status quo of all applied recent corrections (sqoaarc)

**ESSENTIAL PROGRAM ARTIFACT- I**

**Table of Contents**

[**1) Location of the Program/Project Folder (SAS Server & DAPGS Shared Folder) 2**](#_Toc114045347)

[**2) README Instructions for the SAS Program (SQOAARC) 2**](#_Toc114045348)

[**3) SQOAARC Program Code 3**](#_Toc114045349)

## **Location of the Program/Project Folder (SAS Server & DAPGS Shared Folder)**

* 1. On the ***server***, it is available at: /proj/sas/sasdata/risk/discovery/ipds/DAPGS *(not a clickable link, please copy and paste it in SAS)*. You can also locate it manually on the server by following this stepwise procedure: **RRMInteractive\_PRD 🡪 Files 🡪 ipds 🡪 DAPGS 🡪** **Status Quo of Recently Applied Corrections\_SAS Scripts\_V1** (*2 programs (versions) currently exist, but we are primarily concerned with the modified program*).
  2. On the ***shared folder***, it can be located at:

*\\Svrau100qsm00.oceania.corp.anz.com\GRPRM RRMg\Risk Australia\Retail and Commercial Risk Infrastructure\Squads\DAPGS\ERT\SAS\SAS Code - Status Quo of All Applied Recent Corrections (SQOAARC) (not a clickable link, please copy and paste it in offline Folder Path search area).* Kindly note that there are three subfolders containing different elements – one is with the SAS program scripts in its native SAS format; the second one is the entire SQOAARC program (modified and final version) along with procedural instructions rendered in Word format (*for easy access for anyone and everyone to read the program codes*). The final folder in here is for the SAS sample output presentation in Excel format (*derived from the modified version*).

# **README Instructions for the SAS Program (SQOAARC)**

1) To begin with, use the modified version of the program only for best results, you can *Select All* (Ctrl + A) and run the entire (modified version) program in one go. All the output tables would eventually be created right at the end of the program post a successful run.

2) This program only deals with all CRDH corrections undertaken in the past *4 weeks* or so (flexible date ranges also supported here as per the requirements). The results obtained are of the *highest granularity*.

3) Firstly, all the *RHI & Default Amendment* Tables are merged into a single table called **Recent\_Corrections** (Step 1).

4) Similarly, all the *RHI & Default Removals* are also merged into another unified table called **Recent\_Removals** (Step 2).

5) All the events table are clubbed into a single consolidated *Events* table next which is named **Complete\_Events** (Step 3).

6) This all-in-one events table is then joined separately with the two tables obtained from *Step 1 & 2*. A combined table by the name of **Total\_Corrections** is hereafter rendered by joining the two separate tables created in this process (Step 4).

7) All the *Response* tables are thereafter stacked into a single consolidated table called **Complete\_Responses** in an identical fashion to the events table (Step 5).

8) Subsequently, this all-in-one response table is joined with the filtered table from *Step 4* containing all the plausible available corrected events. Thus, we now obtain our penultimate stage output table with all the requisite details by virtue of the **Recent\_Correction\_Details** table (Step 6).

9) Finally, we modify this obtained table from *Step 6* and transform it to a new final output table in the form of **All\_Applied\_Recent\_Corrections** containing a few *additional fields* (columns) and *removal of logically duplicate records* (this is done in 2 sub steps, 7A (for additional fields) & 7B (for removal of logically duplicate records and columns reordering)). The extra fields are basically some calculated measures introduced to provide some *useful insights* for deciding further course of actions to be taken (if any).

***(Please read all the comments carefully throughout the SAS program for more info.)***

# **SQOAARC Program Code**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* STATUS QUO OF ALL APPLIED RECENT CORRECTIONS - ALL-IN-ONE \*/

options compress=binary;

/\* Variables Macros \*/

%let from\_date = '2022-08-01';

%let intermediate\_date = '2022-08-25';

%let to\_date = '2022-09-09';

%let alternative\_to\_date = '2022-09-12';

/\* PART-I - CORRECTIONS APPLIED RECENTLY (DEFAULT & RHI BOTH; ADJUSTMENT/AMENDMENT & REMOVAL BOTH) \*/

/\* STEP 1 (DEFAULT & RHI ADJUSTMENT/AMENDMENT CASES) \*/

**proc** **sql**;

connect to odbc(authdomain='ODBCAuthCAMO' DATASRC="AbSQL0");

create table recent\_corrections

as select \* from connection to odbc

(select 'CC', account\_number, correction\_date, process\_date, subrecord,

field\_name, old\_field\_value, new\_field\_value

from ipds\_cc.correction\_store\_src\_cc\_latest

where correction\_date between &from\_date. and &to\_date.

