

# Optimizing Reader Settings

## Version 1.3

Jim Peternel  
Intermec Technologies  
June 24, 2009

### Notes

- All uses cases are for fixed readers unless otherwise specified.
- All read methods shown below are for optimal performance for the use case.
- Often during testing you want to see how many times each tag is read as it passes through the read zone. When applicable, alternative testing read methods are included for the uses cases.

### Reader's Internal Tag List

- When a tag is read, the reader adds it to its tag list. It will also return the EPC ID to you.
- Whenever you issue a new read command to the reader it clears its internal tag list.
- The READ POLL command which is used with REPORT=NO mode also clears the internal tag list.

### Reading Large Memory Blocks (2 examples):

The following two examples explain how to not read and how to read large data fields. Please do not forget that not all tags have user memory.

#### Example One: READ HEX(3:0,64)

- Will read 64 bytes from user memory (not all tags have user memory).
- Intermec readers split up long reads into multiple reads of up to 12 bytes.
- For the above example the reader protocol stack will do 5 reads of 12 bytes and one read of 4 bytes.
- It is these low-level reads to which a retry (attrib READTRIES) is applied so an error in the last read block does not result in a retry of all blocks - just the last one that caused an error.
- **RDERR OK**
  - Since only one long field is defined the reader will return RDERR for the entire block.
  - You have to reread all 64 bytes even though the reader may have read 99% of the data.

#### Example Two: READ HEX(3:0,12) HEX(3:12,12) HEX(3:24,12) HEX(3:32,12) HEX(3:0,12)

- It is better to define the large read field as a group of smaller fields. The advantage to do this versus one large read field is that if any of

these smaller fields fails then you only need to reissue a read command for that one small field not all of 64 bytes.

- Assuming only field 2 fails, here is the readers response
  - <EPCID> <DATA> RDERR <DATA> <DATA> <DATA> OK

### **Read/Write Command Still Fails, Why?**

- The FCC regulations define a 400ms dwell time per channel. After 400ms the reader must stop and hop channels.
- If you are in the process of reading or writing a tag when this dwell time limit is reached, the reader will return a RDERR or WRERR for that tag.
- If you are experiencing a large number of RDERR or WRERR then perhaps your command is trying to read/write too much data and its taking more than 400ms. Do not assume that the tag is singulate at the very start of the 400ms timer. For example, the tag may be singulate at 300ms into the dwell time and so you only have 100ms left to finish the read or write.
- If this is the issue, you will need to be prepared to resend the read or write command for the fields that failed. Using smaller field definitions will allow you to issue a much smaller follow up read/write command which focuses only on the fields that failed.

### **How Tags Are Singulated**

The actual singulation process is complex so I will simplify it here to help explain the overall process. Let us assume the following attribute settings:

- ATTRIB IDTRIES=1
- ATTRIB ANTTRIES=1
- ATTRIB INITIALQ=4
- ATTRIB ANTS=1,2,3,4

We will also assume only one tag is expected to be read. Here is what the reader will do when using the above attribute settings:

1. Reader sends reset command on each antenna. This will reset any tags in the RF field. All tags reset when they first energize so this is not a critical command for tags to hear.
  2. Reader sends the read command which includes the value of *ATTRIB INITIALQ* which tells the tags range of their random number generator (1-16 in this case).
  3. The tag will generate a random number in the range of 1 to 16. Let us assume the tag generates a random number of 15.
  4. Reader starts at slot 1 and asks if there are any tags there. It will then continue rolling through the slots until a collision occurs. For our example the reader will attempt to find a tag in 14 empty slots before it finds the single tag in the field. It will then waste more time checking for a tag in slot 16.
- A collision occurs when 2 or more tags generate the same random number. When this occurs the reader tells all tags not yet read to increase the Q

value by one and regenerate their random numbers. The reader then starts over at slot 1.

- To optimize reading speed you want to minimize the number of empty slots the reader has to check. So if you know you will always be reading a single tag, then you want to set ATTRIB *INITIALQ* =0 or 1.
- In testing I have not seen a case yet where exceeding an *INITIALQ* value of five will improve read rates.

