CCDD Release Candidate QA Process

# Initial File Loading

Load all files into a relational database.

Note: in order to run the queries, there needs to be a copy of all the files from the previous release also available, in a separate database in the same tooling.

## Points to consider

Set the load to exactly the data type and field lengths of the specification for each file (the database tool used will probably determine the exact thing for that; in SQL we use Unicode\_string for the STs).

Set the coding to UTF-8.

Use the “ delimiter.

A load process could (should) be developed (very like stored procedures – an “Import module”); we started to do this, but the datasets have not had enough consistency to make it viable. When there are issues, the “trial and error” process needed to get the data to load somehow and then find the issues has needed more flexibility than an import module would (or should) give.

Again, there has not been enough consistency for this to have been viable yet, but the end goal is to have the databases stored in such a way that queries can be run without editing. In SQL, this would probably mean taking “views” and then being able to run each set of queries just using “old” and “new” as the differentiators. Other database tools may have other procedures for this.

Any variation from the Technical Specification should be reported immediately. This includes:

* any file where fields are not populated correctly (especially those fields where an NA should be present if there is no other data; particularly ntp\_type)
* any data where special characters are required (e.g. CLÉO-35; mp\_code 02436736 in both mp\_file and the Relationship file)
* checking that no separation has occurred incorrectly (incorrect placing of the quote (“) character in a product name)

## Initial Counts

Take basic row counts of each file as it loads into its table and report. Any reduction from the previous release must be investigated immediately.

The MP and Relationship Files should have exactly the same number of rows.

##### Report as:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **New Release Candidate**  **2018-MM-DD** | **Previous Release**  **2018-01-22** | **Change** |
| **TM** |  | 288 | +xx |
| **NTP** |  | 1284 | +xx |
| **MP** |  | 5767 | +xx |
| **Relationship** |  | 5767 | +xx |
| **Special Grouping** |  | 1002 | +xx |
| **Coded Attribute** |  | 6 | No change |
| **NTP Device** |  | 3 | No change |

Note that the Mapping File may become redundant if the Technical Specification changes proposed in January 2018 are accepted.

## TM File Checks

### File compare

Using Comparator tool, compare Release Candidate file to Previous Month’s Publication Release to find

* Additions
* Changes
* Deletions – **there should be none**

#### Results

* Concept additions should be as expected; these must be checked by an SME
* Concept changes should be rare and must be checked by an SME; they can only be
* minor changes to tm\_formal\_name
* status changes to “inactivated”; these occur only when all related NTPs are have been inactivated; status date should be within the preceding month

Report results in a spreadsheet and summarise in a Word document.

##### SQL used:

SELECT \*

FROM [tm table new] as tm

WHERE NOT EXISTS (

SELECT \*

FROM [tm table old] as tm\_old

WHERE tm.[tm\_code] = tm\_old.[tm\_code]

AND tm.[tm\_formal\_name] = tm\_old.[tm\_formal\_name]

AND tm.[tm\_status] = tm\_old.[tm\_status]

AND tm.[tm\_status\_effective\_time] = tm\_old.[tm\_status\_effective\_time]

)

order by tm\_code

Then re-run query, reversing old and new.

### Duplication check

To be run for tm\_code and tm\_formal\_name; there should be no duplication.

#### Results

Report as statements.

##### SQL used:

TM CODE DUPLICATION

With temp (duplicateTMcode, x) as (

SELECT tm\_code, count ([tm\_code])

FROM [tm table new]

group by tm\_code

having count ([tm\_code]) >1

)

Select \* from [tm table new]

inner join temp on tm\_code = temp.duplicateTMcode

order by tm\_code

--==========================================

TM FORMAL NAME DUPLICATION

With temp (duplicateTMname, x) as (

SELECT tm\_formal\_name, count ([tm\_formal\_name])

FROM [tm table new]

group by tm\_formal\_name

having count ([tm\_formal\_name]) >1

)

Select \* from [tm table new]

inner join temp on tm\_formal\_name = temp.duplicateTMname

order by tm\_formal\_name

### Blank fields check

SELECT \*

FROM dbo.tm\_table as tm

where LEN(tm.tm\_code) = 0 OR LEN(tm.tm\_formal\_name) = 0 OR LEN(tm.tm\_status) = 0 OR LEN(tm.tm\_status\_effective\_time) = 0

There should be no blank fields.