UNION

select 'HL', account\_number, correction\_date, process\_date, subrecord,

field\_name, old\_field\_value, new\_field\_value

from ipds\_hl.correction\_store\_src\_hl\_latest

where correction\_date between &from\_date. and &to\_date.

UNION

select 'PL', account\_number, correction\_date, process\_date, subrecord,

field\_name, old\_field\_value, new\_field\_value

from ipds\_pl.correction\_store\_src\_pl\_latest

where correction\_date between &from\_date. and &to\_date.

UNION

select 'OD', account\_number, correction\_date, process\_date, subrecord,

field\_name, old\_field\_value, new\_field\_value

from ipds\_od.correction\_store\_src\_od\_latest

where correction\_date between &from\_date. and &to\_date.

order by correction\_date desc);

disconnect from odbc;

**quit**;

/\* STEP 2 (DEFAULT & RHI REMOVAL CASES) \*/

**proc** **sql**;

connect to odbc(authdomain='ODBCAuthCAMO' DATASRC="AbSQL0");

create table recent\_removals

as select \* from connection to odbc

(select 'CC', account\_number, correction\_date, process\_date, dataset\_to\_be\_removed,

field\_name, field\_value

from ipds\_cc.correction\_store\_removal\_cc\_latest

where correction\_date between &from\_date. and &to\_date.

UNION

select 'HL', account\_number, correction\_date, process\_date, dataset\_to\_be\_removed,

field\_name, field\_value

from ipds\_hl.correction\_store\_removal\_hl\_latest

where correction\_date between &from\_date. and &to\_date.

UNION

select 'PL', account\_number, correction\_date, process\_date, dataset\_to\_be\_removed,

field\_name, field\_value

from ipds\_pl.correction\_store\_removal\_pl\_latest

where correction\_date between &from\_date. and &to\_date.

UNION

select 'OD', account\_number, correction\_date, process\_date, dataset\_to\_be\_removed,

field\_name, field\_value

from ipds\_od.correction\_store\_removal\_od\_latest

where correction\_date between &from\_date. and &to\_date.

order by correction\_date desc);

disconnect from odbc;

**quit**;

/\* STEP 3 - TEMPORARY CONSOLIDATED EVENTS TABLE CREATION FOR SUBSEQUENT JOINING \*/

**proc** **sql**;

connect to odbc(authdomain='ODBCAuthCAMO' DATASRC="AbSQL0");

create table complete\_events

as select \* from connection to odbc

(select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_cc.crb\_events\_cc\_experian

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_cc.crb\_events\_cc\_equifax

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_cc.crb\_events\_cc\_illion

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_hl.crb\_events\_hl\_experian

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_hl.crb\_events\_hl\_equifax

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_hl.crb\_events\_hl\_illion

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_pl.crb\_events\_pl\_experian

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_pl.crb\_events\_pl\_equifax

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_pl.crb\_events\_pl\_illion

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_od.crb\_events\_od\_experian

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_od.crb\_events\_od\_equifax

where process\_date between &from\_date. and &to\_date.

and correction = **1**

UNION

select account\_number, account\_sub\_id, primary\_unformatted\_name as full\_name, birth\_date,

current\_unformatted\_address as current\_address, previous\_unformatted\_address as previous\_address,

date\_corrected, record\_id, correction, process\_date as event\_date

from ipds\_od.crb\_events\_od\_illion

where process\_date between &from\_date. and &to\_date.

and correction = **1**);

disconnect from odbc;

**quit**;

/\* PHYSICAL STRUCTURE INSPECTION \*/

/\*

PROC CONTENTS DATA = COMPLETE\_EVENTS;

RUN;

PROC CONTENTS DATA = RECENT\_CORRECTIONS;

RUN;

PROC CONTENTS DATA = RECENT\_REMOVALS;

RUN;

