|  |
| --- |
| Predictive Health Notification  Alinity I Process Path Jams 5758 |
| ALINITY ci Immunoassay Analyzer  Anthony Schuler  June 25, 2018 |

**PHN - Alinity IA Process Path Jams 5758 Spec Sheet for Apollo**

**References**

APLM D000041546/A Prognostic Health Notification for Alinity I Process Path jams based on 5758 errors

**Summary**

To implement a Prognostic Health Notification (PHN) for Alinity-i analyzers that will detect a buildup of process path retries (AimCode: 5758) before the customer begins experiencing process path jams (AIM CODE: 5752 or 5757).

|  |  |
| --- | --- |
| PHN Descriptor | Alinity IA Process Path Jams 2 |
| PHN Experience Code / Name | CCG1 PHN\_Alinity\_IA; PHN\_Process Path\_IA; Jam Recovery  CCI1 PHN\_Alinity\_IA; PHN\_Pretreatment Path\_IA; Jam Recovery |
| PHN KM Article Number/ Name | KM1158157 PHN\_IA: Process Path Jam Recovery |
| Service Level | 2- Advanced Service |
| Always On Package | Always On 2 01DP5-02 |
| IDA Table | IDAQOWNER.ICQ\_MESSAGEHISTORY,  IDAQOWNER.ICQ\_RESULTS |
| IDA Table Fields | From IDAQOWNER.ICQ\_MESSAGEHISTORY:  MODULESN, LOGDATE\_LOCAL, AIMCODE, AIMSUBCODE  From IDAQOWNER.ICQ\_RESULTS:  MODULESN, LOGDATE\_LOCAL, CORRECTEDCOUNT |
| Analysis Frequency | Daily |
| Data Required | Previous 1 day |
| Data Aggregation | None |
| Run Time Estimate | 7 seconds (2 months of data from each table, 8 algorithm flags) |
| Flag Criteria | If the number of 5758 process path errors is greater than or equal to 4 and the number of CORRECTEDCOUNT not null is greater than or equal to 10, then flag. |
| Probable Failure Modes | Process Path experiences too many consecutive retries. This can occur during system initialization, maintenance, or test processing. This could be to buildup of salt on Process Path, RV jams, or process path move failures. |
| Suppression Experience Codes | Z75L: 5752 Process path move error on (0) lane  Z75P: 5757 Process path jam recoveries on (0) excessive  Z75Q: 5758 Process path jam recovery successful on (0). Process path may require servicing  A3G1: Process Path Issue, Process Path Jams/Stuck  A3G7: Process Path Issue, Process Path Movement Not as Expected/No Movement  B5G7: Robotics Issue: Process Path/Supply Center Process Path Movement Not as Expected/No Movement |
| Applicable Work Done Codes (WDC) | DF\*\*: Transport, Robotics, Waste Disposal & Other Hardware, Process Path |

**Data Processing Steps**

|  |  |
| --- | --- |
| Data Processing Steps | |
| 1 | Query all data from the previous day for each process path/instrument from both database tables (IDA Table: IDAQOWNER.ICQ\_MESSAGEHISTORY and IDAQOWNER.ICQ\_RESULTS). |
| 2 | Unique process paths will be identified by MODULESN. |
| 3 | Unique process path locations (i.e. process path vs. pretreatment path) will be identified by MODULESN and AIMSUBCODE. |
| 4 | Exclude data where AIMCODE is not equal to 5758 from the IDAQOWNER.ICQ\_MESSAGEHISTORY table. |
| 5 | Exclude data where AIMSUBCODE is not equal to D298 or D299 from the IDAQOWNER.ICQ\_MESSAGEHISTORY table. |
| 6 | Summarize the data by summing the number of data points where CORRECTECOUNT is not null from the IDAQOWNER.ICQ\_RESULTS table, by MODULESN. |
| 7 | Summarize the data by summing the number of data points where AIMSUBCODE is equal to D298 or D299 from the IDAQOWNER.ICQ\_MESSAGEHISTORY table, by AIMSUBCODE and MODULESN. |
| 8 | Summarize the data by summing the number of data points where CORRECTEDCOUNT is not null from the IDAQOWNER.ICQ\_RESULTS table, by MODULESN. |
| 9 | Flag any process path/instrument combination where the number of process path errors (i.e. AIMCODE = 5758) is greater than or equal to 4 and the number of CORRECTEDCOUNT not null is greater than or equal to 10. Note that AIMSUBCODE will be used to identify process path vs. pretreatment path. |

