



Cúram SPM Platform

Guidelines for Archiving CER Case Determination Data

6.0.4.0, 6.0.5.0

Author: Clodagh Hannon
Owner: Cúram SPM Platform Group

Dated 24 October 2013 - Version 1.0 Draft 1
IBM Confidential - © IBM Corporation 2013

Document Control Information

Revision Status: 2.0

Document Author: Clodagh Hannon

Document Owner: Cúram SPM Platform Group

Summary of Changes

To request a change to this document, contact the Document Author or Owner. The Document Author or Owner typically allows appropriate designees to update this document.

Changes to this document are summarized in the following table in reverse chronological order (latest version first).

Revision	Date	Created by	Short Description of Changes
1	[14/05/2013]	Clodagh Hannon	Initial Version
2	[24/10/2013]	Clodagh Hannon	Added section (4.3)

Contents

1	Introduction.....	3
1.1	Scope.....	3
2	CER Case Determination Details Overview.....	3
3	CER Case Determination tables.....	4
3.1	Tables, Key Attributes and Descriptions.....	4
3.2	Relationships Between Tables.....	5
4	Identification of Table Rows for Archiving.....	5
4.1	Initial CER Case Determination selection.....	5
4.2	CreoleCaseDetermination.....	5
4.3	Filter CreoleCaseDeterminations that should not be archived.....	6
4.4	CreoleCaseDecision.....	7
4.5	CreoleCaseDeterminationData.....	7

1 Introduction

The purpose of this document is to provide high level guidance, to assist customers on archiving CER Case Determination records from the application. The document will describe briefly the tables associated with these records, the relationships between them and the order in which they must be archived.

1.1 Scope

A fully tested archiving process is not provided with IBM Cúram Social Program Management. As such, the purpose of this document is to provide guidance on a best effort basis to assist a customer implementing such a strategy.

This document only describes the tables released as part of the IBM Cúram Social Program Management product v6.0.4.0. It does **not** describe any customizations made by the customer that would have to be taken into account when archiving such data.

Any data archiving should be first completed on a test system and a complete suite of test cases executed before being applied to a live system.

2 CER Case Determination Details Overview

Note: For a full description of determination storage, and the various factors that can effect this, see the 'How Determinations Are Stored' section in the 'Inside Cúram Eligibility and Entitlement Using Cúram Express Rules' guide.

Once a determination result has been calculated, the system consults the Determination Comparison Strategy for the product before deciding if the result should be stored. If it should be then a single row is added to the **CreoleCaseDetermination** table which includes all data relevant to the determination and links to rows on the child table, **CreoleCaseDeterminationData**, where full details of the determination result, and if configured, a snapshot of all CER rule objects used in the calculation of the result are stored.

The **creoleSnapshotData** attribute on **CreoleCaseDeterminationData** can be quite large; in certain cases it is necessary to spread this data over more than one row. The **overflowID** attribute is used to chain related rows together, by storing the identifier of the overflow row.

In addition to **CreoleCaseDetermination** and **CreoleCaseDeterminationData**, eligibility and entitlement information is also stored on the **CaseDecision** table and its child tables, which are used by Cúram Financials processing to generate financial components. **CaseDecision** rows are linked to a **CreoleCaseDetermination** row by an entry on the **CreoleCaseDecision** table.

When new assessment determinations are stored, the system will firstly supersede the current stored CreoleCaseDetermination record (if one exists) and then supersedes any CaseDecision rows linked to the determination.

These records are considered to be logically deleted. However, they can provide valuable information about the work previously done by the application, and so they are not physically deleted.

Caveats

Note that this document does **not** deal with the archiving of the Case Decision tables used by Cúram Financials processing. The set of determination related tables can be safely archived without impacting financial processing by following the guidelines below.

Customers should properly assess the impact to traceability, and also the impact to any items visible through the application before determining the criteria they will use to select records for archiving. For example, CreoleCaseDetermination records are used to populate the out of the box Determination History screen. Any archived items will no longer appear on this list screen.

That said, once satisfactory criteria have been established, these records can be safely archived to reduce storage and increase performance of the related tables and indexes.

