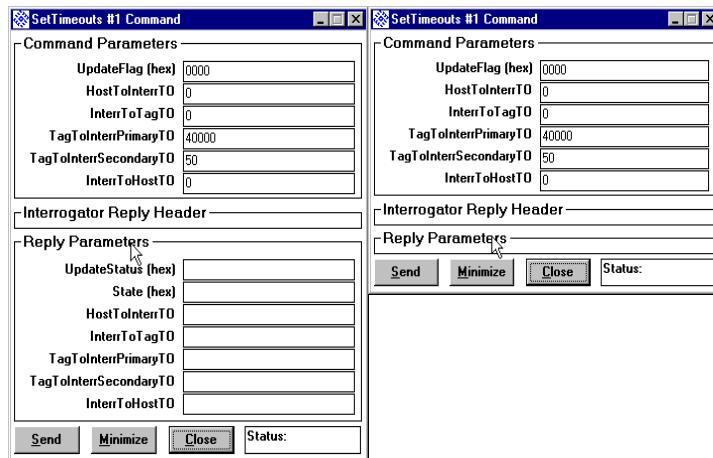


Figure 1 - Command Window Example

0.3.1 Command Window Layout

Tag Cmnds windows contain a **Command Parameters** group box, an **Interrogator Reply Header** group box, a **Reply Parameters** group box, a **Send** button, a **Minimize** button, a **Close** button, and a **Status** box.

Interr Cmnds windows contain the **Send**, **Minimize**, and **Close** buttons, a **Status** box, and possibly a **Command Parameters** group box and/or a **Reply Parameters** group box.

Conven Cmnds windows vary depending on command functionality. At a minimum however, they contain **Send**, **Minimize**, and **Close** buttons.

0.3.1.1 Command Parameters Group Box

The **Command Parameters** group box varies depending on the function. Some functions found in the **Interr Cmnds** and **Conven Cmnds** menus do not have a **Command Parameters** group box. Users may modify all fields in the **Command Parameters** group box.

When fields in the **Command Parameters** group box require a hexadecimal (hex) value, only valid hex characters are permitted. A hex field defined as UBYTE will not allow more than two valid hex characters to be entered.

When fields in the **Command Parameters** group box require decimal (dec) values, only digits 0 through 9 are accepted. A decimal field defined as UBYTE is limited to three valid decimal characters, and a hex field is limited to two valid hex characters. MsAssist does not verify if the UBYTE decimal value is greater than 255.

When the **Send** button is selected, a formatting function ensures that each field in the **Command Parameters** group box is formatted correctly. If a decimal field is empty, a zero is placed in the field. If a hexadecimal field is empty, the field is prefixed with zeros, according to the field's type. For example, the **SubCmnd** field is ULONG, and if "20" is entered, the **SubCmnd** field will be formatted to "00000020". Refer to the Tag SubCmnd Bit Assignments on page 193 for command parameter type definitions.

0.3.1.2 Interrogator Reply Header Group Box

The **Interrogator Reply Header** parameters remain constant for all functions that require it. Some functions found in the **Interr Cmnds** and **Conven Cmnds** menus do not have an **Interrogator Reply Header** group box in their data structure.

Fields in the **Interrogator Reply Header** group box display either a hexadecimal (hex) or decimal (dec) value. The **Interrogator Reply Header** fields cannot be edited. Some interrogators do not use an **Interrogator Reply Header**.

0.3.1.3 Reply Parameters Group Box

The **Reply Parameters** fields vary depending on the function. Some functions found in the **Interr Cmnds** and **Conven Cmnds** menus do not use the **Reply Parameters** group box. Fields in the **Reply Parameters** group box display either a hexadecimal (hex) or decimal (dec) value. The **Reply Parameters** fields cannot be edited, but a reply parameter (such as *TagId*) may be copied and pasted into a command parameter box.

0.3.1.4 Status Box

The **Status** box displays both the *TagStatusReg* and *InterrStatusReg*. The *TagStatusReg* is the most significant two bytes of the status. For **Interr Cmnds** and **Conven Cmnds**, the status is the value returned by the function. The MSL Function Descriptions on page 55 provide detailed information about the reply status for each tag-specific and interrogator-specific function reply.

0.3.1.5 Send Button

A command is executed by selecting the **Send** button located at the bottom left corner of each command window. After **Send** is selected, a check is made to determine if a command is being processed. If so, the new command will not be processed. This prevents the possibility of stack overflows. If a command is not being processed, the command parameters are formatted, all reply parameters are cleared, the command is sent, and the command's reply is displayed. Tag-specific commands return a status that is evaluated and displayed in the **Retries**, **Antenna**, and **Attenuation** boxes.

An **Identify Command** window is shown below. The left image displays the command window before selecting the **Send** button, and the right image displays the results following the send. Notice that the window title lists the command window type and the numeric position in the command list. This method is used to distinguish between command windows.

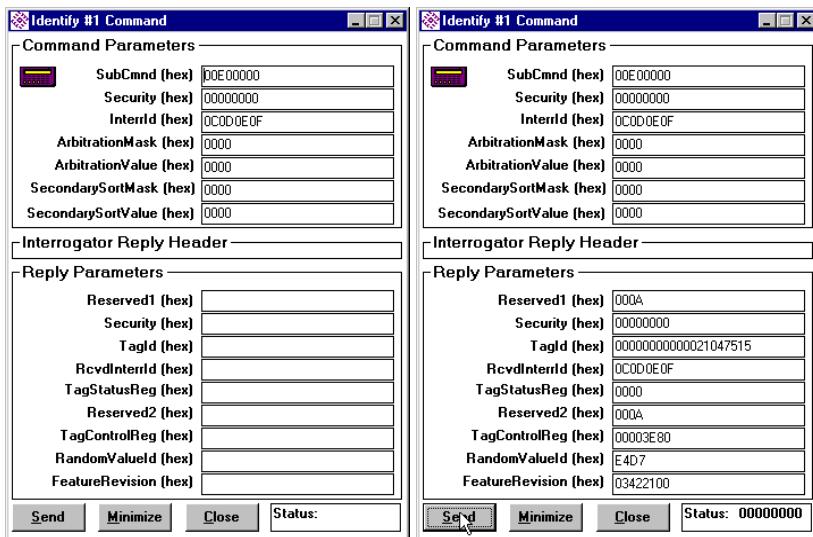


Figure 2 - Command Window Results

0.3.1.6 Minimize Button

Minimize command windows by selecting the **Minimize** button at the bottom of each command window or by selecting the minimize button in the upper right corner of the command window.

0.3.1.7 Close Button

Close command windows by selecting the **Close** button at the bottom of the command window or by selecting **Close** from the control box in the upper left corner of the command window.

0.3.1.8 Calculator Button

Certain MSL bit-specific registers, such as SubCmnd, have an associated calculator to assist the user in setting the appropriate bits with minimal effort. For

the convenience of the user, calculator buttons have also been added to windows containing these registers. Only one calculator instance is available for each register. See MsAssist Calculators for calculator use. In addition, these calculators are accessible through the **Ref** drop-down menu.

0.4 MsAssist Calculators

The MsAssist calculators serve a dual purpose. They provide the bit name and number for each bit in a selected register. They also allow users to calculate a register's value quickly, based on required bit settings. A bit toggling function is used to update the value of the register as each bit is selected. The calculator's display window shows the value of the register, in hex, based on the bits selected. See the References (Ref) Drop-Down Menu section on page 36 for a description of each of the calculators available in MsAssist.

As each bit is selected, the calculator's display window changes to reflect the new value.

In addition, if a calculator is selected by clicking a window's calculator button, the windows parameter also displays the new value. To copy the value, the user may use either the double-click or drag-and-drop method. Note that only appropriate parameters are enabled to accept values using the drag-and-drop method. To drag the value, double-click the value, hold the left mouse button down, drag the value to the appropriate parameter box and release the left mouse button. The value displays where it was dropped.

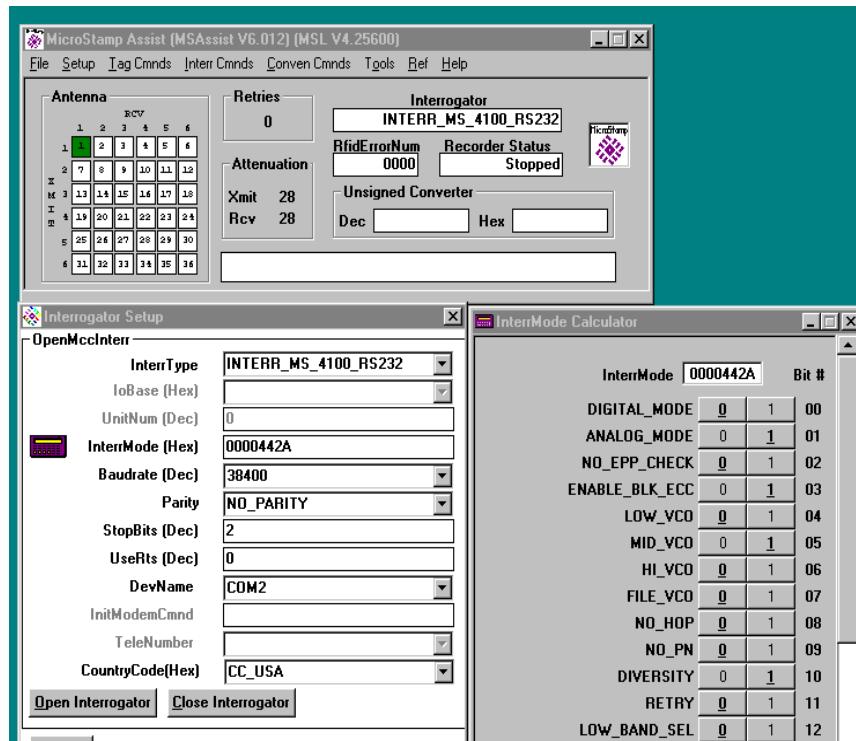


Figure 3 - MsAssist Calculator

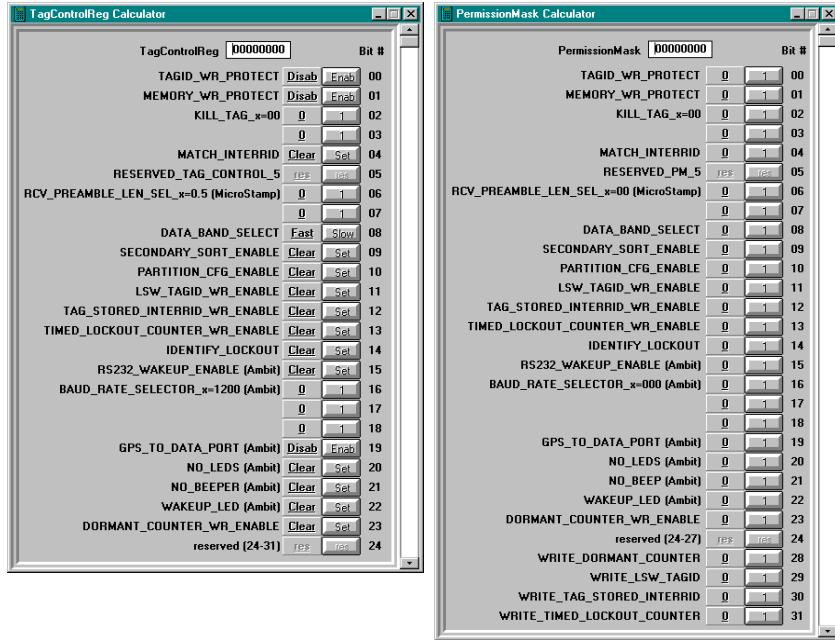


Figure 4 - TagControlReg and PermissionMask Calculators

In most cases, every bit may be set or cleared independently of the other bits, with some exceptions. For example, the *BaudRateSelection* found in the *TagControlReg Calc*, *PermissionMask Calc*, and *WriteTagRegs UpdateFlag Calc*, combines bits 6 and 7. The *RCV_PREAMBLE_LEN_SEL_x* label displays the value of the *BaudRateSelection*, depending on the settings of bits 6 and 7 in the *TagControlReg Calc*. The *RCV_PREAMBLE_LEN_SEL_x* label in the *PermissionMask Calc* and *WriteTagRegs UpdateFlag Calc* displays selected bits. Calculators may be minimized by selecting the **Minimize** button in the upper right-hand corner of the calculator. Calculator windows may be closed by selecting **Close** from the control box.