**Define Reusable Routine**

**Note:** Make sure that reusable routine**,** Alinity IA Process Path Jams – Generic is already defined before defining this algorithm.

**APPENDIX 1:** CCG1 PHN\_Alinity\_IA; PHN\_Process Path\_IA; Jam Recovery

**Algorithm Code**

SELECT

EVALS.MODULESN

FROM

(SELECT

INNER1.MODULESN,

COUNT(INNER1.MODULESN) AS NUM\_RETRIES

FROM

(SELECT

M.MODULESN

FROM

IDAQOWNER.ICQ\_MESSAGEHISTORY M

WHERE

M.LOGDATE\_LOCAL >= TRUNC(SYSDATE) - 1

AND M.LOGDATE\_LOCAL < TRUNC(SYSDATE)

AND M.AIMCODE = '5758'

AND M.AIMSUBCODE = 'D298') INNER1

LEFT JOIN

(SELECT

R.MODULESN,

COUNT(R.CORRECTEDCOUNT) AS NUM\_RESULTS

FROM

IDAQOWNER.ICQ\_RESULTS R

WHERE

R.LOGDATE\_LOCAL >= TRUNC(SYSDATE) - 1

AND R.LOGDATE\_LOCAL < TRUNC(SYSDATE)

AND R.CORRECTEDCOUNT IS NOT NULL

GROUP BY

R.MODULESN

) INNER2

ON INNER1.MODULESN = INNER2.MODULESN

WHERE

INNER2.NUM\_RESULTS >= 10

GROUP BY

INNER1.MODULESN) EVALS

WHERE

EVALS.NUM\_RETRIES >= 4

**Apollo Algorithm Details**

(\* is Mandatory)

|  |  |
| --- | --- |
| **Apollo Details** |  |
| Algorithm ID \* | Alinity IA Process Path Jams 5758 |
| Algorithm Name \* | Alinity IA Process Path Jams 5758 |
| Algorithm Description \* | To detect a buildup of process path retries (AimCode: 5758) before the customer begins experiencing process path jams (AIM CODE: 5752 or 5757). |
| Product Family \* | Alinity IA |
| Algorithm Group \* | Alinity Process Path |
| Functional Area | N/A |
| Algorithm Category 1 | N/A |
| Algorithm Category 2 | N/A |
| Algorithm Category 3 | N/A |
| Remaining Useful Life Value | 7 |
| Remaining Useful Life Unit | Day |
| Keep Results Num Days | 14 |
| **Routine Details** |  |
| Routine Source | Use Reusable Routine |
| Reusable Routines | Alinity IA Process Path Jams – Generic |
| Run Mode | Batch |
| Status | Enable |
| **ODS Routine Details** |  |
| ODS Routine Name | PHM\_ODS\_ICQ\_RESULTS\_PROC |
| **Prognostic Health Notification Details** |  |
| PHN Code | PHN\_Alinity\_IA\_CCG1 |
| Issue Description (Use Algorithm Name) |  |
| Experience Code | CCG1 |
| **Knowledge Management DB Articles** |  |
| KM Article ID | KM1158157 |
| KM Article | PHN\_IA: Process Path Jam Recovery |
| **Parameters** |  |
| Parameter Group Name | PROCPATHJAMS\_THRESHOLD |
| **Parameter Name** | **Parameter Values** |
| IHN\_LEVEL3\_DESC | Alinity IA Process Path Jams 5758 |
| PROCPATHJAMS\_THRESHOLD\_AIMCODE | 5758 |
| PROCPATHJAMS\_THRESHOLD\_AIMSUBCODE | D298 |
| PROCPATHJAMS\_THRESHOLD\_NUMRETRIES | 4 |
| PROCPATHJAMS\_THRESHOLD\_NUMRESULTS | 10 |
| THRESHOLDS\_COUNT | 1 |
| THRESHOLD\_DESCRIPTION | Alinity IA Process Path Jams 5758 |
| **Chart Details** |  |
| Chart Title | Alinity IA Process Path Jams 5758 |
| Chart Type | Line |
| Chart Threshold Parameter | PROCPATHJAMS\_THRESHOLD-THRESHOLDS\_COUNT |
| Group ID | Group 7 |
| Chart X Axis Name | Date |
| Chart Y Axis Name | Threshold Count |