### TM Status Check

SELECT \*

FROM dbo.tm\_table as tm

where NOT ((tm.tm\_status) like 'Active') AND NOT ((tm.tm\_status) like 'Inactive')

AND NOT ((tm.tm\_status) like 'Deprec')

## NTP File Checks

### File compare

Using Comparator tool, compare Release Candidate file to Previous Month’s Publication Release to find

* Additions
* Changes
* Deletions – **there should be none**

#### Results

* Concept additions should be as expected; these must be checked by an SME
* Concept changes should be rare and must be checked by an SME; they can only be
* minor changes to ntp\_formal\_name
* status changes to “inactivated”; these occur only when all related NTPs are have been inactivated; status date should be within the preceding month

Report results in a spreadsheet and summarise in a Word document.

##### SQL used:

SELECT \*

FROM [ntp table new] as ntp

WHERE NOT EXISTS (

SELECT \*

FROM [ntp table old] as ntp\_old

WHERE ntp.[ntp\_code] = ntp\_old.[ntp\_code]

AND ntp.[ntp\_formal\_name] = ntp\_old.[ntp\_formal\_name]

AND ntp.[ntp\_en\_description] = ntp\_old.[ntp\_en\_description]

AND ntp.[ntp\_fr\_description] = ntp\_old.[ntp\_fr\_description]

AND ntp.[ntp\_status] = ntp\_old.[ntp\_status]

AND ntp.[ntp\_status\_effective\_time] = ntp\_old.[ntp\_status\_effective\_time]

AND ntp.[ntp\_type] = ntp\_old.[ntp\_type]

)

order by ntp\_code

Then re-run query, reversing old and new.

### Duplication check

To be run for ntp\_code and ntp\_formal\_name; there should be no duplication.

#### Results

Report as statements.

##### SQL used:

==========================================

NTP FORMAL NAME DUPLICATION

With temp (duplicateNTPname, x) as (

SELECT NTP\_formal\_name, count ([NTP\_formal\_name])

FROM [ntp table new]

group by NTP\_formal\_name

having count ([NTP\_formal\_name]) >1

)

Select \* from [ntp table new]

inner join temp on NTP\_formal\_name = temp.duplicateNTPname

order by NTP\_formal\_name

==========================================

NTP CODE DUPLICATION

With temp (duplicateNTPcode, x) as (

SELECT NTP\_code, count ([NTP\_code])

FROM [ntp table new]

group by NTP\_code

having count ([NTP\_code]) >1

)

Select \* from [ntp table new]

inner join temp on NTP\_code = temp.duplicateNTPcode

order by NTP\_code

### Blank fields check

SELECT \*

FROM dbo.ntp\_table as ntp

where LEN(ntp.ntp\_code) = 0 OR LEN(ntp.ntp\_formal\_name) = 0 OR LEN(ntp.ntp\_en\_description) = 0 OR LEN(ntp.ntp\_fr\_description) = 0 OR

LEN(ntp.ntp\_status) = 0 OR LEN(ntp.ntp\_status\_effective\_time) = 0 OR LEN(ntp.ntp\_type) = 0

There should be no blank fields.

### NTP Type Check

SELECT \*

FROM dbo.ntp\_table as ntp

where NOT ((ntp.ntp\_type) like 'NA') AND NOT ((ntp.ntp\_type) like 'Comb')

### NTP Status Check

SELECT \*

FROM dbo.ntp\_table as ntp

where NOT ((ntp.ntp\_status) like 'Active') AND NOT ((ntp.ntp\_status) like 'Inactive')

AND NOT ((ntp.ntp\_status) like 'Deprec')

## MP File Checks

### File compare

Using Comparator tool, compare Release Candidate file to Previous Month’s Publication Release to find

* Additions
* Changes
* Deletions

#### Results

* Concept additions should be as expected; these should be checked by an SME
* Concept changes do happen and must be checked by an SME; they can only be
* changes to mp\_formal\_name
  + Company name changes, DIN retained
  + Brand name changes, DIN retained
* status changes to “inactivated”; these occur only when all related NTPs are have been inactivated; status date should be within the preceding month
* Health Canada product name changes; these should mirror the brand name changes above
* Any change in association between the mp\_code and the Health Canada identifier must be carefully checked

Report results in a spreadsheet and summarise in a Word document.