0.5 MsAssist Functionality

The **MicroStamp Assist**¹ main window consists of the following:

- Title bar
- Antenna box
- Retries box
- Attenuation box
- Interrogator field
- RfidErrorNum field
- Recorder Status field
- Converter group box
- (unlabeled) Line Animation/Status field
- Menu bar

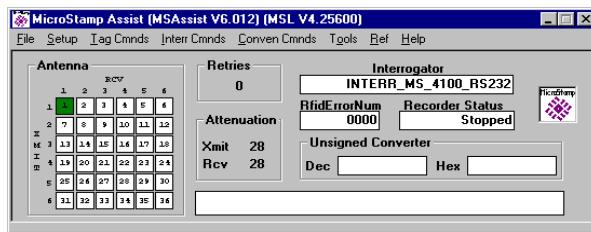


Figure 5 - MsAssist Main Window

The following sections describe each menu bar item and each feature (boxes and fields) within the MsAssist main window.

0.5.1 Title Bar

The title bar of the MsAssist program displays the software version and the version of the MicroStamp Library (in parentheses).



Figure 6 - MsAssist Title Bar

0.5.2 Antenna Box

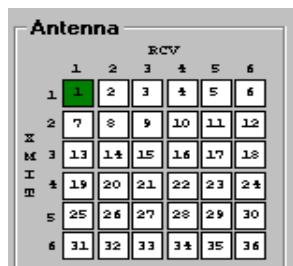


Figure 7 - Antenna Box

The **Antenna** box displays up to six transmit and six receive antenna pair combinations and the associated numeric value of that pair. Only valid antenna pair values are displayed for the requested interrogator. The highlighted value identifies the current antenna pair. The *GetReplyStats* function returns *Antenna* as one of the reply parameters. In the example graphic to the left, the *Antenna* value returned is one, which indicates that the current antenna pair is Transmit 1/Receive 1 (T1R1). If a 4000 series interrogator is open, valid antenna values are 1,2,4, and 8, representing T1R1, T1R2, T2R1, and T2R2.

For the 4100 series interrogators, the *Antenna* box may be used as an antenna pair reference for initializing the *AntennaQueue* or for the *SelectAntenna* function.

1. The **MicroStamp Assist** window is also referred to as the **MsAssist** window throughout this document.

The **Antenna** box indicates the current antenna pair and the status of a tag's reply. If the last tag communication was successful, the green box reflects the antenna pair successfully used. If the attempt resulted in error, the red box represents the last antenna pair tried.

0.5.3 Retries Box



Figure 8 - Retries Box

After every tag reply is read, the current *Retries* values appear. The **Retries** box displays the current command retry count (the number of times the command was re-transmitted (by the MSL) during the last tag-specific command). The displayed *Retries* value indicates which retry attempt caused the successful reply, or for an unsuccessful retry attempt, the *Retries* value indicates the maximum *Retries* value.

0.5.4 Attenuation Box

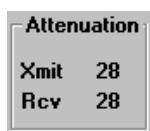


Figure 9 - Attenuation Box

The **Attenuation** box displays the attenuation settings for the current transmit and receive antenna pair. This value is obtained by sending a *GetReplyStats* command following a tag-specific command. This feature is valid for 4000 series interrogators and above.

0.5.5 Interrogator Field



Figure 10 - Interrogator Field

The **Interrogator** field displays the current interrogator selection (See “Interrogator Commands (Interr Cmnds) Drop-Down Menu” on page 19. for a discussion on how to select an interrogator). If the interrogator fails to open, the **Interrogator Setup**

window will be displayed to indicate the failure. A beep and red text in the **Interrogator** field will also indicate that a problem exists.

0.5.6 RfidErrorNum Field



Figure 11 - RfidErrorNum Field

The **RfidErrorNum** field displays the current *RfidErrorNum* after each function call to the MSL. It is similar in function to the *errno* variable used in C programming. The values are defined in rfidhost.h and may be referenced by selecting **RfidErrorNum** from the

Ref drop-down menu. If an open interrogator function call fails, the *RfidErrorNum* is retrieved to further clarify the type of failure. If an attempt to open an interrogator fails, the *RfidErrorNum* description is displayed in the **Status** box (See “Line Animation/Status Field” on page 11.). The *RfidErrorNum* is also displayed during execution of **Dots Animation** or **Line Animation** if bad reads or writes are detected.

0.5.7 Recorder Status Field

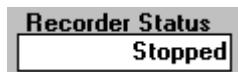


Figure 12 - Recorder Status Field

The **Recorder Status** field displays the current state of the **Macro Play/Record** function. The state is dependent on the function button selected. The possible states include Checking, Loading, Loaded, No Load File, Recording, Continuous, Playing, Paused, Play done, and Stopped.

0.5.8 Converter Group Box

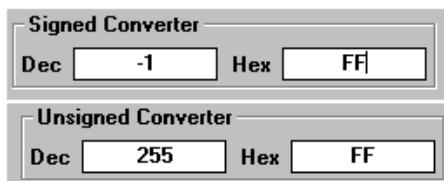


Figure 13 - Converter Group Box

The **Converter** group box converts decimal and hexadecimal number systems. For example, the decimal equivalent to a valid hexadecimal number is obtained by entering the hexadecimal number into the **Hex** field of the converter. The decimal equivalent simultaneously displays in the **Dec** field of the converter. A decimal value entered in the **Dec** field causes its equivalent hexadecimal value to display in the **Hex** field.



Figure 14 - Tools Drop-Down

The **Converter** group box may be toggled from unsigned to signed by selecting the converter option in the **Tools** drop-down menu or using the hot key **<CTRL><C>**. The label on the **Converter** group box indicates the current state of the **Converter** group box. The **Converter** option in the **Tools** drop-down menu indicates the next state.

The unsigned **Hex** range is 0x00000000 to 0xFFFFFFFF and the **Dec** range is 0 to 4,294,967,295. The signed hexadecimal values are calculated based on the most significant nibble. **Hex** values 0x00 to 0x7F are positive and 0x80 to 0xFF are negative. The **Dec** range is -2,147,483,648 to 2,147,483,647. Values that are outside of the defined ranges will result in an overflow error.

0.5.9 Line Animation/Status Field



Figure 15 - Line Animation/Status Field

The **Line Animation/Status** field is unlabeled. This field is used during **Line Animation** and displays various system information. When **Line Animation** is

enabled (through the **Tools** drop-down menu), the mnemonic representation of each tag-specific command reply displays. When **Line Animation** is disabled, the **Line Animation/Status** field is no longer updated. The data in the **Line Animation/Status** field is cleared when **Line Animation** is re-enabled. The **Line Animation/Status** field is also used to display the **RfidErrorNum** description when the **RfidErrorNum** is greater than zero following a tag-specific command or when an interrogator fails to open.

0.5.10 Menu Bar

The Menu bar provides environment setup, command selections, tools for arranging command windows, function utilities, and references for selected function parameters.



Figure 16 - MsAssist Menu Bar

The MSL Function Descriptions on page 55 provides an overview for all selectable functions in the **Tag Cmnds**, **Interr Cmnds**, and **Conven Cmnds** drop-down menus. A command is executed by selecting the command window's **Send** button. Each command window contains the parameters needed to execute the function. The user is responsible for entering valid parameters.

0.5.10.1 File Drop-Down Menu



Viewer or **Exit** can be selected from the **File** drop-down menu.

Figure 17 - File Menu

0.5.10.1.1 Viewer

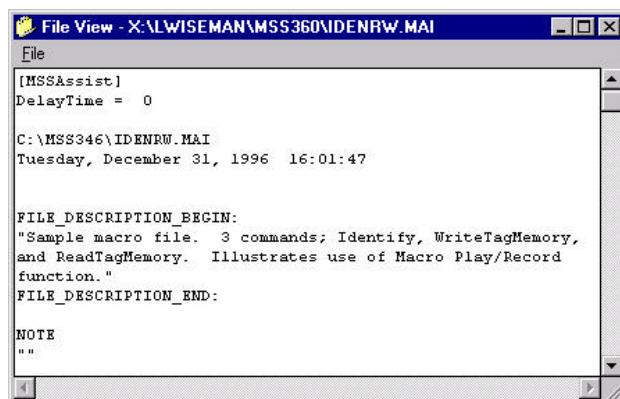


Figure 18 - Viewer

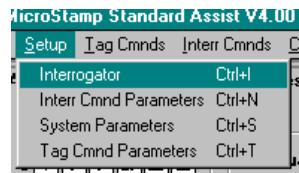
The **Viewer** window displays small files less than 30K. This is helpful for viewing macro and result files. The **Viewer** window does not provide file editing features. The title of the **Viewer** window lists the name and path of the displayed file.

Close the **Viewer** window by selecting **Exit** from the **File** drop-down menu.

0.5.10.1.2 Exit

The **Exit** option closes the current interrogator (if it is open) and all **MsAssist** windows.

0.5.10.2 Setup Drop-Down Menu



Modifications to the interrogator, interrogator command parameters, system parameters, or the global tag command parameters are made from the **Setup** drop-down menu.

Figure 19 - Setup Menu

0.5.10.2.1 **Interrogator <CTRL><I>**

The interrogator parameters are read from the MsAssist.ini file on startup. Supported interrogators include enhanced parallel port (EPP) and serial (RS-232, RS-422, and RS-485) ports. The EPP is an 8-bit bi-directional digital port (IEEE-1284). RS-232 uses the standard PC serial port. RS-422 and RS-485 require special communication cards to be installed in the PC.

If MsAssist is executed without the MsAssist.ini file, the **Interrogator Setup** window appears, and the **Interrogator** field (See “Interrogator Field” on page 10.) will indicate (in blue) “No Interrogator.”