\*/

/\* STEP 4 - MATCHING CORRECTED ACCOUNTS WITH CORRESPONDING EVENTS GENERATED FOR THEM (FOR ALL PORTFOLIOS & CRBS) \*/

**proc** **sql**;

create table correction\_events as

select distinct(e.account\_number), coalesce(e.account\_sub\_id, 'N/A') as account\_sub\_id, e.full\_name, e.birth\_date,

e.current\_address, coalesce(e.previous\_address, 'No Previous Address Available') as previous\_address,

coalesce(c.subrecord, 'Not A Part of The Correction Store') as c\_subrecord,

coalesce(c.field\_name, 'Not A Part of The Correction Store') as c\_field\_name,

coalesce(c.old\_field\_value, 'Not A Part of The Correction Store') as c\_old\_field\_value,

coalesce(c.new\_field\_value, 'Corrected') as c\_new\_field\_value,

c.correction\_date, e.date\_corrected, e.correction

from WORK.recent\_corrections as c left join WORK.complete\_events as e

on c.account\_number = e.account\_number

where c.process\_date is not null

and e.account\_number is not null

order by c.correction\_date desc;

**quit**;

**proc** **sql**;

create table removal\_events as

select distinct(e.account\_number), coalesce(e.account\_sub\_id, 'N/A') as account\_sub\_id, e.full\_name, e.birth\_date,

e.current\_address, coalesce(e.previous\_address, 'No Previous Address Available') as previous\_address,

coalesce(c.dataset\_to\_be\_removed, 'Not A Part of The Removal Store') as r\_dataset\_to\_be\_removed,

coalesce(c.field\_name, 'Default Status') as r\_field\_name,

coalesce(c.field\_value, 'Retracted or Removed') as r\_field\_value,

c.correction\_date, e.date\_corrected, e.correction

from WORK.recent\_removals as c left join WORK.complete\_events as e

on c.account\_number = e.account\_number

where c.process\_date is not null

and e.account\_number is not null

order by c.correction\_date desc;

**quit**;

**data** total\_corrections;

set correction\_events removal\_events;

**run**;

/\* STEP 5 - TEMPORARY CONSOLIDATED RESPONSE TABLE CREATION FOR SUBSEQUENT JOINING \*/

**proc** **sql**;

connect to odbc(authdomain='ODBCAuthCAMO' DATASRC="AbSQL0");

create table complete\_responses

as select \* from connection to odbc

(select 'EX' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_cc.ipds\_response\_experian\_cc

where process\_date between &from\_date. and &to\_date.

UNION

select 'EQ' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_cc.ipds\_response\_equifax\_cc

where process\_date between &from\_date. and &to\_date.

UNION

select 'I' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_cc.ipds\_response\_illion\_cc

where process\_date between &from\_date. and &to\_date.

UNION

select 'EX' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_hl.ipds\_response\_experian\_hl

where process\_date between &from\_date. and &to\_date.

UNION

select 'EQ' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_hl.ipds\_response\_equifax\_hl

where process\_date between &from\_date. and &to\_date.

UNION

select 'I' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_hl.ipds\_response\_illion\_hl

where process\_date between &from\_date. and &to\_date.

UNION

select 'EX' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_pl.ipds\_response\_experian\_pl

where process\_date between &from\_date. and &to\_date.

UNION

select 'EQ' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_pl.ipds\_response\_equifax\_pl

where process\_date between &from\_date. and &to\_date.

UNION

select 'I' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_pl.ipds\_response\_illion\_pl

where process\_date between &from\_date. and &to\_date.

UNION

select 'EX' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_od.ipds\_response\_experian\_od

where process\_date between &from\_date. and &to\_date.

UNION

select 'EQ' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_od.ipds\_response\_equifax\_od

where process\_date between &from\_date. and &to\_date.

