## Potential Improvement Opportunities

1. New system parameter for whther RTCIS expects two messages (i.e. two A14s or two A23) for stacked pallets from ACTIV and only one from all other systems instead of keying off location name.
2. Should actsignon just send the slot and not slot+level

## Stacked Pallet Induction

The stacked pallet induction sequence should be roughly the same as RTCIS 7.4 (and earlier).  After the 5 message is received, RTCIS does perform some additional checks on stacked pallets that may cause them to be rejected.  Specifically:

* If the pallets are not the same item, the stack will be rejected.
* If any pallet in the stack has a non-zero UL stat, then check
  + If the "Store Partial Pallets in ASRS?" pallets is no, then reject the stack (I am not sure why we are equating stacks and partials, but only when the UL status is set)
  + If any of the associated non-zero UL Statuses has the reject flag, then reject the stack

After it passes these checks, RTCIS will issue an A8 for each pallet in the stack.  In the old version of RAI, RTCIS would execute the library routine for each pallet in the stack (normally twice for stacked pallets).  So the ASRS is responding with a location for the first pallet in the stack before it receives the second pallet in the stack.  After the A8 response for the last pallet in the stack is received, RTCIS will send 6 back to the PLC.

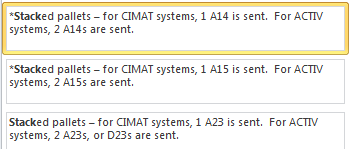
In the RAIng version, RTCIS will issue the A8 messages in the same sequence when the “A8 Loc Mode (RAIng): 1-Loc Assigned Later, 2-Wait For ASRS Loc” system parameter is setting to 2 (wait). If the system parameter is set to 1 (assign later), RTCIS will now potentially send both A8 messages before receiving the ASRS response for either.

FYI- Were you able to test this sequence (for both settings of the system parameter)?  I suspect that I was (mistakenly) sending two six messages when the system parameter was set to 1 (assign later). In fact, I am so sure of this that I put in a fix for it tonight (so please bounce the dtlmsgdrv before testing again).

## Stacked Pallets and Staging

Question

I’ve successfully sent an A14 for the base\_ulid of a stacked pallet and RTCIS properly put both the base and the non-base into the VTL location.  I’m debating whether to test further the A14, A15, and A23.  My question is, does RTCIS have any different logic for the “2 A14” scenario or does it work like the “1 A14” scenario.  If they are different, how are the 2 scenarios configured?  Is the logic hardcoded by ASRS location name system name?



Answer via Aaron

This one is really a bit of a kludge.   RTCIS expects two messages (i.e. two A14s or two A23) for stacked pallets from ACTIV and only one from all other systems.  To enforce this, we check to see if the system parameter ASRS location = ‘ACTIV’.  If it is, it processes just the ulid in the message.  If it is not, we process the entire stack (assuming we will get only one message per stack).

So, if you want to test this, you will need to setup another test instance with an ASRS location called something other than ‘ACTIV’.

Note: I am not suggesting that we should change this, but I think it would make a lot more sense to base this logic on a system parameter, rather than keying on the ASRS location name.

## actsignon

You are correct that RTCIS is doing this inconsistently.  I am sure that you are aware of the logic that we have in RTCIS for RAI that will take the base slot from the ASRS and attempt to find the lowest open level, matching the base slot.  So “400” turns into “400A”.

In the RDT, we are calling the signoff with the ACTIV slot location, being 400 in your example. We are saving the ASRS slot in USERS.ACTIV\_STGLOC, and then echoing it back to the ASRS in the D32.   In actsignon, we are calling the signoff with the RTCIS location (normally named as slot + level) that we find by selecting the RTCIS location using the slot matching the data on screen, being 400A in you example.

As far as I can tell, it has always been this way.  I would also guess that Lima is not using actsignon to signoff slots very much.  I don’t even know if the ASRS will accept the message with the level appended, or the ASRS is rejecting the message.