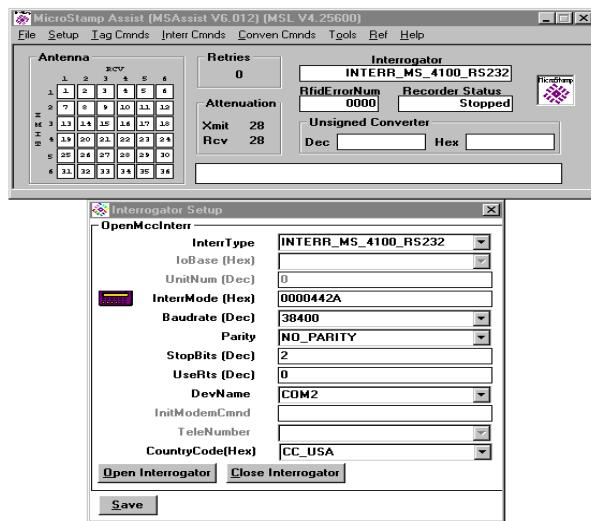


Figure 20 - Select Interrogator

Select a valid interrogator from the **InterrType** list box. The *Open_MCC_interr* function is used to connect to a selected interrogator (*InterrType*). For the user’s convenience, MsAssist enables and disables parameters in the **Interrogator Setup** window, depending on the selected interrogator type.

Open Interrogator Button

The **Interrogator Setup** window provides a list of parameters available from the *OPEN_MCC_INTERR_IO* structure.¹ Refer to the *OpenMccInterr* function

1. Not all parameters are required for all interrogator types.

section on page 84 for valid parameter values. The **InterrMode** parameter has an associated calculator (See “MsAssist Calculators” on page 7.) with which the user can calculate a valid *InterrMode*.

When the **Open Interrogator** button is selected, any previously opened interrogator will be closed before an attempt is made to open the interrogator defined in the **InterrType** field. After a successful open, MsAssist displays (in black) the selected interrogator name in the **Interrogator** field on the main window.

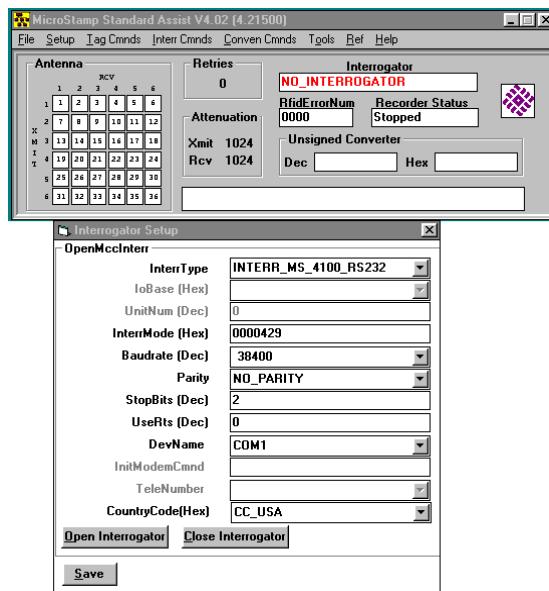


Figure 21 - Interrogator Setup

If the interrogator fails to open, the selected interrogator shown in the **Interrogator** field will change to red. If an error occurs, the error message associated with the *RfidErrorNum* is displayed (in red) in the **Line Animation/ Status** field, and the **RfidErrorNum** field will display the numeric value.

Close Interrogator Button

The **Close Interrogator** button closes an open interrogator. To open another interrogator select the **Open Interrogator** button.

Save Button

Select the **Save** button to close the **Interrogator Setup** window and save valid interrogator information to the MsAssist.ini file. Interrogator information is considered valid only if the interrogator opens successfully. For modems, new telephone number entries are added to the telephone number list in the MsAssist.ini file. If **Save** is selected and the interrogator fails to open, the MsAssist.ini file is not modified.

0.5.10.2.2 Interr Cmnd Parameters <CTRL><N>

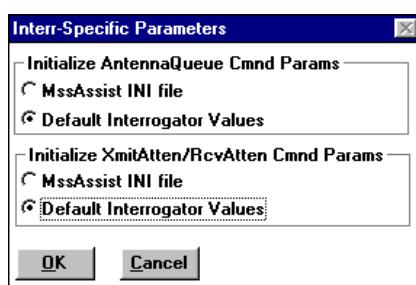


Figure 22 - Interr-Specific Parameters

The **Interr Cmnd Parameters** option is available for the 4100 series interrogators only. The *AntennaQueue* and *XmitRcvAtten* initialization parameters are saved in the **Interr Defaults** section of the INI file when **OK** is selected. This window is modal and must be closed to access other windows in MsAssist.

If the **MsAssist INI file** button is selected, an attempt to initialize the local

AntennaQueue, *XmitAtten*, and *RcvAtten* arrays from the MsAssist.ini file is made. If the **Default Interrogator Values** button is selected, the local arrays will be initialized with the interrogator's default *AntennaQueue*, *XmitAtten*, and *RcvAtten* values.

These local arrays are used with the **SetInterrRegs**, **SetPowerAttenDac**, and **Interrogator/Antenna Test** windows.

0.5.10.2.3 System Parameters <CTRL><S>

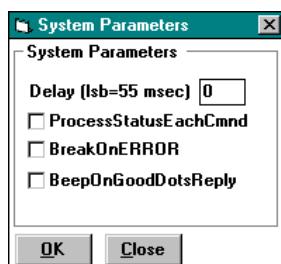


Figure 23 - System Parameters

Some system parameters are saved in the MsAssist.ini file by selecting the **OK** button. The **System Parameters** window is modal and must be closed to access other windows in MsAssist.

Delay Box

Delay is used during **Play** or **Continuous** mode by the **Macro Play/Record** option. After each command executes, a delay occurs based on this parameter. A 0 value causes no additional delays to occur between commands. Delay ranges from 0 to 20 and represent multiples of 55 milliseconds (e.g. a Delay of 2 equals approximately 110 milliseconds). Selecting the **OK** button saves this value to the MsAssist.ini file.

ProcessStatusEachCmnd Button

When **ProcessStatusEachCmnd** option is selected, the **Line Animation/Status** field, **Antenna** box, **Retries** box, **Attenuation** box, and **RfidErrorNum** field are updated after processing the status of tag-specific commands. Selecting **ProcessStatusEachCmnd** causes the **MsAssist** window to remain the top-most window on the screen. When **ProcessStatusEachCmnd** is not selected, these windows are updated after the

last command in the command list executes. See the Command List section on page 4 for a description of the command list. This value is not stored in the MsAssist.ini file.

BreakOnERROR Button

BreakOnERROR is used during play or continuous mode (selectable using the *Macro Play/Record* function) to halt a command if an error occurs. See the Command List section on page 4, for a description of the commands in the command list. If **BreakOnERROR** has been selected, and an uncorrectable reply error occurs, the executed command list macro terminates following the function returning the error.

Do not enable this parameter while recording macros: an error causes the recording session to terminate before completion. This value is not stored in the MsAssist.ini file.

BeepOnGoodDotsReply

When **BeepOnGoodDotsReply** is selected, and the **Dots Animation** function is running, a good response will be punctuated by a beep from the computer.

OK Button

Selecting the **OK** button updates the value of **Delay** in the MsAssist.ini file.

Close Button

Selecting the **Close** button closes the **System Parameters** window without modifying the MsAssist.ini file.

0.5.10.2.4 Tag Cmnd Parameters <CTRL><T>

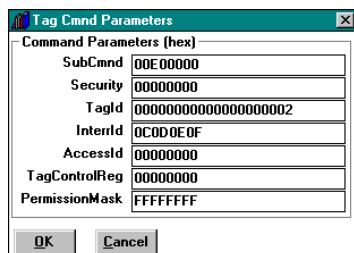


Figure 24 - Tag Cmnd Parameters

The **Tag Cmnd Parameters** window is modal and must be closed to access other windows in MsAssist. Changes made to these parameters are saved in the MsAssist.ini file after selecting **OK**. Options from the **Tag Cmnds** drop-down menu opened after this window is closed will reflect the new parameter values.

Command Parameters Group Box

The **Command Parameters** group box contains the global command parameters used by multiple tag-specific functions. These values are default settings and may be changed. Each field is automatically formatted according to its type. For example, if "20" is entered in the **SubCmnd** field, the field is formatted to "00000020" after selecting **OK**. To use the new parameters, these values must be changed before any of the options from the **Tag Cmnds** drop-down menu are selected. All of the parameters are updated in the MsAssist.ini

file after selecting **OK**.

OK Button

Selecting the **OK** button updates any changes to the command parameters in the MsAssist.ini file and closes the **Tag Cmnd Parameters** window.

Cancel Button

Selecting the **Cancel** button closes this window without updating the MsAssist.ini file.

0.5.10.3 Tag Commands (Tag Cmnds) Drop-Down Menu

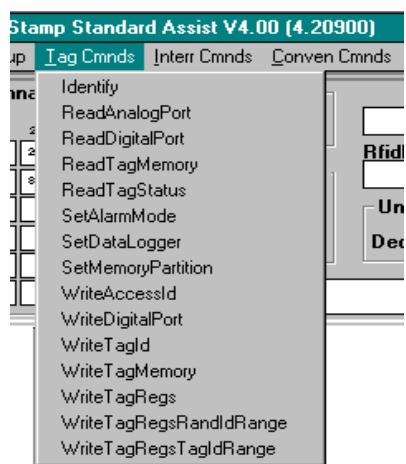


Figure 25 - Tag Commands Menu

The **Tag Cmnds** drop-down menu provides a list of the tag-specific MSL functions available in MsAssist. For each function selected, a unique command window displays the applicable command parameters, interrogator reply header, and reply parameters. Some commands are interrogator-specific and may not be available.

“Command Window Layout” on page 4 address the command windows in general. The following is a list of the current tag-specific functions supported by the MSL. Detailed information about each function is covered in the MSL Function Descriptions on page 55.

0.5.10.3.1 Identify

The *Identify* function is used when attempting to determine an unknown *TagId*.

0.5.10.3.2 ReadDigitalPort

The *ReadDigitalPort* function returns data read from a tag's data port.

0.5.10.3.3 ReadTagMemory

The *ReadTagMemory* function returns data from a tag's user memory.

0.5.10.3.4 ReadTagStatus

The *ReadTagStatus* function returns system information about a tag.

0.5.10.3.5 SetMemoryPartition

The *SetMemoryPartition* function initializes the parameters of a memory partition.

0.5.10.3.6 WriteAccessId

The *WriteAccessId* function is used to update an *AccessId* for one of the memory partitions.

0.5.10.3.7 WriteDigitalPort

The *WriteDigitalPort* function writes data to a tag's data port.

0.5.10.3.8 WriteTagId

The *WriteTagId* function is used to update a tag's *TagId*.

0.5.10.3.9 WriteTagMemory

The *WriteTagMemory* function is used to write to a tag's user memory.