UNION

select 'I' as bureau\_name, accountid\_accountnumber, recordid, accounttype, acctprocessingmessages\_code, acctprocessingmessages\_category,

acctprocessingmessages\_messagetext as processing\_messagetext, acctprocessingmessages\_inputpath,

process\_date as response\_date

from ipds\_od.ipds\_response\_illion\_od

where process\_date between &from\_date. and &to\_date.);

disconnect from odbc;

**quit**;

/\* STEP 6 - CONSOLIDATED PENULTIMATE OVERALL CORRECTIONS TABLE (RESPONSES JOINED WITH THE TOTAL CORRECTIONS FROM STEP 4) \*/

**proc** **sql**;

create table recent\_correction\_details as

select distinct(e.account\_number) as event\_account\_no, coalesce(e.account\_sub\_id, 'N/A') as event\_account\_sub\_id,

e.full\_name as event\_full\_name, e.birth\_date as event\_birth\_date, e.current\_address as event\_current\_address,

coalesce(e.previous\_address, 'No Previous Address Available') as event\_previous\_address,

c.correction\_date as corr\_store\_correction\_date, e.date\_corrected as event\_date\_corrected,

c.r\_dataset\_to\_be\_removed as removal\_dataset\_to\_be\_removed, c.r\_field\_name as removal\_field\_name,

c.r\_field\_value as removal\_field\_value, c.c\_subrecord as correction\_subrecord, c.c\_field\_name as correction\_field\_name,

c.c\_old\_field\_value as correction\_old\_field\_value, c.c\_new\_field\_value as correction\_new\_field\_value,

e.event\_date, r.response\_date, e.record\_id as event\_record\_id,

r.recordid as response\_record\_id, e.correction as event\_correction\_flag, r.accounttype as response\_account\_type,

r.bureau\_name as response\_bureau\_name,

r.acctprocessingmessages\_code as response\_message\_code,

r.acctprocessingmessages\_category as response\_message\_category,

r.processing\_messagetext as response\_message\_text,

r.acctprocessingmessages\_inputpath as response\_input\_path

from WORK.total\_corrections as c left join WORK.complete\_events as e

on c.account\_number = e.account\_number

left join WORK.complete\_responses as r

on e.account\_number = r.accountid\_accountnumber

and e.record\_id = r.recordid

order by c.correction\_date asc;

**quit**;

/\* STEP 7(A) - FINAL OUTPUT TABLE - UNFILTERED (ADDITIONAL VARIABLES TO THE TABLE OBTAINED FROM STEP 6) \*/

**data** applied\_recent\_corrections;

set recent\_correction\_details;

current\_date = today();

format current\_date date9.;

days\_since\_corrected = INTCK('day', corr\_store\_correction\_date, current\_date);

if (days\_since\_corrected > **14** and response\_date =**.**)

then action\_recommended = "Take Immediate Action for Resolution";

else if (**10** <= days\_since\_corrected <= **14** and response\_date =**.**)

then action\_recommended = "Take Notice For Probable Actions";

else if (response\_message\_category = 'E')

then action\_recommended = "Swift Error Rectification Needed";

else action\_recommended = "No Action Needed";

**run**;

/\* STEP 7(B) - FINAL OUTPUT TABLE - FILTERED (REMOVAL OF ALL THE LOGICALLY DUPLICATE RECORDS FROM STEP 7A) \*/

/\* data all\_applied\_recent\_corrections;

set applied\_recent\_corrections;

if correction\_new\_field\_value = "Corrected" then delete;

run; \*/

**data** all\_applied\_recent\_corrections;

retain event\_account\_no event\_account\_sub\_id current\_date corr\_store\_correction\_date event\_date\_corrected

days\_since\_corrected action\_recommended event\_full\_name event\_birth\_date event\_current\_address event\_previous\_address

removal\_dataset\_to\_be\_removed removal\_field\_name removal\_field\_value correction\_subrecord correction\_field\_name

correction\_old\_field\_value correction\_new\_field\_value event\_correction\_flag event\_date response\_date event\_record\_id

response\_record\_id response\_account\_type response\_bureau\_name response\_message\_code response\_message\_category

response\_message\_text response\_input\_path;

set applied\_recent\_corrections;

if correction\_new\_field\_value = "Corrected" then delete;

**run**;

/\* CODE SNIPPETS FOR TESTING \*/

/\* PROC SQL;

SELECT \* FROM WORK.TOTAL\_CORRECTIONS WHERE

ACCOUNT\_NUMBER NOT IN (SELECT ACCOUNT\_NUMBER FROM WORK.RECENT\_CORRECTION\_DETAILS); \*/

/\* proc sql;

select COUNT (DISTINCT account\_number) as total\_count

from WORK.recent\_correction\_details;

quit; \*/

\*\*\*\*\*\*\*\*\*\*\*\* THE END \*\*\*\*\*\*\*\*\*\*\*\*