## Read Methods

- READ REPORT=DIRECT
  - Default mode
  - Same as sending READ
  - Worst method for reading tags.
  - Very inefficient use of the RF.
- READ REPORT=EVENT
  - Continuous mode. The RF will not turn off until a READ STOP is sent.
  - READ STOP turns off the RF and also clears the tag list in the reader.
  - Returns each EPC ID one time only.
  - If a tag remains in the read zone the reader may continue reading it but will not return the EPC ID to you.
  - You can also send a READ POLL while the reader is running to clear the tag list. This will allow tags that have already been read to be reported again.
  - If you send a READ STOP and then restart the read all tags in the RF field will be reported again.
- READ REPORT=NO
  - Continuous mode. The RF will not turn off until a READ STOP is sent.
  - READ STOP turns off the RF and also clears the tag list in the reader.
  - Reader will store the tags in its internal read list until the user requests them. If you requested information such as ANT, COUNT, TIME, or a field the reader will also add that information to the tag list.
  - READ POLL command tells the reader to return the tags list to the user. It will then clear the list.
  - If a tag remains in the read zone the reader may continue reading it and will add the tag to the list or update its information such as COUNT.
- READ REPORT=EVENTALL
  - Continuous mode. The RF will not turn off until a READ STOP is sent.
  - READ STOP turns off the RF and also clears the tag list in the reader.
  - Returns each EPC ID every time its read.
  - WARNING: this mode can overwhelm your communication link. Depending on the settings, the reader could flood your communication link with tag data.
  - This method is best used with a small numbers of tags and for test purposes only.

## Important Attribute Descriptions

- ATTRIB SCHEDOPT
  - Valid values: 0-1
  - ATTRIB SCHEDOPT=1 the reader will automatically try to use timeouts. If you set IDTIMEOUT to zero the reader will then use IDTRIES.
  - If you set ANTTIMEOUT to zero the reader will then use ANTTRIES.
  - ATTRIB SCHEDOPT=0 the reader will use what ever attributes are currently active. You can use ATTRIB TIMEOUTMODE to determine which are active (IDTRIES/ANTTRIES or IDTIMEOUT/ANTTIMEOUT).
- ATTRIB TIMEOUTMODE
  - Valid values: ON, OFF
  - Get/Set which set of attributes are active or should be active.
  - ATTRIB SCHEDOPT=1 will disables TIMEOUTMODE.
  - ATTRIB TIMEOUTMODE=OFF will enable IDTRIES and ANTTRIES.
  - ATTRIB TIMEOUTMODE=ON will enable IDTIMEOUT and ANTTIMEOUT.
  - If ATTRIB TIMEOUTMODE=ON and IDTIMEOUT=0 then the reader will use IDTRIES.
  - If ATTRIB TIMEOUTMODE=ON and ANTTIMEOUT=0 the reader will use ANTTRIES.
- ATTRIB INITTRIES should always be set to one.
- ATTRIB SESSION should be set to ONE for all use cases except for use case one listed below.
- ATTRIB ANTS enables antenna ports
  - ATTRIB ANTS=1,2,3,4 turns on all four antennas
  - ATTRIB ANTS=2,4 turns on antennas 2 and 4
  - ATTRIB ANTS=3 turns on antenna 3.
- ATTRIB INITIALQ
  - Valid values: 1-255
  - I do not recommend exceeding a value of five.
  - This attribute determines how many empty slots the reader creates for the tags to respond in. If you create too many empty slots the reader will waste time trying to find tags in the empty slots instead of reading tags.
  - One mistake people make is to assume that since they are reading a large number of tags they need to set this value to a big number (>5). However the problem is that not all of the tags enter the read zone at the exact same moment. They will also not all be visible to the antenna that is on at that moment. Typically on a large pallet of goods, only a small number of tags will be visible at any given time. As the pallet moves through the read zone more and more tags will enter however they will be read as they enter so the total number of tags waiting to be read at any given time will not be anywhere near the total number of tags on the pallet. Also if you set your attributes so that you are not resetting tags after they have been read (optimal performance) the tags that have been read will not respond again while passing through the read zone.

## Case One: Single Tag Speed Reading Test

You want to test a single tag passing a single antenna. The goal of the test is to see how many times the tag can be read. This is a performance test and not a real use case. When setting up an RFID station its good to determine how many times a tag can be read which helps you determine how reliable the station will work. For example, if you can consistently read a single tag 5 or more times then you can feel very certain that your read station will be very reliable.