0.5.10.3.10 WriteTagRegs

The *WriteTagRegs* function modifies only those bits and registers indicated in the *UpdateFlag*. Tag registers include the *LswTagId*, *TagStoredInterrId*, *TimedLockoutCounter*, *DormantCounter*, and *TagControlReg* (each bit in the *TagControlReg* may be individually modified).

0.5.10.3.11 WriteTagRegsRandIdRange

The *WriteTagRegsRandIdRange* function modifies only those bits and registers indicated in the *UpdateFlag*. Tag registers include the *LswTagId*, *TagStoredInterrId*, *TimedLockoutCounter*, *DormantCounter*, and *TagControlReg* (each bit in the *TagControlReg* may be individually modified) for a range of *RandomValueIds*.

0.5.10.3.12 WriteTagRegsTagIdRange

The *WriteTagRegsTagIdRange* function modifies only those bits and registers indicated in the *UpdateFlag*. Tag registers include the *LswTagId*, *TagStoredInterrId*, *TimedLockoutCounter*, *DormantCounter*, and *TagControlReg* (each bit in the *TagControlReg* may be individually modified) for a range of *TagIds*.

0.5.10.4 Interrogator Commands (Interr Cmnds) Drop-Down Menu

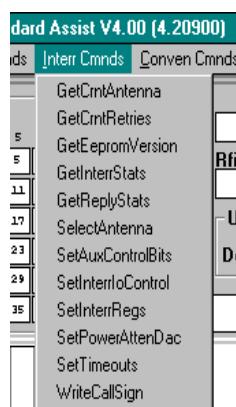


Figure 26 - Interrogator

The **Interr Cmnds** drop-down menu lists each of the MSL interrogator-specific functions available in MsAssist. Each function displays the applicable command parameters, interrogator reply header, and reply parameters or a subset of these parameters. Some commands are interrogator-specific and may not be available.

“Command Window Layout” on page 4 address the command windows in general. The following is a list of interrogator-specific functions available in MsAssist. Detailed information about each of these functions is located in the MSL Function Descriptions on page 55.

0.5.10.4.1 GetCrntAntenna

The *GetCrntAntenna* function returns the current antenna set used to communicate with a tag.

0.5.10.4.2 GetCrntRetries

The *GetCrntRetries* function returns the number of times a command was retransmitted during the last tag-specific command.

0.5.10.4.3 GetXilinxVersion

The *GetXilinxVersion* function returns information about the permanent and updateable embedded firmware.

0.5.10.4.4 GetInterrStats

The *GetInterrStats* function returns record-keeping parameters (as long as the interrogator is programmed to perform this function).

0.5.10.4.5 GetReplyStats

The *GetReplyStats* function returns values that are specific to the last tag-specific reply (as long as the interrogator is programmed to perform this function).

0.5.10.4.6 SelectAntenna

The *SelectAntenna* function is used to select a specific antenna pair. Diversity should be disabled to ensure the selected antenna pair is used. This function is available on 4100 series interrogators only.

0.5.10.4.7 SetAuxControlBits

The *SetAuxControlBits* function allows users to regulate general purpose

control lines to other devices, such as external multiplexors with different antenna pairs.

0.5.10.4.8 SetInterrIoControl

The *SetInterrIoControl* function is not implemented at this time.

0.5.10.4.9 SetInterrRegs

The *SetInterrRegs* function is used to set various communication parameters on an interrogator. Not all of the parameters are used on all interrogators.

0.5.10.4.10 SetPowerAttenDac

The *SetPowerAttenDac* function is used to set the power of the forward and return links for each antenna pair.

0.5.10.4.11 SetTimeouts

The *SetTimeouts* function is used to set the system watchdog timers.

0.5.10.4.12 WriteCallSign

The *WriteCallSign* function is used to return the interrogator's *CallSign*. A call sign is required for all interrogators operating in Japan.

0.5.10.5 Convenience Commands (Conven Cmnds) Drop-Down Menu

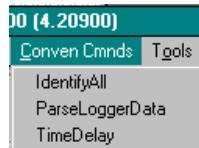


Figure 27 - Convenience Command Menu

The **Conven Cmnds** drop-down menu lists MSL convenience functions available in MsAssist in addition to MsAssist-specific convenience functions. The **Conven Cmnds** are designed to decrease the workload of application programmers by providing functions, with possible combinations of tag-specific and interrogator-specific commands, to allow simplified access to commonly performed tasks. For each selected convenience

function, a unique command window displays applicable parameters available for that function.

“Command Window Layout” on page 4 address the command windows in general. The following is a list of the convenience functions available in MsAssist. Detailed information about each MSL function is located in the MSL Function Descriptions on page 55.

0.5.10.5.1 IdentifyAll

The **IdentifyAll** function should only be used with an analog interrogator. This function returns the number of tags found within the system's communication range. The *IdentifyAll* reply parameters include the *TagId* and *RandomValueId* for each tag identified.

0.5.10.5.2 TimeDelay

The **TimeDelay** function provides a delay for a specified time in milliseconds. This function may be used between tag commands to allow the specified amount of time to pass before the next function executes. This function differs from the **Delay** function found in the **System Parameters** window (See “System Parameters <ctrl><s>” on page 15.). The **TimeDelay** function is a Convenience Command, specific to MsAssist, and is not found in the MSL. This function is similar to other commands and only executes when selected. The **Delay** function, if selected, forces the PC to delay for the requested time period, following each command.

0.5.10.6 Tools Drop-Down Menu



The **Tools** drop-down menu provides window manipulation utilities and other useful utilities. The first four tool selections (Cmnds tools) may be used to manipulate the command list windows. **|Dots Animation** and **Interrogator/Antenna Test** incorporate a combination of functions from the MSL to perform specific tasks. The macro tools provide functionality to save and restore frequently used MSL command sequences and the converter may be toggled to display signed or unsigned values.

Figure 28 - Tools Menu

0.5.10.6.1 Cascade All Cmnds <SHIFT><F5>

Cascade All Cmnds may be selected to rearrange all command windows in the command list, including any iconized command windows. See the Command List section on page 4 for a description of the command list.

When the **Cascade All Cmnds** function is selected, the cursor changes to a crosshair. Place the cursor where the command windows are to be positioned, and click the left mouse button. All command windows, including any iconized command windows, are cascaded starting at the cursor position. Any iconized window is restored to normal size before being repositioned.

0.5.10.6.2 Cascade Cmnds <SHIFT><F6>

Cascade Cmnds may be selected to rearrange all non-iconized command windows in the command list. The **Cascade Cmnds** function operates the same as the **Cascade All Cmnds**, but only repositions command windows which are not iconized. See the Command List section on page 4 for a description of the command list.

0.5.10.6.3 Iconize Cmnds <SHIFT><F7>

Iconize Cmnds may be selected to iconize all command windows in the

command list. See the Command List section on page 4 for a description of the command list.

0.5.10.6.4 Reposition Cmnds <SHIFT><F8>

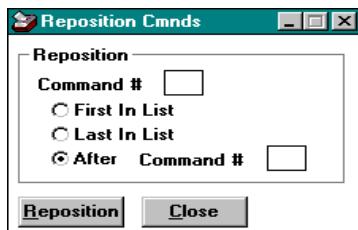


Figure 29 - Reposition Cmnds

The **Reposition Cmnds** window allows users to reorder the commands in the command list. The **Reposition Cmnds** window does not display if fewer than two commands are in the command list. A command can be positioned first, last, or after a specific command. See the Command List section on page 4 for a description of the command list.

To reposition a command window, enter the command's number (found in the command's title) into the **Command #** box and select the new position. To initiate the reordering, press the **Reposition** button.

Reposition Button

The **Reposition** button initiates the reordering of the command list.

Close Button

The **Close** button closes the **Reposition Cmnds** window. The window may also be closed by selecting **Close <ALT><F4>** from the control box in the upper left corner of the **Reposition Cmnds** window.

0.5.10.6.5 Dots Animation

The **Dots Animation** option sends commands continuously to a tag and displays the command status in a mnemonic format. The tag-specific function parameters *SubCmnd*, *InterrId*, *Security*, and *AccessId* (found in **Setup**, under **Tag Cmnd Parameters**), are used as the function parameters.

When the **Start** button is selected, an *Identify* command, with the *CLR_IDENTIFY_LOCKOUT* bit set in the *SubCmnd*, is continuously sent until only one tag within range of the interrogator responds. The tag's *TagId* is then displayed next to the **TagId** label. If a tag does not respond, an 'N' is displayed in the **Reply Mnemonics** window each time an *Identify* command is sent. This occurs until a response is received or execution is stopped.

After the *TagId* is obtained, *ReadTagMemory* or *WriteTagMemory* commands are sent to the tag, and the mnemonic representation of each reply status is displayed in the **Reply Mnemonics** window.¹ The **Packets** group box displays a running count and percent packet error.

1. The **Mnemonics Definitions** button at the lower right corner of the **Dots Animation** window may be selected to view the currently defined mnemonics.

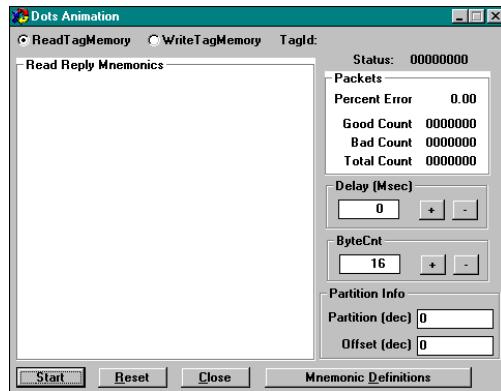


Figure 30 - Dots Animation Window

The *ReadTagMemory* or *WriteTagMemory* command is continuously sent until the execution is stopped by selecting the **Stop** button. The **Start** button label changes to **Stop** when **Start** is selected and changes to **Start** when **Stop** is selected.

ReadTagMemory Button

When the **ReadTagMemory** button is selected, *ReadTagMemory* commands are sent to a tag. The reply window title displays "Read Reply Mnemonics." **ByteCnt** bytes of data from **Partition's Offset** are read from the tag's memory.

WriteTagMemory Button

When the **WriteTagMemory** button is selected, *WriteTagMemory* commands are sent to a tag. The reply window title displays "Write Reply Mnemonics." The tag's memory from **Partition's Offset** is overwritten with **ByteCnt** bytes of 0xA5 data.

TagId Field

The **TagId** may be entered manually or left empty. If the **TagId** field is empty, selecting the **Start** button will cause continuous *Identify* commands to be sent until an interrogator receives a good reply. While waiting for a good response, "Searching..." is displayed in the **TagId** field. After receiving a good *Identify* reply, the *TagId* is displayed. Selecting the **Reset** button while *Dots Animation* is executing, causes *Identify* commands to be sent continuously until a tag responds. If the **TagId** field is not empty, selecting the **Start** button causes continuous read or write commands to be sent to a tag whose *TagId* is displayed.