3 CER Case Determination tables

There are three tables related to the storing of determinations. When archiving records from these tables, it should be noted that there are foreign key relationships between them, so the order in which the tables should be dealt with needs to be carefully considered.

3.1 Tables, Key Attributes and Descriptions

Table Name	Key Attribute	Description
CreoleCaseDetermination	creoleCaseDeterminationID	The determination of the varying eligibility, entitlement and explanation of a case for a CER-enabled product.
CreoleCaseDeterminationData	creoleCaseDeterminationDataID	A block of XML data representing a determination result or a rule object snapshot taken as part of a determination.
CreoleCaseDecision	caseDecisionID	The details of a case decision for a CER-enabled case.

3.2 Relationships Between Tables

The majority of the data relating to determinations is stored on the CreoleCaseDetermination table. However before a row from this table is archived, any related rows on other tables must be dealt with first to maintain database integrity. The table below shows the relationships between CreoleCaseDetermination and the associated database tables. (Note that a CreoleCaseDeterminationData record can be a child of a record on the same table if it is an overflow entry.)

Parent Table Name	Foreign Key Attribute	Child Table Name
CreoleCaseDetermination	determinationResultDataID	CreoleCaseDeterminationData
CreoleCaseDeterminationData	ruleObjectSnapshotDataID	CreoleCaseDeterminationData
CreoleCaseDecision	creoleCaseDeterminationID	CreoleCaseDetermination

4 Identification of Table Rows for Archiving

The starting point for identifying the full list of table rows for archiving is to retrieve the list of CreoleCaseDetermination records. There are various attributes which could be used to filter the list of these records depending on your implementation. The example detailed below uses the **determinationDateTime** field, but customers are not limited to this and can filter based on other attributes e.g. assessmentReason, type, depending on the business requirement.

Once the initial list of CreoleCaseDeterminations is selected, associated child or other related records must first be archived to avoid issues with foreign key constraints.

The provided SQL is merely a sample to indicate how to identify the specific rows that should be archived, and should be adjusted as appropriate for the customer's own implementation and specific business needs.

4.1 Initial CER Case Determination selection

The first table to examine is the **CreoleCaseDetermination** table, as this contains the header information and links to all the child tables.

The following sample SQL queries are used to identify the list of records to be archived, to examine the impact to other tables, and also to determine the order with which the data must be archived. For the purpose of these examples, dummy values for the record keys have been used.

4.2 CreoleCaseDetermination

```
SELECT creoleCaseDeterminationID, determinationDateTime
FROM CreoleCaseDetermination
WHERE caseID='111';
```

This query will return the full list of identifiers for each determination for the specified case along with the date/time it was created. The returned items can then be examined and reduced further so that it only contains the determinations to be archived. For example, the list could be further filtered by only selecting records which were created in years before the current year.

```
SELECT creoleCaseDeterminationID
FROM CreoleCaseDetermination
WHERE caseID='111'
AND determinationDateTime < '2013-01-01 00:00:00.0';
```

Using queries such as those above will help to produce the final list of CreoleCaseDetermination records to be archived, but before this archiving is carried out, the impact to other tables and application screens must be examined.

4.3 Filter CreoleCaseDeterminations that should not be archived

To avoid breaking any links on application screens certain CreoleCaseDetermination records should not be archived. These include any records which resulted from case events and payment generations.

Case Events

Records resulting from Case Approvals/Rejections can be retrieved using the following query:

```
SELECT CreoleCaseDeterminationID
FROM CreoleCaseDetermination
WHERE CaseID = '111'
AND ((SnapshotReason = 'CSDR1')
OR (SnapshotReason = 'CSDR2')
OR (SnapshotReason = 'CSDR3'));
```

Records resulting from Case Activations can be retrieved using:

```
SELECT CreoleCaseDeterminationID  
FROM CreoleCaseDetermination  
WHERE CaseID = '111'  
AND ((AssessmentReason = 'CADR1') ;
```

Any records retrieved by these queries should be filtered out from the final CreoleCaseDetermination list.