An ‘A’ is appended to the entered Slot ID for Sign On or Sign Off

Task: ACTSIGNON 8.0.0     EUS RTCIS 8.0 BDLG3256 tst09         05-Mar-2015 13:08

                             ACTIV Signon - TT6855

Subsite:      DE08

Slot ID:          400                Sign: ON

Control No:

Signon to 400A successful - Press ENTER to Continue

## Integrator System Ports via /etc/services

A change has been implemented for storing the inbound listener ports used with most external systems in Integrator (e.g., CSI).  An algorithm will now be used to look up the port in /etc/services.

# Integrator Systems

rtcis-int-devel3-csi         5536/tcp          #RTCIS Integrator for CSI interface for Development

rtcis-int-test3-csi          5546/tcp          #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst01-csi          55461/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst02-csi          55462/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst03-csi          55463/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst04-csi          55464/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst05-csi          55465/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst06-csi          55466/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst07-csi          55467/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst08-csi          55468/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst09-csi          55469/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-tst10-csi          55460/tcp         #RTCIS Integrator for CSI interface for Testing

rtcis-int-train3-csi         5526/tcp          #RTCIS Integrator for CSI interface for Training

rtcis-int-live3-csi          5516/tcp          #RTCIS Integrator for CSI interface for Production

# Integrator PRIMEDC Systems

rtcis-int-live3-primedc      5816/tcp          #RTCIS Integrator for PRIMEDC interface for Production

rtcis-int-train3-primedc     5826/tcp          #RTCIS Integrator for PRIMEDC interface for Training

rtcis-int-devel3-primedc     5836/tcp          #RTCIS Integrator for PRIMEDC interface for Development

rtcis-int-test3-primedc      4552/tcp          #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst01-primedc      58461/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst02-primedc      58462/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst03-primedc      58463/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst04-primedc      58464/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst05-primedc      58465/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst06-primedc      58466/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst07-primedc      58467/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst08-primedc      58468/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst09-primedc      58469/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

rtcis-int-tst10-primedc      58460/tcp         #RTCIS Integrator for PRIMEDC interface for Testing

Sites Using CSI

*select act\_sys\_id, snd\_hostname, snd\_port\_num, sys\_id, client\_id, user\_id from rtcis\_session where req\_enabled = 'Y';*

Site ID SND\_HOSTNAME SND\_PORT\_NUM USER\_ID ACTDAT

aalx1 CSI 192.168.113.10 5547 <WCS> 04-MAR-15

celx1 CSI 143.21.72.128 5547 <WCS> 04-MAR-15

eulx3 CSI 143.21.220.14 5547 <WCS> 04-MAR-15

gglx1 CSI 143.21.97.216 5649 <WCS> 04-MAR-15

hulx2 CSI 143.21.92.71 5547 <WCS> 04-MAR-15

hulx2 CSI <WCS> 16-FEB-10

limlx3 CSI <WCS> 07-DEC-11

limlx3 CSI <E80> 31-JUL-12

melx1 CSI 143.26.71.56 5548 <WCS> 14-JAN-13

sealx1 CSI 143.40.205.217 6666 <WCS> 06-AUG-14

sealx1 CSI 155.128.123.10 5547 <WCS> 03-MAR-15

Port Troubleshooting

TST08INT SQL>select \* from SL\_SYS\_COMM\_VAL where COMM\_MTHD\_ATTR = 'PORTNUM' and sys\_id = 'CSI';

SYS\_ID                         COMM\_ COMM\_ COMM\_MTHD\_ID                   COMM\_MTHD\_ATTR

------------------------------ ----- ----- ------------------------------ ------------------------------

COMM\_MTHD\_ATTR\_VAL

------------------------------------------------------------------------------------------------------------------------------------

ALG\_ID                         KW\_ID                                 U\_VERSION INS\_DT              LAST\_UPD\_DT

------------------------------ ------------------------------ ---------------- ------------------- -------------------

INS\_USER\_ID                              LAST\_UPD\_USER\_ID

---------------------------------------- ----------------------------------------

CSI                            ASYNC I     SL\_INB\_SOCKET                  PORTNUM

VC\_GET\_LISTEN\_PORT                                                          53 01/26/2015 15:12:38 02/24/2015 04:48:28