Mnemonic Definitions Button

Mnemonic Definitions		
	Bit Weight	Description
N	0x00000000	Tag not identified (Status = 0x00000020)
.	0x00000008	Good reply (Status & 0x0037F3FF = 0)
P	0x00000004	Invalid command parameter detected.
X	0x000000100	Forward-link message not sent by interrogator.
H	0x000000040	Host to interrogator communications error.
I	0x000000080	Interrogator timed out sending RIC unit command.
T	0x000001000	Host timed out waiting for interrogator reply.
E	0x000002000	Interrogator to host communications error.
R	0x000004000	Host to interrogator CRC error detected.
G	0x000008000	Tag to interrogator CRC error detected.
C	0x000000010	Interrogator to host CRC error detected.
D	0x000000020	Collision detected while receiving reply.
F	0x000000001	Interrogator timed out first byte.
S	0x000000002	Interrogator timed out secondary byte.
L	0x0000000001	Valid lock detect signal not observed.
U	0x0000000002	Uncorrectable error detected from the return link.
-	0x0000000004	Correctable error detected from the return link.
c	0x0000000008	Correctable error detected from the forward link.
v	0x0000000010	Battery voltage on RIC unit low.
p	0x0000000020	Attempt to write protected area.
i	0x0000000040	Attempt to access invalid address.
a	0x0000000080	AccessId received for partition is invalid.
M	0x0000000100	Received InterId does not match.
B	all others	all others

Figure 31 - Mnemonic Definitions

This button is located in the lower right corner of the **Dots Animation** window. When pressed, the **Mnemonic Definitions** window displays a list of characters and the associated reply status that the character represents. After each reply status is received, its associated mnemonic character displays in the **Reply Mnemonics** window.

Packets Group Box

The **Packets** group box displays a Total Count of *ReadTagMemory* or *WriteTagMemory* commands sent. It also displays the number of good (**Good Count**) and bad (**Bad Count**) responses and the **Percent Error**. Good command responses include '.', 'c', '-' and 'v'. Refer to the **Mnemonics Definitions** window for the definition of these mnemonics.

Delay Box

The **Delay** box indicates the time delay between commands. The + and - buttons increment or decrement the delay in multiples of 55 milliseconds. **Delay** can be modified without halting execution.

ByteCnt Box

The **ByteCnt** box displays the amount of memory the command reads or writes. The + and - buttons increment or decrement the *ByteCnt* through a pre-defined set. The set is 0, 1, 2, 4, 8, 16, 32, 64, 127-129, 255-257, 512, and 1024. The *ByteCnt* may also be entered manually. The *ByteCnt* is read before each command is sent.

Partition Info Group Box

The **Partition** and **Offset** fields are used to determine the location within the **UserMemory** that the command reads or writes. These values are in decimal format and unlike the **Delay** and **ByteCnt** fields, are only read when **Start** is selected. Note that a *ReadTagStatus* command will return partition information that may be useful.

Start Button

The **Start** button, when selected, begins the process of acquiring a tag's *TagId* and if successful, continuously send *ReadTagMemory* or *WriteTagMemory*

commands to a tag. The button label will toggle between **Start** and **Stop**. When **Stop** is selected, command execution halts.

Reset Button

When the **Reset** button is selected, the data in the **Packets** group box is reset to zeros, the **TagId** field is cleared, and the **Reply Mnemonics** window is cleared. If **Dots Animation** is running when the **Reset** button is selected, an *Identify* command sequence will be sent (until a valid tag response is received) before continuing with the *ReadTagMemory* or *WriteTagMemory* command.

Close Button

Pressing the **Close** button closes the **Dots Animation** window. This window may also be closed by selecting **Close** from the control box in the upper left corner of the **Dots Animation** window. **Dots Animation** must be stopped (**Stop**) before closing the window.

0.5.10.6.6 (Start / Stop) Line Animation <CTRL><A>

Selecting **Start Line Animation** from the **Tools** drop-down menu enables the **Status** box (See the Line Animation/Status Field section on page 11), which displays the mnemonic representation of a command's status. When the **Tools** drop-down menu is selected again, **Stop Line Animation** can be selected to disable line animation. While animation is enabled, every reply from a tag command is evaluated and displayed in mnemonic format in the **Status** box.

0.5.10.6.7 Macro Play/Record



Figure 32 - Macro Play/Record

The **Macro Play/Record** function saves command sequences that users can later load and replay. The **Macro Play/Record** option operates similarly to a tape player by replaying an existing command list or loading a previously saved command sequence

(macro). See the Command List section on page 4 for a description of the command list. **Macro Play/Record** can be selected from the **Tools** drop-down menu. The buttons are enabled or disabled depending on the current state of the play status. The **Recorder Status** field always displays the current state of the player (See "Recorder Status Field" on page 11.).

Load Button

The **Load** button allows users to load previously saved macro files. Macro files usually have an extension of .MAI, but may be named anything, as long as the title follows proper MS-DOS file naming conventions. The **Recorder Status** field indicates "Checking" until the requested file is located. A signature stamp in the macro file is used for file verification and the command list description at the end of the file is used to rebuild the command list windows.

Macro files are text files, and an example file is included on the installation disks. While the file is being processed, "Loading" appears in the **Recorder Status** field. If the file loads without error, the **Recorder Status** field displays "Loaded."

Pause Button

The **Pause** button changes the **Recorder Status** to "Pause." This state will remain in effect until a different selection is made.

Record Button

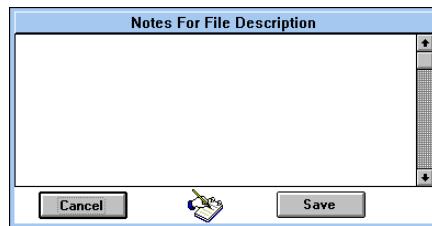


Figure 33 - Notes for File Description

If the command list is not empty, the user is prompted to enter a filename before saving the macro. A dialog box for the file description displays automatically. This allows users to enter a description of the file. Select the **Save** button to save the comments to the macro file or **Cancel** to close the dialog box without saving the description. The command list is executed

from the first command to the last. For each command that is sent, the command parameters and reply parameters are saved. When the last command executes, a copy of the command list is stored to the macro file. This list is used to define the command list when users load a macro file. When the process completes, the macro file closes.

The signature stamp, the name of the file, and a timestamp is recorded at the beginning of the macro file. The description block is stored next, followed by command data, and finally the command list description. The **Recorder Status** field indicates "Recording" while the command list processes, and it indicates "Stopped" when the process stops.

Notes Button

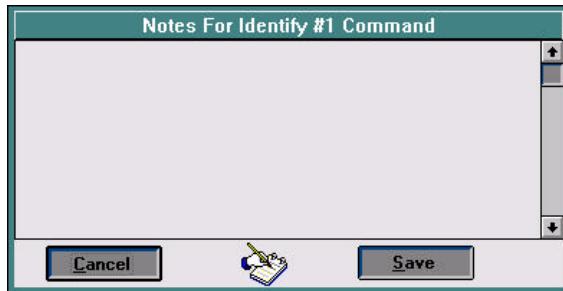


Figure 34 - Notes Window

The **Notes** button allows comments to be saved for each command in the command list. When the **Notes** button is selected, a note is attached to the last command window to have the focus prior to selecting **Notes**. The **Notes** window title includes the caption from the associated command window.

To include notes in the macro file, they must be saved before selecting the **Record** button. When the macro is loaded again, the notes are read from the macro file. This makes it possible to load a macro file, make changes, and record it again. The **Recorder Status** field does not change.

Play Button

The **Play** button executes each command in the command list, beginning with the first command. The **Recorder Status** field indicates "Play" while the command list plays and "Play done" when the process completes.

Continuous Button

The **Continuous** button executes each command in the command list beginning with the first command. This process repeats until the **Stop** button is selected, or the <ESC>, <RTN>, or <SPACE> key is pressed.¹ The **Recorder Status** field is set to "Continuous" while the command list is playing and "Play done" when execution stops.

Stop Button

The **Stop** button halts the current process and set the **Recorder Status** field to "Stopped."

Exit Button

The **Exit** button resets the **Recorder Status** field to "Stopped" and closes the **Macro Play/Record** window. This window may also be closed by selecting **Close** from the control box in the upper left corner of the **Macro Play/Record** window.

0.5.10.6.8 Replay Macros

The **Replay Macros** option executes macro file(s) without generating a command list (See "Command List" on page 4. for a description of the command list). The **Replay Macros** option differs from the **Macro Play/Record** function in that a macro file is executed sequentially, while ignoring

1. In order to use the mouse to stop a command execution, the "Snap To" function in the mouse manager (in **Control Panels**) must be deselected.

the command list information. Each command's reply is compared to the corresponding reply in the macro file. If they do not match, a results file is created. The results file contains each command read from the macro file, the results of the command from the macro file, and the reply received from executing the command. The results file is saved using the prefix name from the macro file and the extension .RES.

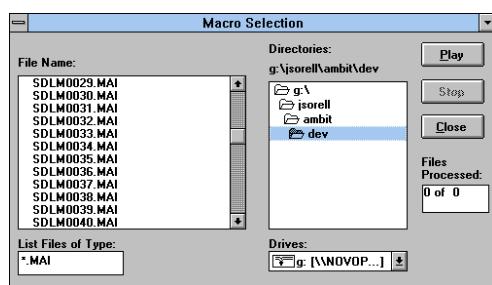


Figure 35 - Macro Selection

files may be halted at any time by selecting the **Stop** button. The macro replay stops after the completing the currently executing macro file.

The **Macro Status** window displays the results of the last executed macro file. As each macro file is executed, the results appear in this window. The purpose of the **Macro Status** window is only to provide visual feedback on the status of each macro file. The **BreakOnERROR** check box from the **Setup System Parameters** window may be selected to stop the execution of a macro if an error occurs.

File Name Box

The **File Name** box lists all files located in the current directory that match the pattern displayed in the **List Files of Type** field.

List Files of Type Box

The **List Files of Type** box allows only selected files to be displayed in the **File Name** box. The default type is *.MAI which is the default extension for macro files.

Directories Box

The **Directories** box allows users to select which directory to search. The currently selected directory name is displayed under the **Directories** caption.

Drives Field

The **Drives** field allows users to select which drive to search. The currently selected drive name is displayed under the **Directories** caption.

Because multiple file selection is available, a series of macro files may be replayed without user interaction. If errors are encountered in the macro file being executed, a results file is generated and saved before the next macro file executes. If the macro file executes without error, a results file will not be generated for that file. The execution of the macro

Play Button

The **Play** button executes selected macro files. If an invalid macro file is selected, a message window appears, indicating that an invalid file has been selected. After the message window's **OK** button is selected, the next macro file begins to execute.

Stop Button

At any time during the execution of the selected macro files, the **Stop** button may be selected to abort the sequence.

Close Button

The **Close** button will unload the **Replay Macro** window. The **Macro Status** window remains open until the user closes it. This window may also be closed by selecting **Close** from the control box in the upper left corner of the **Replay Macro** window.

Files Processed Field

The **Files Processed** field displays the number of files that have been processed and the number of files selected. As each macro file is executed, the number of files processed increments by one, and the name of the processed file is de-highlighted in the **File Name** box.