Case Payments and Reassessments

For each CreoleCaseDetermination record remaining in the list established in section 4.2 and by running the queries above, get the associated caseDecisionID:

```
SELECT caseDecisionID  
FROM CreoleCaseDecision  
WHERE creoleCaseDeterminationID = '222';
```

The following queries should be run for each CaseDecision to determine which to filter out.

```
SELECT Count(*)  
FROM CaseDecisionFinancialComp  
WHERE CaseDecisionID = '999';
```

```
SELECT Count(*)  
FROM ReassessmentDecisionInfo  
WHERE OldDecisionID = '999' or ReassessedDecisionID = '999'
```

If either of the above returns a result greater than zero, then the CaseDetermination needs to be filtered out of the list as underlying functionality depends on it.

The steps in the remaining sections below should be followed for each creoleCaseDetermination record that is still in the list.

4.4 CreoleCaseDecision

The next table to be considered is the CreoleCaseDecision table. This is a simple link table so selecting the rows to be archived is straightforward.

```
SELECT caseDecisionID  
FROM CreoleCaseDecision  
WHERE creoleCaseDeterminationID = '222';
```

Each of the rows retrieved by the above query can be safely archived.

4.5 CreoleCaseDeterminationData

For each row of the CreoleCaseDetermination table that is to be archived, any rows on child tables must firstly be archived. This section deals with child rows for the determinationResultDataID and ruleObjectSnapshotDataID attributes.

```
SELECT determinationResultDataID, ruleObjectSnapshotDataID  
FROM CreoleCaseDetermination  
WHERE creoleCaseDeterminationID = '222';
```

The above query will retrieve the identifiers for the rows on the CreoleCaseDeterminationData table that need to be archived. However, as with CreoleCaseDetermination, the impact to other rows of archiving these needs to be checked first.

Impact of archiving CreoleCaseDeterminationData rows

For each row on the CreoleCaseDeterminationData table to be archived, check if this is also a child row for any other determinations.

```
SELECT count (*)  
FROM CreoleCaseDetermination  
WHERE caseID='111'  
AND determinationResultDataID = '555'  
AND creoleCaseDeterminationID != '222';
```

If the above query returns a count of more than zero then this row cannot yet be archived.

The same query should be run for the result snapshot, as below, and again if this count returns a value of more than zero, then the row cannot yet be archived.

```
SELECT count (*)
FROM CreoleCaseDetermination
WHERE caseID='111'
AND resultSnapshotDataID = '666'
AND creoleCaseDeterminationID != '222';
```

Where either of these queries have returned a count of more than zero, then these rows should not be archived yet as they are used on other CreoleCaseDetermination rows. The originally selected row from CreoleCaseDetermination can now be safely archived. Once this is done, it is safe to move on to the next CreoleCaseDetermination record on the archive list and begin examining the relationships for that record (the step described in section 4.4).

Checking CreoleCaseDeterminationData table for overflow rows

Where the associated row on CreoleCaseDeterminationData is not a child row for another determination record, i.e. the previous query returned a count of zero, then this can be safely archived. However, before archiving the overflowID field needs to be checked. If this field is populated then that row needs to be processed before the current one is.

To check if the overflowID is populated simply run:

```
SELECT overflowID
FROM creoleCaseDeterminationData
WHERE creoleCaseDeterminationID = '555';
```

This query will return either the unique identifier for another row on the CreoleCaseDeterminationData table or else it will return a value of *null*. Where the overflowID is populated then this row in turn needs to be checked in the same way, until a value of null is returned.

When archiving these rows it is important that this proceeds in the correct order, with the row farthest down the overflow chain being removed first.

Graphical View

Below is a summarized table view of the rows used in the sample queries above. The column marked # denotes the order in which the records would be archived following the steps in this document.

CreoleCaseDetermination				
CCDetID	CaseID	DetResult	ROSnapshot	#
222	111	555	666	2
333	111	555	666	7
.....	

CreoleCaseDecision		
CCDecID	CCDetID	#
444	222	1
888	333	3
.....	

CreoleCaseDeterminationData		
CCDetDataID	OverflowID	#
555	null	4
666	777	6
777	null	5
.....	