**APPENDIX 2:** CCI1 PHN\_Alinity\_IA; PHN\_Pretreatment Path\_IA; Jam Recovery

**Algorithm Code**

SELECT

EVALS.MODULESN

FROM

(SELECT

INNER1.MODULESN,

COUNT(INNER1.MODULESN) AS NUM\_RETRIES

FROM

(SELECT

M.MODULESN

FROM

IDAQOWNER.ICQ\_MESSAGEHISTORY M

WHERE

M.LOGDATE\_LOCAL >= TRUNC(SYSDATE) - 1

AND M.LOGDATE\_LOCAL < TRUNC(SYSDATE)

AND M.AIMCODE = '5758'

AND M.AIMSUBCODE = 'D299') INNER1

LEFT JOIN

(SELECT

R.MODULESN,

COUNT(R.CORRECTEDCOUNT) AS NUM\_RESULTS

FROM

IDAQOWNER.ICQ\_RESULTS R

WHERE

R.LOGDATE\_LOCAL >= TRUNC(SYSDATE) - 1

AND R.LOGDATE\_LOCAL < TRUNC(SYSDATE)

AND R.CORRECTEDCOUNT IS NOT NULL

GROUP BY

R.MODULESN

) INNER2

ON INNER1.MODULESN = INNER2.MODULESN

WHERE

INNER2.NUM\_RESULTS >= 10

GROUP BY

INNER1.MODULESN) EVALS

WHERE

EVALS.NUM\_RETRIES >= 4

**Apollo Algorithm Details**

(\* is Mandatory)

|  |  |
| --- | --- |
| **Apollo Details** |  |
| Algorithm ID \* | Alinity IA Pretreat Path Jams 5758 |
| Algorithm Name \* | Alinity IA Pretreat Path Jams 5758 |
| Algorithm Description \* | Detect a buildup of process path retries (AimCode: 5758) before the customer begins experiencing pretreatment path jams (AIM CODE: 5752 or 5757). |
| Product Family \* | Alinity IA |
| Algorithm Group \* | Alinity Process Path |
| Functional Area | N/A |
| Algorithm Category 1 | N/A |
| Algorithm Category 2 | N/A |
| Algorithm Category 3 | N/A |
| Remaining Useful Life Value | 7 |
| Remaining Useful Life Unit | Day |
| Keep Results Num Days | 14 |
| **Routine Details** |  |
| Routine Source | Use Reusable Routine |
| Reusable Routines | Alinity IA Process Path Jams – Generic |
| Routine Type | Oracle Procedure |
| Run Mode | Batch |
| Status | Enable |
| **ODS Routine Details** |  |
| ODS Routine Name | PHM\_ODS\_ICQ\_RESULTS\_PROC |
| **Prognostic Health Notification Details** |  |
| PHN Code | PHN\_Alinity\_IA\_CCI1 |
| Issue Description (Use Algorithm Name) |  |
| Experience Code | CCI1 |
| **Knowledge Management DB Articles** |  |
| KM Article ID | KM1158157 |
| KM Article | PHN\_IA: Process Path Jam Recovery |
| **Parameters** |  |
| Parameter Group Name | PROCPATHJAMS\_THRESHOLD |
| **Parameter Name** | **Parameter Values** |
| IHN\_LEVEL3\_DESC | Alinity IA Pretreat Path Jams 5758 |
| PROCPATHJAMS\_THRESHOLD\_AIMCODE | 5758 |
| PROCPATHJAMS\_THRESHOLD\_AIMSUBCODE | D299 |
| PROCPATHJAMS\_THRESHOLD\_NUMRETRIES | 4 |
| PROCPATHJAMS\_THRESHOLD\_NUMRESULTS | 10 |
| THRESHOLDS\_COUNT | 1 |
| THRESHOLD\_DESCRIPTION | Alinity IA Pretreat Path Jams 5758 |
| **Chart Details** |  |
| Chart Title | Alinity IA Pretreat Path Jams 5758 |
| Chart Type | Line |
| Chart Threshold Parameter | PROCPATHJAMS\_THRESHOLD-THRESHOLDS\_COUNT |
| Group ID | Group 7 |
| Chart X Axis Name | Date |
| Chart Y Axis Name | Threshold Count |

