Allianz Global Corporate & Specialty

AGCS

OPMS to IT Platform

Migration & Development Approach

**Table of Contents**

[1. Overview 4](#_Toc484780579)

[1.1. Introduction 4](#_Toc484780580)

[1.2. Scope 4](#_Toc484780581)

[1.3. References 5](#_Toc484780582)

[1.4. Relevant Links 5](#_Toc484780583)

[2. Migration & Development Approach 5](#_Toc484780584)

[2.1. Migration & Development Types and Phases 5](#_Toc484780585)

[2.2. Back End Migration 5](#_Toc484780586)

[2.3. Front End Development 6](#_Toc484780587)

Change History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change | Date | By |
| 1.0 | First draft | 06/09/2017 | Thomas G. Tuhtan |
|  |  |  |  |
|  |  |  |  |

Distribution & Sign-Off Signatories

*<<Complete with stakeholder details relevant to the competency centre or project>>*

| Name | Role | Responsible | Accountable | Consulted | Informed |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Approval Signatories

*<<Complete with stakeholder and approver details relevant to the competency centre or project>>*

| Name | Role | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |

1. Overview
   1. Introduction

This Project Migration & Development Approach document has been created to provide a description of the approach for all of the software development and database migration work to be undertaken for the OPMS to IT Platform project.

AGCS recognises the need for consistency in the application of software development across all software deliveries. All application development must provide risk mitigation, be cost effective and optimise the quality of delivered products.

* 1. Scope

This Migration & Development Approach document is applicable to the software development activities carried out on 21 OPMS systems comprising of 12 AGCS systems & nine (9) legacy FFIC systems. Each of these 21 systems consists of a separate front end database for the presentation layer and a back end database for the storage of transaction and reference data.

The following systems items are ***included*** in the scope of this project:

**AGCS OPMS Systems (12 systems)**

|  |  |  |
| --- | --- | --- |
| System Name | Front End Database File | Back End Database File |
| US Financial Lines | Commercial.mdb | Commercial\_be.mdb |
| Finance | Billing and Collections OPMS.mdb | Billing and Collections OPMS.be.mdb |
| CAT Management | CAT OPMS V1.1.mdb | CAT OPMS V1.1.be.mdb |
| Aviation | Aviation.mdb | Aviation.be.mdb |
| Inland East | Inland East.mdb | Inland East.be |
| Inland South | Inland South.mdb | Inland South.be |
| Inland West | Inland West.mdb | Inland West.be |
| Canada Commercial | Can Commercial.mdb | Can Commercial.be.mdb |
| Hull & Cargo | Hull and Cargo.mdb | Hull and Cargo.be |
| Aviation Submissions | Sub AT Box.mdb | Sub AT Mailbox.be.mdb |
| Marine Submissions | Sub AT Box.mdb | Sub AT Box.be.mdb |
| Corporate Submissions | Sub AT Corp.mdb | Sub AT Corp.be.mdb |

**FFIC Legacy OPMS Systems (9 systems)**

|  |  |  |
| --- | --- | --- |
| System Name | Front End Database File | Back End Database File |
| Entertainment Submissions | TMG ENT OPMS.mdb | TMG ENT OPMS\_be.mdb |
| Entertainment | ENT OPMS.mdb | ENT OPMS\_be.mdb |
| CI Billing | CI BILLING OPMS.mdb | CI BILLING OPMS\_be.mdb |
| CAR | CAR.mdb | CAR\_be.mdb |
| Commercial | Commercial.mdb | Commercial\_be.mdb |
| Premium Entry | Premium Entry OPMS.mdb | Premium Entry OPMS\_be.mdb |
| Professional Liability | Professional Liability.mdb | Professional Liability\_be.mdb |
| Programs | Programs.mdb | Programs\_be.mdb |
| Unit Stat | Unit Stat.mdb | Unit Stat\_be.mdb |

The following work items are ***included*** in the scope of this approach:

* Back end database migration from Access to SQL Server
* VBA Front end references to back end data changed from Access to SQL Server
* Incorporating 12 tables of the db\_Management.mdb into the Inland East back end database
* Developing a tool to facilitate the modification of back end database migration scripts
* Developing a tool to facilitate scanning of the VBA code of the front end databases
* Developing a tool to facilitate the extraction of properties contained on Access tables not transferable to SQL Server
* Defining backup / restore scripts to create monthly snapshots of the SQL Server databases to facilitate reporting
* Defining an update script to be run daily to update certain records of the CAT Management database
* FMS setup where not currently in place (FFIC databases CAR, Commercial, Professional Liability and Programs)

The following work items are explicitly ***excluded*** from the scope of this strategy:

* Adding SSO to any OPMS systems without it (currently not present on AGCS OPMS systems US Financial Lines and Finance, FFIC OPMS systems CAR, Commercial, Professional Liability, Programs and Unit Stat)
* Adding any enhancements to or fixing any production bugs of any of the OPMS systems
  1. References

The table below provides details of referenced and supporting documentation used to provide input and context in the OPMS to IT Platform Migration & Development Approach.

|  |  |
| --- | --- |
| Document Name | Description |
| Various | Various OPMS reference materials stored in the OPMS to IT Platform Project Sharepoint site. |

* 1. Relevant Links

OPMS to IT Platform Project sharepoint site: <https://sp-11-003.portal.allianz.com.awin/sites/IT/OPMS/Shared%20Project%20Documents/Forms/AllItems.aspx>

1. Migration & Development Approach
   1. Migration & Development Types and Phases

Each of the 21 OPMS systems have previously been separated into two .mdb files, one for the “front end” (primarily consisting of UI forms and reports) and one for the “back end” (storing of the UI entered transactions and reference table). The end result of this project will be to have moved the back end databases of each OPMS system from an Access database hosted on a shared file server to a SQL Server database hosted on a dedicated IT supported database server. The presentation layers of the front end databases will not have any perceptible changes to the end users, but development will be required on each front end to enable the back end migration. In order to use SQL Server for the transactional and reference data stored in the back databases, must be converted (migrated) to SQL Server, and all references to that data from the front end must be changed to both point to the new database and also use the appropriate corresponding values, functions and queries for SQL Server vs Access.