##### SQL used:

SELECT \*

FROM as mp

WHERE NOT EXISTS (

SELECT \*

FROM as mp\_old

WHERE mp\_old.[mp\_code] = mp.[mp\_code]

AND mp\_old.[mp\_formal\_name] = mp.[mp\_formal\_name]

AND mp\_old.[mp\_en\_description] = mp.[mp\_en\_description]

AND mp\_old.[mp\_fr\_description] = mp.[mp\_fr\_description]

AND mp\_old.[mp\_status] = mp.[mp\_status]

AND mp\_old.[mp\_status\_effective\_time] = mp.[mp\_status\_effective\_time]

AND mp\_old.[mp\_type] = mp.[mp\_type]

AND mp\_old.[Health\_Canada\_identifier] = mp.[Health\_Canada\_identifier]

AND mp\_old.[Health\_Canada\_product\_name] = mp.[Health\_Canada\_product\_name]

)

order by mp\_code

Then re-run query, reversing old and new.

### Duplication check

To be run for mp\_code and mp\_formal\_name; there *should* be no duplication, but likely there will be

#### Results

Report as statements; if duplication is present, report as a table with pairs (or more) of mp\_code and mp\_formal\_name.

##### SQL used:

==========================================

MP FORMAL NAME DUPLICATION

With temp (duplicateMPname, x) as (

SELECT mp\_formal\_name, count ([mp\_formal\_name])

FROM [mp table new]

group by mp\_formal\_name

having count ([mp\_formal\_name]) >1

)

Select \* from [mp table new]

inner join temp on mp\_formal\_name = temp.duplicateMPname

order by mp\_formal\_name

==========================================

MP CODE DUPLICATION

With temp (duplicateMPcode, x) as (

SELECT mp\_code, count ([mp\_code])

FROM [mp table new]

group by mp\_code

having count ([mp\_code]) >1

)

Select \* from [mp table new]

inner join temp on mp\_code = temp.duplicateMPcode

order by mp\_code

### Blank fields check

SELECT \*

FROM dbo.mp\_table as mp

where LEN(mp.mp\_code) = 0 OR LEN(mp.mp\_formal\_name) = 0 OR LEN(mp.mp\_en\_description) = 0 OR LEN(mp.mp\_fr\_description) = 0 OR

LEN(mp.mp\_status) = 0 OR LEN(mp.mp\_status\_effective\_time)=0

### MP Type Check

SELECT \*

FROM dbo.mp\_table as mp

### where NOT ((mp.mp\_type) like 'DIN') AND NOT ((mp.mp\_type) like 'CCDD')

### MP Status Check

SELECT \*

FROM dbo.mp\_table as mp

where NOT ((mp.mp\_status) like 'Active') AND NOT ((mp.mp\_status) like 'Inactive')

AND NOT ((mp.mp\_status) like 'Deprec')

## Relationship File Check

Ensure all rows have fully populated data (no “NA”s or blank fields)

SELECT \*

FROM dbo.rel\_table as rel

where LEN(rel.mp\_code) = 0 OR LEN(rel.mp\_formal\_name) = 0 OR LEN(rel.ntp\_code) = 0 OR LEN(rel.ntp\_formal\_name) = 0 OR

LEN(rel.tm\_code) = 0 OR LEN(rel.tm\_formal\_name)=0

SELECT \*

FROM [dbo.rel\_table as rel

where NOT ((rel.mp\_formal\_name) like 'NA') OR ((rel.ntp\_formal\_name)like 'NA') OR