**APPENDIX 3:** Algorithm Understanding Check – Algorithm Developer to Prognostic Health Monitoring (PHM) Specialist Transition

**Data Set Description**

The data set for this understanding check was retrieved from the ICQOWNER.ODR\_MESSAGEHISTORYICQ and ICQOWNER.ODR\_RESULTSICQ tables within the BSQD1I database. Data was collected for all instruments between May 1, 2017 and June 30, 2017, inclusive. For the ICQOWNER.ODR\_MESSAGEHISTORYICQ table, data collection was limited to the MODULESN, LOGDATE, AIMCODE, and AIMSUBCODE fields. For the ICQOWNER.ODR\_RESULTSICQ table, data collection was limited to the SERIALNUMBER, DATETIMESTAMP, and CORRECTEDCOUNT fields.

**Data Set Retrieval**

The following SQL code was used to retrieve the data set from the ICQOWNER.ODR\_MESSAGEHISTORYICQ table:

SELECT

M.MODULESN,

M.LOGDATE,

M.AIMCODE,

M.AIMSUBCODE

FROM

ICQOWNER.ODR\_MESSAGEHISTORYICQ M

WHERE

TRUNC(M.LOGDATE) >= TO\_DATE('05/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND TRUNC(M.LOGDATE) < TO\_DATE('07/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

The following SQL code was used to retrieve the data set from the ICQOWNER.ODR\_RESULTSICQ table:

SELECT

R.SERIALNUMBER,

R.DATETIMESTAMP,

R.CORRECTEDCOUNT

FROM

ICQOWNER.ODR\_RESULTSICQ R

WHERE

TRUNC(R.DATETIMESTAMP) >= TO\_DATE('05/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND TRUNC(R.DATETIMESTAMP) < TO\_DATE('07/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

**Algorithm Developer Analysis**

The following steps were performed by the Algorithm Developer to analyze the data set and flag algorithm violations:

* Query BSQD1i to get a count of number of Aimcode 5758 messages per day per ModuleSN where AimSubCode matches D298
* Query BSQD1i to get a count of number of CorrectedCounts <> NA per day per SerialNumber
* Join both table on ModuleSN/SerialNumber and Truncated Logdate/DateTimeStamp
* Generate a flag for every ModuleSN if # CorrectedCount/Day >= 10 and if # 5758-D298 messages/day>=4

**PHM Specialist Analysis**

The following SQL code was used by the PHM Specialist to analyze the data set and flag algorithm violations:

SELECT

EVALS.\*,

CASE WHEN EVALS.AIMSUBCODE = 'D298'

THEN 'Process Path'

ELSE 'Incubation Track'