0.5.10.6.9 Signed / Unsigned Converter Group Box <CTRL><C>

For a general description of the converter tool, See “Converter Group Box” on page 11.. The MsAssist **Converter** group box default is unsigned. **Signed Converter** (displayed in the **Tools** drop-down menu) indicates the next state of the converter. When selected, the state of the converter changes to signed, and the title of the **Converter** group box changes to **Signed Converter**. Selecting the **Tools** menu again will show the next state of the converter, which is **Unsigned Converter**. This value is not saved in the MsAssist.ini file.

0.5.10.6.10 Interrogator/Antenna Test

The **Interrogator/Antenna Test** utility is accessible from the **Tools** drop-down menu. This utility provides a means for certified RF technicians or engineers to test the forward and return links on an interrogator. A valid password is required for accessing the **Interrogator/Antenna Test** utility to

prevent access by unqualified personnel.

Accessing The Interrogator/Antenna Test Window

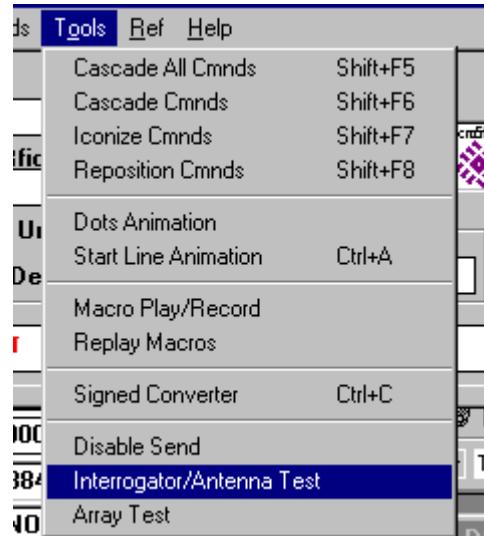


Figure 36 - Interrogator/Antenna Test Selection

The **Interrogator/Antenna Test** window is accessible from the Tools drop-down menu in **MsAssist**.

To prevent access from unqualified personnel, the **Interrogator/Antenna Test** window is password protected. This protection may prevent the equipment from accidentally being operated in a mode which violates FCC and other International Agency Regulations.

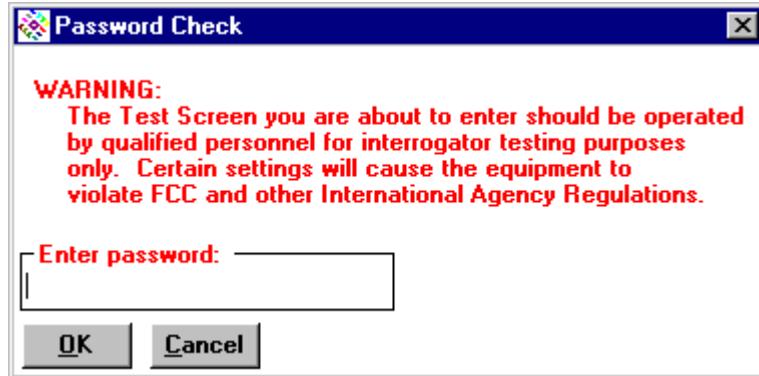


Figure 37 - Password Check Window

Authorized personnel should enter a valid password then select OK to access the test utility. Only valid passwords allow access to this test window.

Interrogator/Antenna Test Window

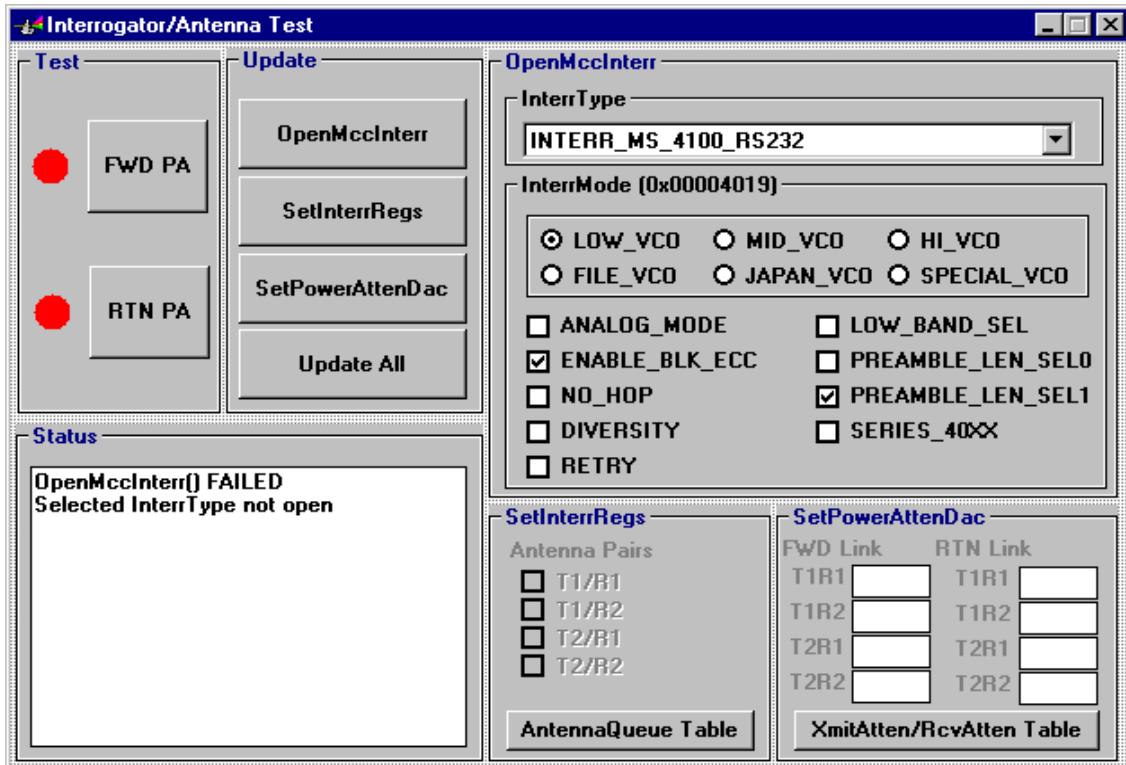


Figure 38 - Interrogator/Antenna Test Window

An open interrogator is closed prior to initializing the **Interrogator/Antenna Test** window. During initialization of the window, the current MsAssist startup parameters are used to determine which interrogator to open. Following a successful open of an interrogator, a *SetInterrRegs* command executes to retrieve the default antenna pair selection and power attenuator

DAC table settings.

The following sections define the various areas of the **Interrogator/Antenna Test** window

Test Window

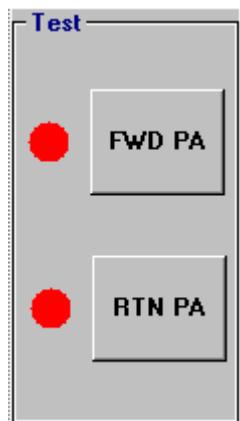


Figure 39 - Test Menu

Selection of either the **FWD PA** or **RTN PA** changes the state of the power amp. The interrogator's power amp is turned on or off depending on the current state of the power amp and the button selected by the user. If the user attempts to exit the **Interrogator/Antenna Test** window while the power amp is on, a command is automatically sent to the interrogator to turn off the power amp prior to closing the test window.

Selecting the **FWD PA** button disables the **RTN PA** button, turns the power amp on, and changes the red light adjacent to the **FWD PA** button to green.

PA button is selected.

FWD PA - *Cycles the power amp on and off. If the power amp is on, the forward signal is continuously transmitted.*

RTN PA - *Cycles the power amp on and off. If the power amp is on, the return CW is continuously transmitted.*

Update

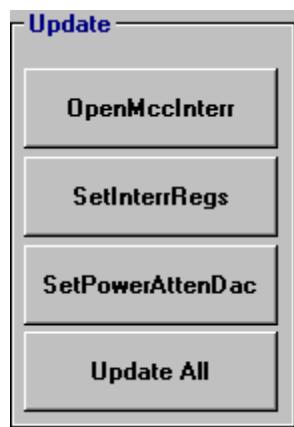


Figure 40 - Update¹ Menu

A subset of the MSL interrogator-specific commands may be executed by selection the appropriate button in this group. Each button's label identifies which interrogator-specific function is executed as well as the group of parameters used during execution of the function.

OpenMccInterr - Uses the parameters from the **OpenMccInterr** group box to open an interrogator. (Refer to the **OpenMccInterr** group description later in this document). Default parameters from the **MsAssist** initialization file are used in place of parameters unavailable in this window.

SetInterrRegs - Uses the transmit/receive antenna pairs from the **SetInterrRegs** group box to initialize the antenna pair selection. (Refer to the **SetInterrRegs** group description later in this document). If the **OpenMccInterr** button is selected, the **SetInterrRegs** group box is updated with the default antenna pair selection.

SetPowerAttenDac - Uses the **SetPowerAttenDac** group box parameters to update interrogator's power attenuator DAC table. (Refer to the **SetPowerAttenDac** group description later in this document). If the **OpenMccInterr** button is selected, the **SetPowerAttenDac** group box is updated with the default DAC table values.

UpdateAll - This button uses the parameters in the **OpenMccInterr** group box to send an **OpenMccInterr** command. In this case the antenna pair and the DAC table defaults are not displayed because the default settings will not be used. The parameters from the **SetInterrRegs** group box are used to select the antenna pair(s). Finally, the parameters from the **SetPowerAttenDac** group box are used to initialize the DAC table.

OpenMccInterr Group

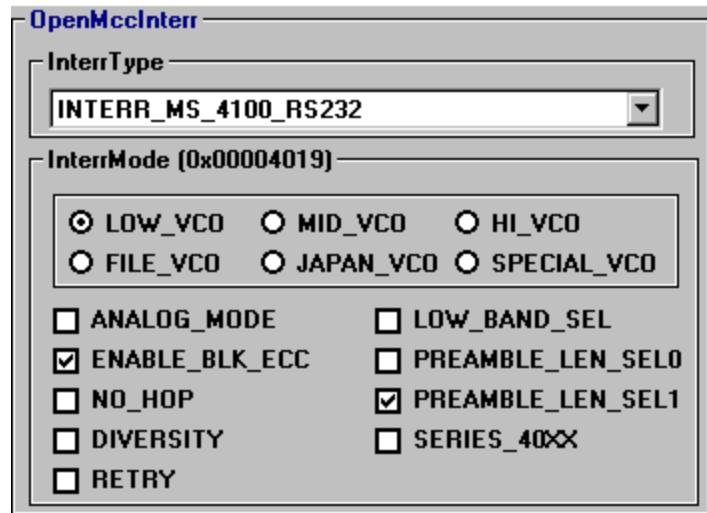


Figure 41 - OpenMccInterr Group

MsAssist default parameters, such as *Baudrate* and *Dev*, are used in place of parameters unavailable through the **Interrogator/Antenna Test** window. *InterrType* and *InterrMode* are the only parameters modifiable for the **OpenMccInterr()** function.

InterrType - A drop-down list of available interrogators is displayed

InterrMode - Listed in parenthesis is the hexadecimal value of the *InterrMode* parameter being used in the *OpenMccInterr()* function call. The available VCO options are grouped separately from the other *InterrMode* bits because only one VCO bit should be selected.