((rel.tm\_formal\_name) like 'NA')

*Result should have same number of rows as the table itself*

### File compare

Using Comparator tool, compare Release Candidate file to Previous Month’s Publication Release to find

* Additions
* Changes
* Deletions

Because of the tripartite nature of the file, comparing every field with every field (so that there are no “pairs of changes” may be necessary, with the change reconciliation done by hand

#### Results

* Additions should be as expected; these must be checked by an SME
* Changes should be as expected; these must be checked by an SME
* Deletions – if any products are “removed” from scope (as has happened with ensulizole) then there will be “deletions” in the Relationship file

### Code consistency

* All MPs present in the Relationship File are represented in the MP File
* All MPs present in the MP File are present in the Relationship File, *with the exception of deprecated MPs*
* All NTPs present in the Relationship File are represented in the NTP File
* All NTPs present in the NTP File are present in the Relationship File, *with the exception of deprecated NTPs*
* All TMs present in the Relationship File are represented in the TM File
* All TMs present in the TM File are present in the Relationship File, *with the exception of deprecated TMs*

#### Results

Report (any inconsistencies) as statements (unless there are more than a handful, in which case use a worksheet).

##### SQL used:

FOR MP CODES

SELECT \* FROM [Relationship table new] as relationship

WHERE NOT EXISTS (

SELECT \* FROM [MP table new] as mp

WHERE mp.mp\_code = relationship.mp\_code

)

SELECT \* FROM [MP table new] as mp

WHERE NOT EXISTS (

SELECT \* FROM [Relationship table new] as relationship

WHERE mp.mp\_code = relationship.mp\_code

)

==========================================

FOR NTP CODES

SELECT \* FROM [Relationship table new] as relationship

WHERE NOT EXISTS (

SELECT \* FROM [NTP table new] as ntp

WHERE ntp.ntp\_code = relationship.ntp\_code

)

SELECT \* FROM [NTP table new] as ntp

WHERE NOT EXISTS (

SELECT \* FROM [Relationship table new] as relationship

WHERE ntp.ntp\_code = relationship.ntp\_code

)

==========================================

FOR TM CODES

SELECT \* FROM [Relationship table new] as relationship

WHERE NOT EXISTS (

SELECT \* FROM [TM table new] as tm

WHERE tm.tm\_code = relationship.tm\_code

)

SELECT \* FROM [TM table new] as tm

WHERE NOT EXISTS (

SELECT \* FROM [Relationship table new] as relationship

WHERE tm.tm\_code = relationship.tm\_code

)

### Duplication check

The following is a proxy for a row duplication check, looking at the triple of “mp, ntp and tm code”:

Run each of the following two sets of SQL, noting the number for each. If they are the same, there are no duplicate rows.

If they are different, this will need temp tables/views to query; this has not been needed to date.

##### SQL used:

SELECT mp\_code, ntp\_code, tm\_code from [Relationship table new]

==========================================

SELECT distinct mp\_code, ntp\_code, tm\_code from [Relationship table new]

## Devices NTP File Check

### File compare

There should be **no** changes at all in the file at this time. There may be in the future. Check using the Comparator tool.

##### SQL used:

SELECT \*

FROM [NTP Device new] as [ntp\_device]

WHERE NOT EXISTS (

SELECT \* FROM [NTP Device old] as [ntp\_device\_old]

WHERE [ntp\_device].[device\_ntp\_code] = [ntp\_device\_old].[device\_ntp\_code]

AND [ntp\_device].[device\_ntp\_formal\_name] = [ntp\_device\_old].[device\_ntp\_formal\_name]

AND [ntp\_device].[device\_ntp\_en\_description] = [ntp\_device\_old].[device\_ntp\_en\_description]

AND [ntp\_device].[device\_ntp\_fr\_description] = [ntp\_device\_old].[device\_ntp\_fr\_description]

AND [ntp\_device].[device\_ntp\_status] = [ntp\_device\_old].[device\_ntp\_status]

AND [ntp\_device].[device\_ntp\_status\_effective\_time] = [ntp\_device\_old].[device\_ntp\_status\_effective\_time]

)

Then re-run query, reversing old and new.