END AS LOCATION

FROM

(SELECT

INNER1.MODULESN,

INNER1.DAY,

INNER1.AIMSUBCODE,

INNER2.NUM\_RESULTS,

COUNT(INNER1.AIMSUBCODE) AS NUM\_RETRIES

FROM

(SELECT

M.MODULESN,

TRUNC(M.LOGDATE) AS DAY,

M.AIMCODE,

M.AIMSUBCODE,

M.EXCEPTIONSTRING

FROM

ICQOWNER.ODR\_MESSAGEHISTORYICQ M

WHERE

TRUNC(M.LOGDATE) >= TO\_DATE('05/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND TRUNC(M.LOGDATE) < TO\_DATE('07/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND M.AIMCODE = '5758'

AND (M.AIMSUBCODE = 'D298' OR M.AIMSUBCODE = 'D299')) INNER1

LEFT JOIN

(SELECT

R.SERIALNUMBER,

TRUNC(R.DATETIMESTAMP) AS DAY,

COUNT(R.CORRECTEDCOUNT) AS NUM\_RESULTS

FROM

ICQOWNER.ODR\_RESULTSICQ R

WHERE

TRUNC(R.DATETIMESTAMP) >= TO\_DATE('05/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND TRUNC(R.DATETIMESTAMP) < TO\_DATE('07/01/2017 12:00:00 AM', 'mm/dd/yyyy hh:mi:ss am')

AND R.CORRECTEDCOUNT IS NOT NULL

GROUP BY

R.SERIALNUMBER,

TRUNC(R.DATETIMESTAMP)

ORDER BY

R.SERIALNUMBER,

TRUNC(R.DATETIMESTAMP)