Status

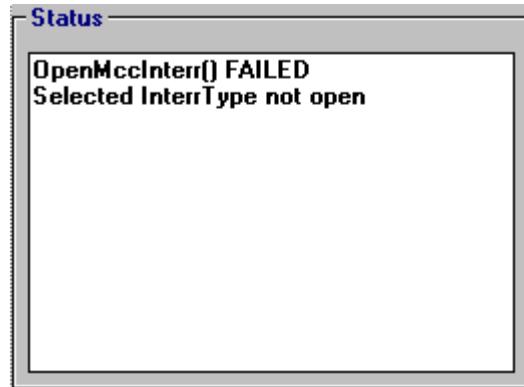


Figure 42 - Status Window

The **Status** window provides feedback to the user, such as the commands being sent, return status from the commands, and any errors, notes, or warnings that may provide additional information to the user.

SetInterrRegs

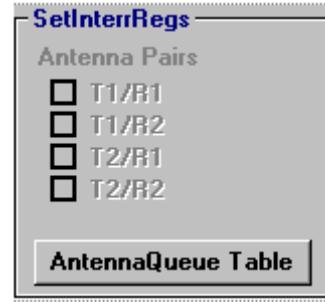


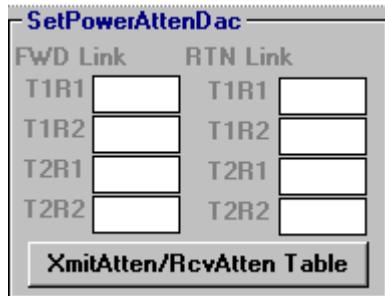
Figure 43 - SetInterrRegs Command

The *SetInterrRegs* command is used for selecting antenna pairs. The interrogator displayed in the **InterrType** edit box determines which parameters are enabled for user modification. The parameters are updated for the current interrogator when the user selects either the **SetInterrRegs** or the **Update All** button located in the **Update** group.

For the 4000 series interrogator, the selected Antenna Pairs (all antenna pairs checked) are used to determine the value of the *DiversityControl*. Selection of any combination of antenna pairs is made by checking the appropriate boxes.

The 4100 series interrogator uses the *AntennaQueue* table to determine the order in which the antenna pairs are selected. The **AntennaQueue** Table button displays the table used for updating the interrogator's *AntennaQueue*.

SetPowerAttenDac Command



SetPowerAttenDac	
FWD Link	RTN Link
T1R1	T1R1
T1R2	T1R2
T2R1	T2R1
T2R2	T2R2
XmitAtten/RcvAtten Table	

Figure 44 - SetPowerAttnDac Command

When the user selects the **SetPowerAttenDac** button in the **Update** group, the interrogator's DAC table is updated with the FWD Link and RTN Link parameters in the **SetPowerAttenDac** group.

The *SetPowerAttenDac* command is used for selecting the forward and return link power output for each antenna pair. The interrogator displayed in the **InterrType** edit box determines which parameters are enabled for user modification. The parameters are updated for the current interrogator when the user selects either the **SetPowerAttenDac** or the **Update All** button located in the **Update** group.

For the 4000 series interrogator, each of the FWD Link and RTN Link parameters are used to update the associated values for the interrogator. Each antenna pair should have an appropriate value displayed before executing the command.

The 4100 series interrogator uses the **XmitAtten** and **RcvAtten** tables to update the associated values for the interrogator. The **XmitAtten/RcvAtten** Table button displays a table used for updating the interrogator's *XmitAtten* and *RcvAtten* tables.

0.5.10.7 References (Ref) Drop-Down Menu

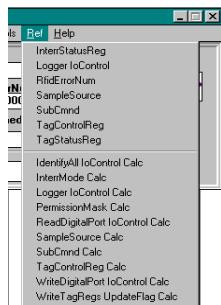


Figure 45 - References

The **Ref** drop-down menu contains bit definitions and calculators for selected MSL parameters. Some references are interrogator-specific and may not be available.

The upper half of the drop-down menu provides on-line bit descriptions for various MSL parameters. Detailed descriptions of these register parameters are located in the Bit Definitions and Assignments on page 189.

The lower portion of the menu contains bit calculators for frequently used MSL parameters. See the “MsAssist Calculators” on page 7 for a general description of the calculators. Each calculator displays the bit name, two buttons that select or de-select the bit, and the associated bit number.

0.5.10.7.1 **InterrStatusReg**

The *InterrStatusReg* function returns the interrogator status resulting from a tag or interrogator command. For each bit in the *InterrStatusReg*, the **InterrStatusReg** window lists the bit number, its hexadecimal representation within the *InterrStatusReg*, and a brief description. Detailed information about the *InterrStatusReg* is located in the InterrStatusReg Bit Assignments on page 199.

0.5.10.7.2 **RfidErrorNum**

The *RfidErrorNum* is set after each function call in the MSL. It is similar in function to the *errno* variable used in C programming. The **RfidErrorNum** window lists the possible values of *RfidErrorNum* and a brief description of the error. More *RfidErrorNum* information is located in the RfidErrorNum Definitions on page 189.

0.5.10.7.3 **TagControlReg**

The *TagControlReg* is used for tag configuration purposes. For each bit in the *TagControlReg*, the **TagControlReg** window lists the bit number, its hexadecimal representation within the *TagControlReg*, and a brief description. More information about the *TagControlReg* is located in the TagControlReg Bit Assignments on page 208.

0.5.10.7.4 **TagStatusReg**

The *TagStatusReg* returns a tag’s status. For each bit in the *TagStatusReg*, the **TagStatusReg** window lists the bit number, its hexadecimal representation within the *TagStatusReg*, and a brief description. More information about the *TagStatusReg* is located in the TagStatusReg Bit Assignments on page 203.

0.5.10.7.5 IdentifyAll IoControl Calc

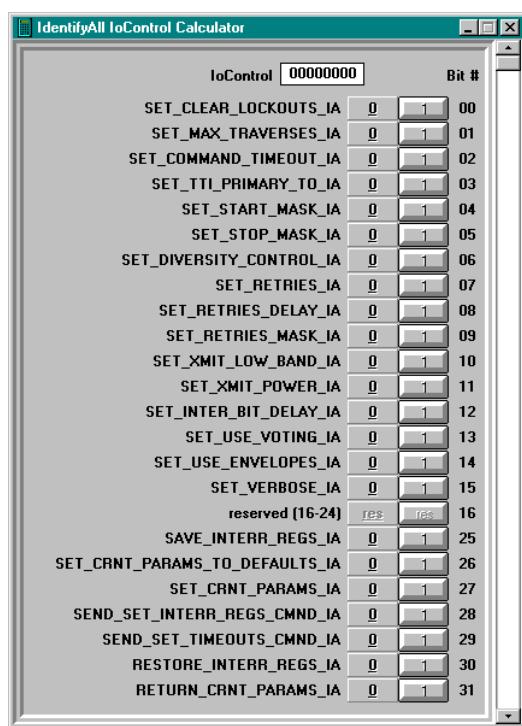


Figure 46 - IdentifyAll IoControl Calculator

0.5.10.7.6 InterrMode Calc

The *InterrMode* is a parameter used to configure certain interrogators during an open interrogator command. Refer to the specific *OpenXXX* interrogator-specific function, for detailed information.

The *IdentifyAll IoControl* parameter is used to select and configure system options during an execution of the *IdentifyAll* function. The calculated value of the *IoControl* is displayed in the **IoControl** window.

0.5.10.7.7 PermissionMask Calc

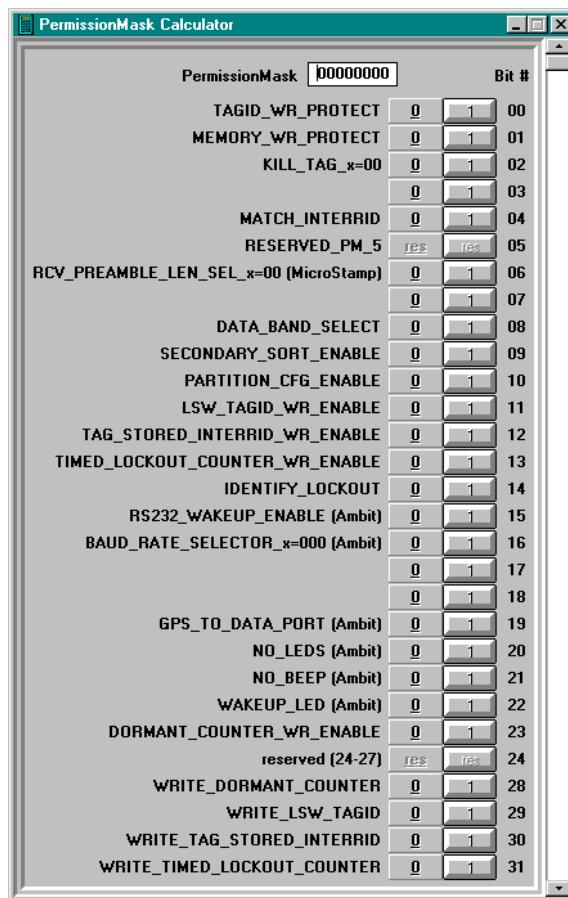


Figure 47 - PermissionMask Calculator

The **KILL_TAG_x**, **RCV_PREAMBLE_LEN_SEL_x**, and **BAUD_RATE_SELECTOR_x** bits require multiple bits to define the range of valid values.

The **KILL_TAG_x** label will change to reflect the *KillTag* bits selected. If both *KillTag* bits are selected, the **KILL_TAG_x** label will display **KILL_TAG_x=11**.

The **RCV_PREAMBLE_LEN_SEL_x** label changes to reflect the selected *RFDetInterval* bits. If both *RFDetInterval* bits are selected, the **RCV_PREAMBLE_LEN_SELx** label will display **RCV_PREAMBLE_LEN_SELx=11** (MicroStamp products only).

The **BAUD_RATE_SELECTOR_x** label will change to reflect which *BaudRateSelector* bits are selected. If none of the *BaudRateSelector* bits are selected (default value), the **BAUD_RATE_SELECTOR_x** label will display **BAUD_RATE_SELECTOR_x=000** (Ambit products only).

The *PermissionMask* parameter is found in the tag-specific functions *ReadTagStatus*, *SetMemoryPartition*, and *WriteTagRegs*. Detailed information about the *PermissionMask* is located in the Tag PermissionMask Bit Assignments on page 215.