Therefore, there will be two distinct software development activity types: back end database migration & front end development. The back end database migration of each system must be performed first, followed by the corresponding front end development.

* 1. Back End Migration

**Purpose:** Move the structure and contents of a specific OPMS system’s back end .mdb database file to its own MS SQL Server database

Back End Migration Approach:

For each back end:

1. Make a copy the production .mdb file to the developer’s local workstation
2. Scrub the contents of that copy with output to a new .mdb database also located on the developer’s local workstation by fixing any data values that will not migrate to MS SQL Server (e.g., dates older than 01-01-1753 set to 01-01-1753) and/or making required structural changes (e.g., adding an identity column to any reference table that can be changed from a front end)
3. Additionally, from the db\_Management.mdb, move the following tables (only applies to Inland East back end):

[Database Paths]

[MultiColumnSortOverrides]

[Offices]

[Pre-fill Hull and Cargo]

[Pre-fill Inland Marine]

[Regions]

[Small Account UW]

[Table Transaction Days IM by Received Date]

[Table Transaction Days Inland Marine]

[Table Transaction Days Validation Inland Marine]

[User Names Hull Cargo]

[User Names Inland Marine]

1. Use the 32-bit version of SQL Server Migration Assistance (SSMA) version 2012 to migrate the scrubbed .mdb file to a **local** SQL Server database
   1. Do not attempt to create the database on logical D: drive, as it will fail
   2. Use dbo schema
   3. Suppress automatic creation of SSMA\_Timestamp column
   4. Convert datatype datetime2 to datetime
   5. Convert datatype TEXT to Varchar(Max) or Varchar(length)
   6. Remove all indexes automatically created by SSMA process by using the special script
   7. Add appropriate indexes (e.g. [Processed Items] table)
2. Backup the local SQL Server database
3. Request that the backup be restored to the non-production SQL Server database server
4. Compare the production .mdb database copy to the scrubbed .mdb database using AQT. The only content differences should be items identified as being fixed. The only column differences should be identified structural changes.
5. Compare the scrubbed .mdb database to the migrated SQL Server database using AQT. There should be no content or column differences.
   1. Front End Development

**Purpose:** Modify front end references to data stored in back end databases from Access databases to SQL Server databases.

Front End Development Approach:

For each front end database:

* Extract all queries to a single file using the special extraction tool, specifically created for this project
* Make and document the following VBA changes:

1. Correct DAO Updatable Recordsets

Add required option "dbSeeChanges" to DAO recordsets opened for editing

Example:

Set rst = db.OpenRecordset("Items Processed", dbOpenDynaset, dbAppendOnly)

to

Set rst = db.OpenRecordset("Items Processed", dbOpenDynaset, dbAppendOnly + dbSeeChanges)

1. Boolean: -1 = True in MS Access, 1 = True in SQL Server

Change “-1” to "True" in VBA code and queries if used as an equivalent for True

Reverse sort order on any queries that sort on a Boolean column

1. Remove Time from Date/Time Fields

In MS Access, the INT function will remove the 'time' aspect for a Date/Time field, but this will result in an error when the data is stored in SQL Server. To resolve this, replace “INT” with DATEVALUE in VBA code and queries if used to remove the time from Date/Time fields. Also, references to Date/Tim fields that simply ignore the time component must be changed to use the Datevalue function to only return the date component.

Example:

Between Fromdate and Todate

to

Between Datevalue(Fromdate) and Datevalue(Todate)

1. Update References to db\_Management.mdb (Applies to 3 Submissions and 5 Marine databases)

Twelve (12) tables of the linked db\_Management.mdb database will be moved to the Inland East back end database. All such references must be changed to point to the Inland East back end database. The tables are:

[Database Paths]

[MultiColumnSortOverrides]

[Offices]

[Pre-fill Hull and Cargo]

[Pre-fill Inland Marine]

[Regions]

[Small Account UW]

[Table Transaction Days IM by Received Date]

[Table Transaction Days Inland Marine]

[Table Transaction Days Validation Inland Marine]

[User Names Hull Cargo]

[User Names Inland Marine]

1. Add Inheritable Properties to the Linked Tables

Columns on MS Access tables can have properties that columns on SQL Server tables cannot. All of the properties of a table column in MS Access are inheritable. For example, the [User Name] column on the [Items Processed] table in the Commercial database has this property:



Use the tool created for this process to extract all inheritable properties so they can be added back. Populate a table with the necessary properties. For example to add this property, the table would look like:



* Configure Front End to use SQL Server Back End

1. Use ODBC form in front end to point back end connection to SQL Server database migrated above
2. Confirm connection changes works by making creating or updating simple transaction stored in back end

* Make New FMS as Needed (FFIC databases CAR, Commercial, Professional Liability and Programs)
* Unit Test
  1. Test all active features to ensure they are able to read/write to the SQL Server database
  2. Create corresponding documentation of the unit test and its results