) INNER2

ON INNER1.MODULESN = INNER2.SERIALNUMBER

AND INNER1.DAY = INNER2.DAY

WHERE

INNER2.NUM\_RESULTS >= 10

GROUP BY

INNER1.MODULESN,

INNER1.DAY,

INNER1.AIMSUBCODE,

INNER2.NUM\_RESULTS) EVALS

WHERE

EVALS.NUM\_RETRIES >= 4

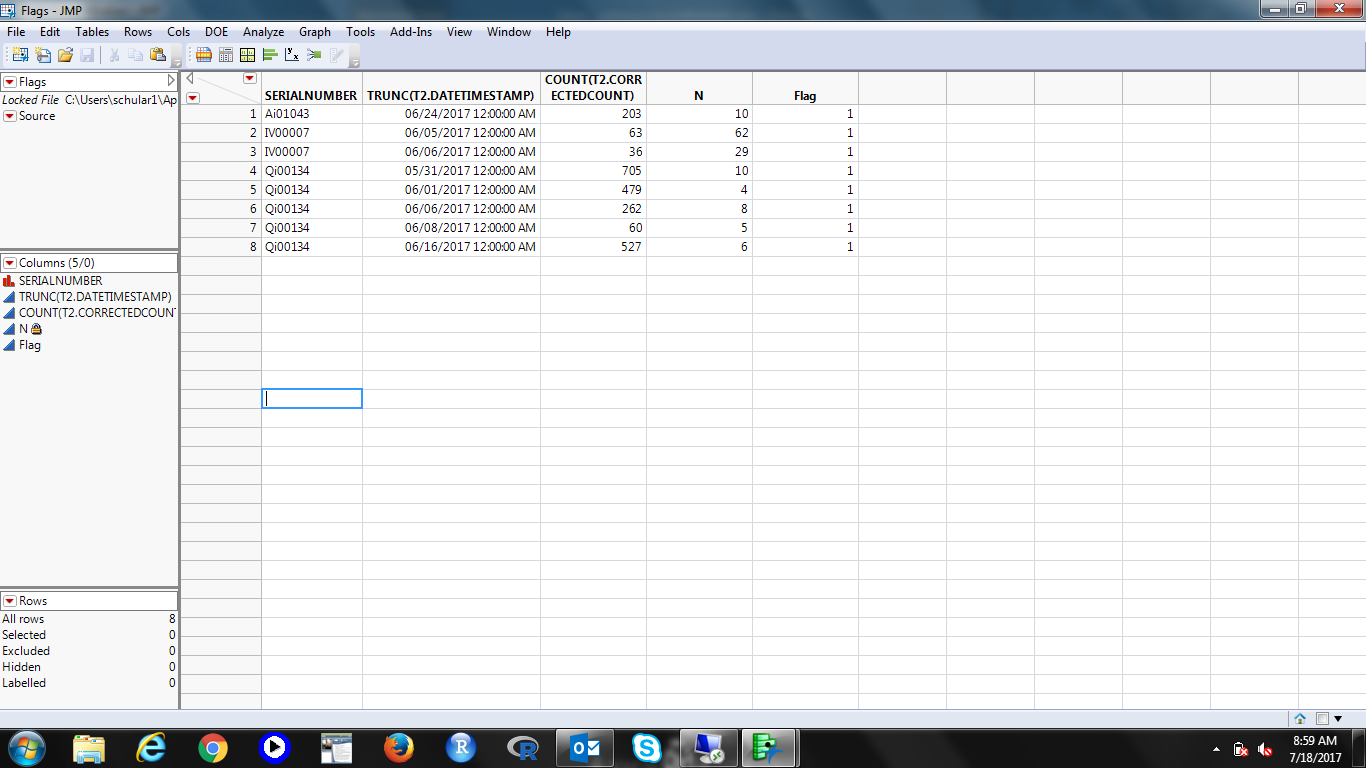
ORDER BY

EVALS.MODULESN,

EVALS.DAY

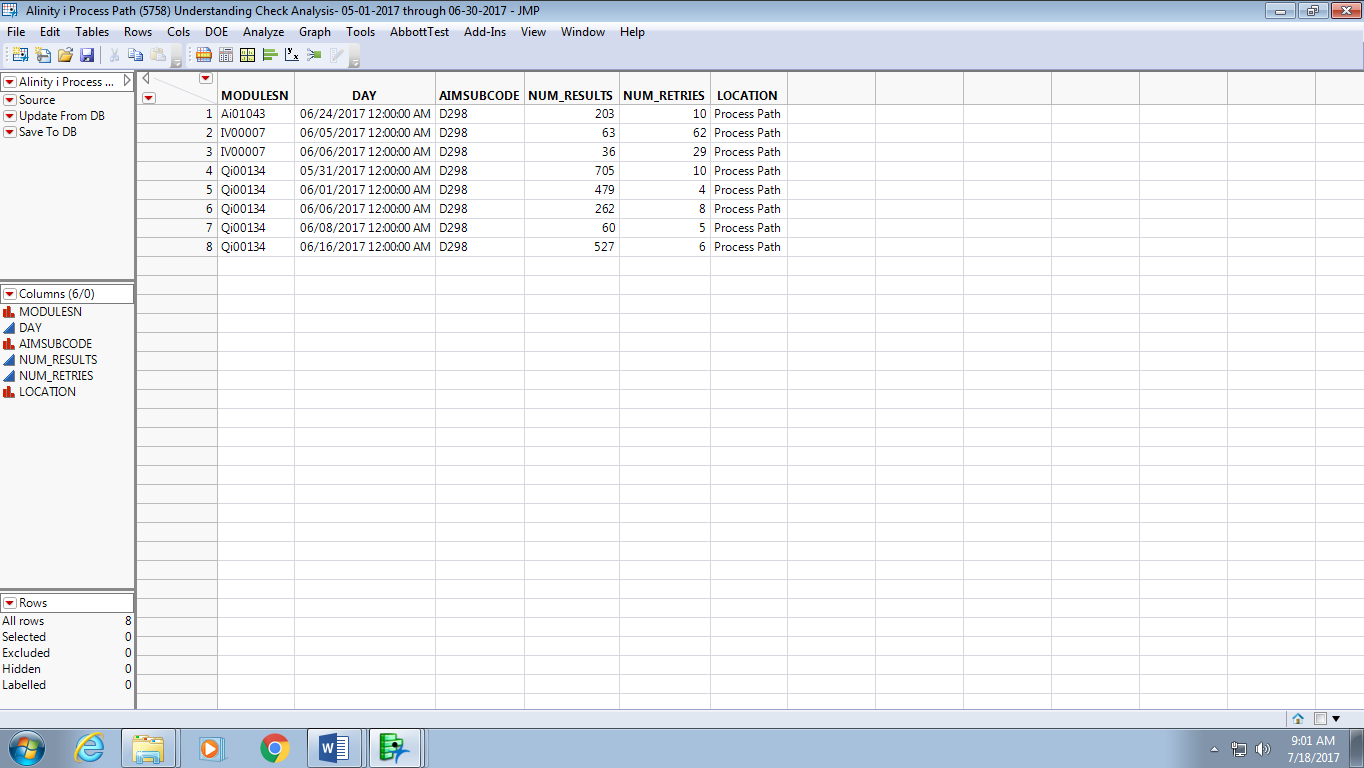
**Algorithm Developer Analysis Output**

The 8 instruments (SERIALNUMBER) in the table below were identified as violating the algorithm by the Algorithm Developer. Note that per the algorithm developer, the above 8 instruments violated the algorithm for the process path location, not the pretreatment path location.



**PHM Specialist Analysis Output**

The following 8 instruments (MODULESN) were identified as violating the algorithm by the PHM Specialist:



**Algorithm Developer & PHM Specialist Output Comparison**

|  |  |  |  |
| --- | --- | --- | --- |
| Total # of Unique Instrument-Days Tested\* | Total # of Algorithm Developer Flags | Total # of PHM Specialist Flags | Total # of Matched Flags (Algorithm Developer vs. PHM Specialist) |
| **11,633** | **8** | **8** | **8** |

**\*** Based on data from the ICQOWNER.ODR\_MESSAGEHISTORYICQ table query above. Based on data from the ICQOWNER.ODR\_RESULTSICQ table, the value would be 2,727.

**Understanding Check Summary**

Based on the outputs from both the Algorithm Developer and PHN Specialist, the PHN Specialist’s understanding of the delivered algorithm is confirmed. Both the Algorithm Developer and PHN Specialist analyzed the same data set and got the same results. In particular, the SERIALNUMBER/MODULESN and TRUNC(T2.DATETIMESTAMP)/DAY fields matched for all 8 instrument-day (SERIALNUMBER/MODULESN-TRUNC(T2.DATETIMESTAMP)/DAY) combinations. This means that both the Algorithm Developer and PHN Specialist flagged the same 8 algorithm violations within the given data set. Furthermore, there were no mismatches between the output from the Algorithm Developer and the output from the PHM Specialist.