SLIMP                                    BD2333

# netstat -tupln |grep 5546  
tcp 0 0 0.0.0.0:5546 0.0.0.0:\* LISTEN 17339/mocasrvprc ts

live3 17339 17322 0 05:58 ? 00:01:04 mocasrvprc tst08int/29140 -P 5 -p 29140 -x 1 -s /opt/redprairie/pngint1/les/test3/log/MOCA.sts

## Timeout Parameters

The web has a timeout response parameter in the web config file /var/www/cgi-bin/rtciscgi.config.  The timeout parameter in this file is the time that the CGI will wait for a response from the Webserver (and, in turn, the RDT control process) before giving up.  On bdlg3256, this is set to 30 seconds.

The RAIng interface also has some timeout system parameters, such as the one being used by the trailer move, ASRS\_A32\_TO – “ASRS: A13 Loc Timeout (in seconds). Used when A13 mode=2 (Wait)”.  This parameter tells RTCIS how long to wait for the ASRS response before giving up.  This is also set to 30 seconds on bdlg3256.

Because RAID was not running in your example, both timers expired at roughly the same time, with the Web timer expiring slightly before the RAIng timer (since the Web time was set first). Obviously, this could happen at a production site, if the ASRS failed to respond in time or if there were some sort of communication/integrator error.  Rather than try to add a lot of code to fix this, I would suggest just requiring the Web timer setting to be a couple of seconds more than the largest ASRS timeout value.

## Message 35

Re: Request with No ULS

I traced the old code far enough to determine we can send a request with a zero line count in the existing interface.   However…..I did not find any real examples of this in production.  Lima has message history data dating back to 1/28 and there are no examples where RTCIS submitted a request for an empty trailer.  Note an empty trailer request would have msghst.trkuls = 0 for a msghst.msgtyp  in (‘A35’, ‘M25’).  So I don’t think this is every happening operationally.

*Note: After seeing the production data, I double checked the code again, and I still could not find anything preventing us from sending an A35 with a line count of zero.  I am positive that we send a transaction for an empty trailer in 8.0 for both the old and the new interface.  And I don’t see any change that would have altered the 8.0 behavior from the 7.3/7.4 behavior.  But I don’t have access to a 7.3 or 7.4 server to test with the real real RAI libraries. I could try this on bdln2372 or bdln2404 with the stub libraries, but I am not sure that this is a good comparison for production.*

But, I would still agree that it makes more sense to prevent sending request with zero ULs to the ASRS (i.e. why would the automation ever care about an empty trailer?).

## Msg8 and New Parameters

3.       What should we log in the msghst table for the A8 when this system parameter is set to 1 (assign later)?  We will not have the actual ASRS location when the msghst row is being added.  Currently, the “9999” is used to indicate that.  (Note: The bug is causing M8 to be treated like an A8, so the M8 should log the real location when fixed.  But with the A8, we don’t have the location……yet).

I am not sure if I explained the “ASRS: A8 Loc Mode (RAIng): 1-Loc Assigned Later, 2-Wait For ASRS Loc” in detail before.  I have two options for the message 8 in the new interface.  The first option is to have the original RTCIS application (being the dtlmsgdrv) wait for the ASRS response via the integrator.  This would be the preferred setting, assuming we believe the XML/socket interface to the ASRS will perform adequately.  I prefer this setting because the dtlmsgdrv can make better decisions immediately for error processing, such as directing the UL to reject, if needed.

The second option is the delayed option, if we believe the new XML/socket interface cannot keep up.  In this mode, we send the message to ASRS and the original RTCIS process (dtlmsgdrv) continues without waiting for a response.  The UL is still pended to the ASRS location (“ACTIV”, “CIMAT”, etc) but there is no error processing/checking performed.  When the integrator (eventually) receives a response, the message 6 is then sent to the PLCs and (some) error processing/checking is performed.  Note there are some differences, due to the timing.  If the ASRS fails to respond (or responds late), it is the PLC’s responsibility to send the UL to reject.  In other words, if the UL reaches a (critical) divert point, the PLC must direct it to reject without a message from RTCIS.  This is probably reasonable and I expect most PLC systems to do this, but RTCIS will not have any record of the UL being directed to reject.