0.5.10.7.8 ReadDigitalPort IoControl Calc

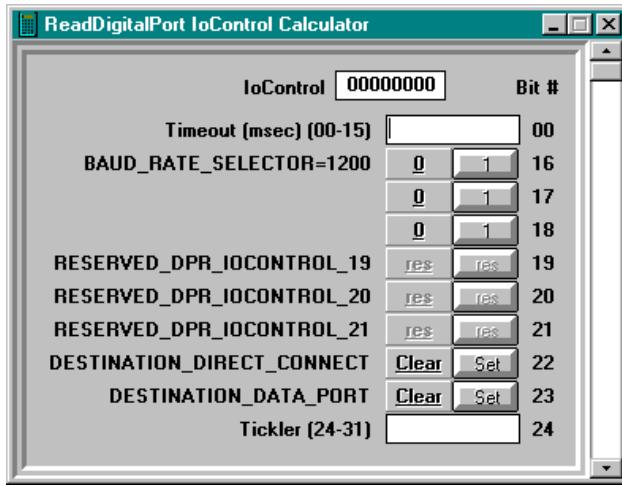


Figure 48 - ReadDigitalPort IoControl Calculator

0.5.10.7.9 SubCmnd Calc

The *SubCmnd* parameter is found in all of the tag-specific commands. More information about the *SubCmnd* is located in the Tag SubCmnd Bit Assignments on page 193.

0.5.10.7.10 TagControlReg Calc

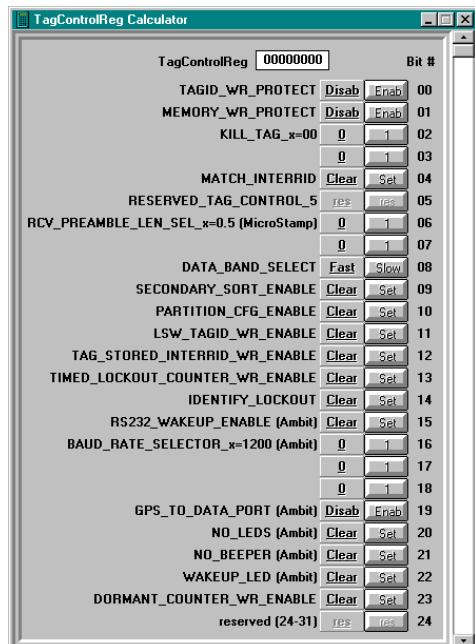


Figure 49 - TagControlReg Calculator

The **KILL_TAG_x**, **RCV_PREAMBLE_LEN_SEL_x**, and **BAUD_RATE_SELECTOR_x** bits require multiple bits to define the range

The *ReadDigitalPort IoControl* parameter is used for Ambit products only to select and configure the digital port. The calculated value of the *IoControl* is displayed in the **IoControl** window. The **Baudrate** label displays the *Baudrate* value based on bits 16-18. Only one digital port may be selected at a time. Ambit products have two RS-232 external port.

of valid values.

The **KILL_TAG_x** label will change to reflect which *KillTag* bits are selected. If both *KillTag* bits are selected, the **KILL_TAG_x** label will display **KILL_TAG_x=11**.

The **RCV_PREAMBLE_LEN_SEL_x** label changes to reflect the selected *RFDetInterval* bits and is specific to MicroStamp tags only. If both *RFDetInterval* bits are selected, the **RCV_PREAMBLE_LEN_SEL_x** label displays **RCV_PREAMBLE_LEN_SEL_x=256** (MicroStamp products only).

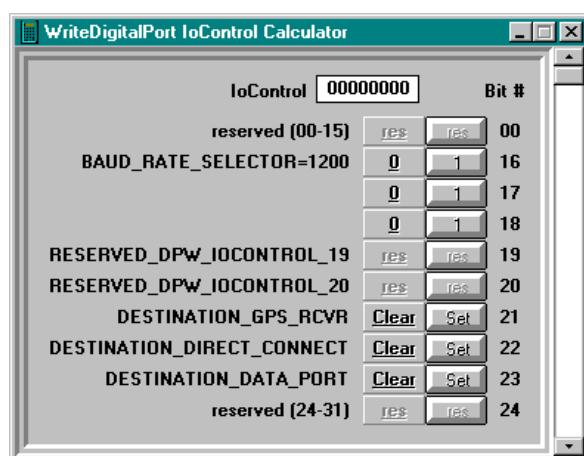
The *BaudRateSelector* in the **TagControlReg** calculator allows only one selection at a time and is specific to Ambit tags only. The **BAUD_RATE_SELECTOR_x** label changes to reflect the selected *BaudRateSelector* bits. If none of the *BaudRateSelector* bits are selected (default value), the **BAUD_RATE_SELECTOR_x** label will display **BAUD_RATE_SELECTOR_x=1200** (Ambit products only).

The following table defines *BaudRateSelector*.

DESCRIPTION	18	17	16
Baudrate = 1200	0	0	0
Baudrate = 2400	0	0	1
Baudrate = 4800	0	1	0
Baudrate = 9600	0	1	1
Baudrate = 19.2K	1	0	0
Undefined	1	0	1
Undefined	1	1	0
Undefined	1	1	1

Table 24 - *BaudRateSelector* Bit Description

0.5.10.7.11 WriteDigitalPort IoControl Calc



The *WriteDigitalPort IoControl* parameter is used for Ambit products only to select and configure the digital port. The **Baudrate** label displays the *Baudrate* value based on the selected bits. Only one digital port may be selected at a time. Ambit products have two RS-232 external ports, and MicroStamp products have one synchronous port.

Figure 50 - WriteDigitalPort IoControl Calculator

0.5.10.7.12 Write TagRegs UpdateFlag Calc

The *WriteTagRegs UpdateFlag* parameter is used in the tag-specific commands *WriteTagRegs*, *WriteTagRegsRandIdRange*, and *WriteTagRegsTagIdRange*. Detailed information about the *WriteTagRegs UpdateFlag* is located in the *WriteTagRegs* on page 176.

The **KILL_TAG_x**, **RCV_PREAMBLE_LEN_SEL_x**, and **BAUD_RATE_SELECTOR_x** bits require multiple bits to define the range of valid values.

The **KILL_TAG_x** label changes to reflect the selected *KillTag* bits. If both *KillTag* bits are selected, the **KILL_TAG_x** label displays **KILL_TAG_x=11**.

The **RCV_PREAMBLE_LEN_SEL_x** label changes to reflect the selected *RFDetInterval* bits and is specific to MicroStamp tags only. If both *RFDetInterval* bits are selected, the **RCV_PREAMBLE_LEN_SEL_x** label displays **RCV_PREAMBLE_LEN_SEL_x=11** (MicroStamp products only).

The **BAUD_RATE_SELECTOR_x** label changes to reflect the selected *BaudRateSelector* bits and is specific to Ambit tags only. If none of the *BaudRateSelector* bits are selected (default value), the **BAUD_RATE_SELECTOR_x** label displays **BAUD_RATE_SELECTOR_x=000** (Ambit products only).

0.5.10.8 Help Drop-Down Menu

The **Help** drop-down menu provides information about MsAssist.

0.6 MsAssist.ini

The MicroStamp Assist program, MsAssist.exe, creates an initialization file (MsAssist.ini), which is stored in the Windows directory. Information such as interrogator parameters and system parameters are saved in the initialization file. If MsAssist cannot locate the MsAssist.ini file, the application creates one.

The MsAssist.ini file is used to initialize certain global parameters when MsAssist is executed. The [Interrogator] section identifies the last interrogator (*InterrType*) successfully opened and saved by MsAssist. All parameters in the **Interrogator** window may be saved to the MsAssist.ini after successfully opening an interrogator, if the **Save** button is selected. The following section is a sample taken from the INI file after interrogator parameters have been saved.

```
[Interrogator]
Type=INTERR_MS_4100_RS232
```

```
[InterrogatorParam]
UnitNumber=0
InternetAddress=
Baudrate=38400
IoBase=378
```

```
UnitNum=0
InterrMode=00004C2A
DevName=COM1
StopBits=2
Parity=NO_PARITY
UseRts=0
CountryCode=CC_USA
```

```
[PhoneNumber AddItem]
Count= 0
```

The [CommandDefaults] section stores parameters from the **Setup Tag Cmnd Parameters** window. Selecting the **OK** button stores these parameters to the INI file.

```
[CommandDefaults]
Security=00000000
TagId=FFFFFFFFFFFFFF
InterrId=0C0D0E0F
AccessId=00000000
TagControlReg=00000000
PermissionMask=00000000
SubCmnd=00E00000
```

The **XmitAtten/RcvAtten Command Parameters** window may be accessed through the **SetInterrRegs**, **SetPowerAttenDac**, and **Interrogator/Antenna Test** windows. Values may be read or saved to the MsAssist.ini file by selecting the **Read from INI file** or **Save to INI file** buttons. In addition, on start up of MsAssist, if the **MsAssist INI file** radio button is selected in the **Setup - Interr Cmnd Parameter - Interr-Specific Parameters** window, the attenuation values initialize MsAssist's attenuation tables from the MsAssist.ini file. A SetInterrRegs or SetPowerAttenDac function must be used to update an interrogator's attenuation and antenna tables.

```
[XmitAtten/RcvAtten]
XmitRcvAttnR1= 28 28 28 28 28 28 28 28 28 28 28 28 28
XmitRcvAttnR2= 28 28 28 28 28 28 28 28 28 28 28 28 28
XmitRcvAttnR3= 28 28 28 28 28 28 28 28 28 28 28 28 28
XmitRcvAttnR4= 28 28 28 28 28 28 28 28 28 28 28 28 28
XmitRcvAttnR5= 28 28 28 28 28 28 28 28 28 28 28 28 28
XmitRcvAttnR6= 28 28 28 28 28 28 28 28 28 28 28 28 28
```

The **AntennaQueue Command Parameters** window may be accessed through the **SetInterrRegs** and **Interrogator/Antenna Test** windows. Values may be read or saved to the MsAssist.ini file by selecting the **Read from INI file** or **Save to INI file** radio buttons. In addition, on start up of MsAssist, if the **MsAssist INI file** button is selected in the **Setup - Interr Cmnd Parameter - Interr-Specific Parameters** window, these antenna queue values are read from the MsAssist.ini file.

```
[AntennaQueue]
AntennaQueueR1= 1 2 7 8 0 6
AntennaQueueR2= 7 8 9 10 11 12
AntennaQueueR3= 13 14 15 16 17 18
AntennaQueueR4= 19 20 21 22 23 24
AntennaQueueR5= 25 26 27 28 29 30
AntennaQueueR6= 31 32 33 34 35 36
```

The [SystemDefaults] section stores parameters from the **Setup System Parameters** window. Selecting the **OK** button stores the *MacroDelay* parameter to the INI file.

```
[SystemDefaults]
MacroDelay=0
ReleaseInfo=9805295005001234
```

The [LogDir] section is used during a **Macro Play/Record** session to identify the logging directory. This value is saved from the **Directory Selection** window.

```
[LogDir]
Directory=C:\
```

The [Notes] section provides additional information to the user and is not used by MsAssist.

```
[Notes]
Type=INTERR_MS_4100_RS232
Type=INTERR_MS_4100_RS485
Type=INTERR_MS_4100_RS422
Type=INTERR_MS_EPP
Type=INTERR_MS_RS232
Type=INTERR_MS_RS485
Type=INTERR_MS_RS422

SubCmnd(MicroStamp RF)=00E00000
SubCmnd(MicroStamp DTS)=00C00000

InterrMode(4000 EPP series)=80004C2A
InterrMode(Japan 4000 EPP series)=80044C0A
InterrMode(other series)=00004C2A
InterrMode(other Japan series)=00044C0A
```