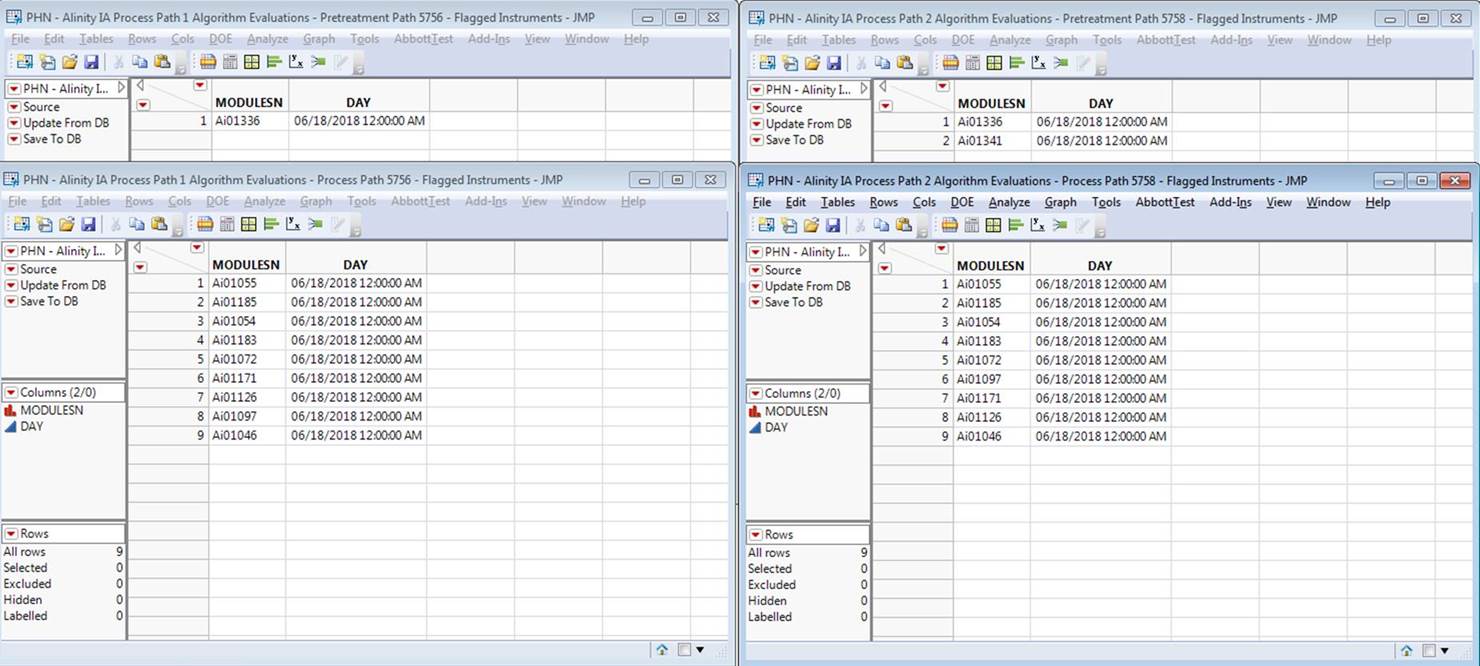
**APPENDIX 4:** Algorithm Transition to Apollo – PHM Specialist to Apollo Developer

**Data Set Description**

The data set for this transition was retrieved from the IDAQOWNER.ICQ\_MESSAGEHISTORY and IDAQOWNER.ICQ\_RESULTS tables within the DABBTO database. Data was collected for all available instruments on June 18, 2018.

**PHM Specialist Analysis Output**

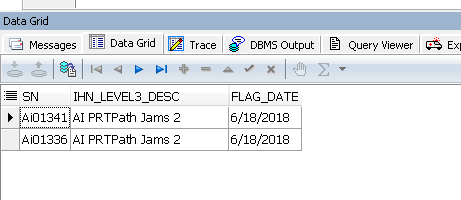
The following 11 instrument-days (MODULESN-DAY) were identified as violating the algorithm by the PHM Specialist:



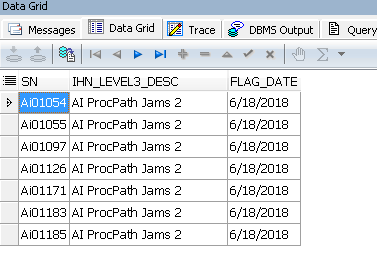
**Apollo Developer Analysis Output**

The following 9 instrument-days (SN-FLAG\_DATE) were identified as violating the algorithm by the Apollo Developer (note that where the Apollo Developer and PHM Specialist don’t match is because the instruments do not exist in the Apollo environment, and therefore have no data to run the algorithm against):

Pretreatment Path:



Process Path:



**Algorithm Transition Summary**

Based on the outputs from both the Apollo Developer and PHM Specialist, the Apollo Developer’s understanding of the delivered algorithm is confirmed. Both the Apollo Developer and PHM Specialist analyzed the same data set and got the same results (with the exception of the instruments not in Apollo as mentioned previously). In particular, the MODULESN/SN and DAY/FLAG\_DATE fields matched by path. This means that both the Apollo Developer and PHM Specialist flagged the same algorithm violations within the given data set.