Note: there have been issues between ntp-device and ntp\_device; the content in the file should use ntp\_device.

## Coded Attribute File Check

### File compare

There should be **no** changes at all in the file at this time. There may be in the future. Check using the Comparator tool.

##### SQL used:

SELECT \*

FROM [Coded data new] as [coded\_data]

WHERE NOT EXISTS (

SELECT \* FROM [Coded data old] as [coded\_data\_old]

WHERE [coded\_data].[attribute] = [coded\_data\_old].[attribute]

AND [coded\_data].[attribute\_code] = [coded\_data\_old].[attribute\_code]

AND [coded\_data].[attribute\_value] = [coded\_data\_old].[attribute\_value]

AND [coded\_data].[ccdd\_file\_type] = [coded\_data\_old].[ccdd\_file\_type]

AND [coded\_data].[attribute\_status] = [coded\_data\_old].[attribute\_status]

AND [coded\_data].[attribute\_status\_effective\_time] = [coded\_data\_old].[attribute\_status\_effective\_time]

)

Then re-run query, reversing old and new.

## Special Groupings File Check

### File compare

Using Comparator tool, compare Release Candidate file to Previous Month’s Publication Release to find

* Additions
* Changes
* Deletions – **there should be none\***

#### Results

* Concept additions should be as expected; these should be checked by a HC SME
* Concept changes should be rare and should be checked by an SME; they can only be
* minor changes to concept formal names

##### SQL used:

SELECT \*

FROM [SG File new] as [grouping]

WHERE NOT EXISTS (

SELECT \* FROM [SG File old] as grouping\_old

WHERE [grouping].[ccdd\_code] = grouping\_old.[ccdd\_code]

AND [grouping].[ccdd\_formal\_name] = grouping\_old.[ccdd\_formal\_name]

AND [grouping].[ccdd\_type] = grouping\_old.[ccdd\_type]

AND [grouping].[policy\_type] = grouping\_old.[policy\_type]

AND [grouping].[policy\_reference] = grouping\_old.[policy\_reference]

AND [grouping].[special\_groupings\_status] = grouping\_old.[special\_groupings\_status]

AND [grouping].[special\_groupings\_status\_effective\_time] = grouping\_old.[special\_groupings\_status\_effective\_time]

)

order by ccdd\_code

Then re-run query, reversing old and new.

### Code Consistency

All MPs present in the Special Groupings File are represented in the MP File

All NTPs present in the Special Groupings File are represented in the NTP File

All TMs present in the Special Groupings File are represented in the TM File

##### SQL used:

SELECT \* FROM [SG File new] as grouping

WHERE NOT EXISTS (

SELECT \* FROM [MP File new] as mp

WHERE mp.mp\_code = grouping.ccdd\_code

)

AND NOT EXISTS (

SELECT \* FROM [NTP File new] as ntp

WHERE ntp.ntp\_code = grouping.ccdd\_code)

AND NOT EXISTS (

SELECT \* FROM [TM File new] as tm

WHERE tm.tm\_code = grouping.ccdd\_code

)

### Duplication Check

No duplicate rows in the file:

##### SQL used:

SELECT [ccdd\_code], count ([ccdd\_code]), count (policy\_type)

FROM [SG File new]

group by [ccdd\_code], policy\_type

having count ([ccdd\_code]) >1

## Miscellaneous Specific Checks

These are written on an ad hoc basis as particular issues arise. For example, when a product concept that has previously been released has somehow been missed in the generation and needs to be added in, it is sensible to query for it explicitly.

* Concept changes do happen and must be checked by an SME; they can only be
* changes to mp\_formal\_name; these should be the same as in the mp\_file
* Health Canada product name changes; these should mirror the brand name changes above
* Any change in association between the mp\_code and the Health Canada identifier must be carefully checked

There are also separate queries that can be run to check specifically the names associated with the codes, but in effect this is already covered by the differences queries above.