- ATTRIB SESSION=0 (this is the only use case for setting it to zero)
- ATTRIB ANTS=1
- ATTRIB SCHEDOPT=0
- ATTRIB IDTRIES=1
- ATTRIB ANTTRIES=1
- ATTRIB INITIALQ=0
- READ COUNT REPORT=NO
  - Set the polling interval to a value greater than the amount of time the tag will be in the RF zone.

## Case Two: Portal

You want to pass a pallet of tags through a portal using four antennas. The key to reading all of the tags is to make sure the reader does not reset any tags while they pass through the portal. To do this, you need to determine how long the pallet will be in the read zone. Then set the IDTIMEOUT to a value greater than that. For example, let us assume that the pallet will be in the read zone for three seconds. I would set IDTIMEOUT=4000 (4 seconds). I want to force the reader to cycle antennas as fast as possible so I would set ANTTIMEOUT=0 which forces the reader to use ANTTRIES. I would then set ANTTRIES=1. So for this use case I would recommend the following settings:

- ATTRIB SESSION=1
- ATTRIB ANTS=1
- ATTRIB SCHEDOPT=1
- ATTRIB IDTIMEOUT=4000
- ATTRIB ANTTIMEOUT=0 (forces reader to use ANTTRIES)
- ATTRIB ANTTRIES=1
- ATTRIB INITIALQ=4 or 5 (play with this setting but do not make it too high)
- For the IF61 reader use:
  - READ REPORT=EVENT
    - Each EPC will be returned one time only.
    - Count is always one so it is excluded.
- For all other readers:
  - READ COUNT REPORT=NO
    - When using other fixed readers or handhelds its best to use this read method. Set the polling interval to a value greater than the amount of time the tag will be in the RF zone. This will prevent tags from being reset while in the read zone. For this example set it to 5 seconds.

- You can also use this mode with the IF61 if you want to see how many times you read each tag. Please note that with the IDTIMEOUT=4000, the tags will not be reset after they are read so most tags will only be read once. You can try setting IDTIMEOUT to a lower value however this could reduce read performance.

### **Case Three: Conveyor One Tag Per Box**

You want to pass a box with ONE tag attached through a read zone on a conveyor. For this example we will assume there are three antennas. The key to reading the tag is to make sure you are cycling through all the antennas as quickly as possible. This is because you will not know which antenna the tag will be visible to. To do this you need spend as little time as possible on each antenna. To force the reader to cycle antennas as fast as possible I would set ANTTIMEOUT=0 which forces the reader to use ANTTRIES. I would then set ANTTRIES=1. You also want to disable IDTIMEOUTS and use IDTRIES. So for this use case I would recommend the following settings:

- ATTRIB SESSION=1
- ATTRIB ANTS=1
- ATTRIB SCHEDOPT=0
- ATTRIB IDTIMEOUT=0 (forces reader to use IDTRIES)
- ATTRIB ANTTIMEOUT=0 (forces reader to use ANTTRIES)
- ATTRIB IDTRIES=1
- ATTRIB ANTTRIES=1
- ATTRIB INITIALQ=0 or 1
- For the IF61 reader use:
  - READ ANT REPORT=EVENT
    - Each EPC will be returned one time only.
    - Count is always one so it is excluded.
- For all other readers
  - READ ANT COUNT REPORT=NO
    - Use this method with other readers.
    - Can be used with any reader (IF61 included) to see how many times the tag was read while passing through the read zone.

### **Case Four: Conveyor Multiple Tags Per Box**

You want to pass a box with multiple tags attached through a read zone on a conveyor. For this example we will assume there are three antennas. This is a tricky use case. You need to read all the tags however you need to cycle through the antennas as quickly as possible. For this use case I would recommend the following settings:

- ATTRIB SESSION=1
- ATTRIB ANTS=1
- ATTRIB SCHEDOPT=1
- ATTRIB IDTIMEOUT=1000
- ATTRIB ANTTIMEOUT=0 (forces reader to use ANTTRIES)
- ATTRIB ANTTRIES=1

- ATTRIB INITIALQ=2 to 4 (depends on how many tags there are, you will need to play with this setting)
- For the IF61 reader use:
  - READ ANT REPORT=EVENT
    - Each EPC will be returned one time only.
    - Count is always one so it is excluded.
- For all other readers
  - READ ANT COUNT REPORT=NO
    - Use this method with other readers.
    - Can be used with any reader (IF61 included) to see how many times the tag was read while passing through the read zone.