WP.13
ENGLISH ONLY

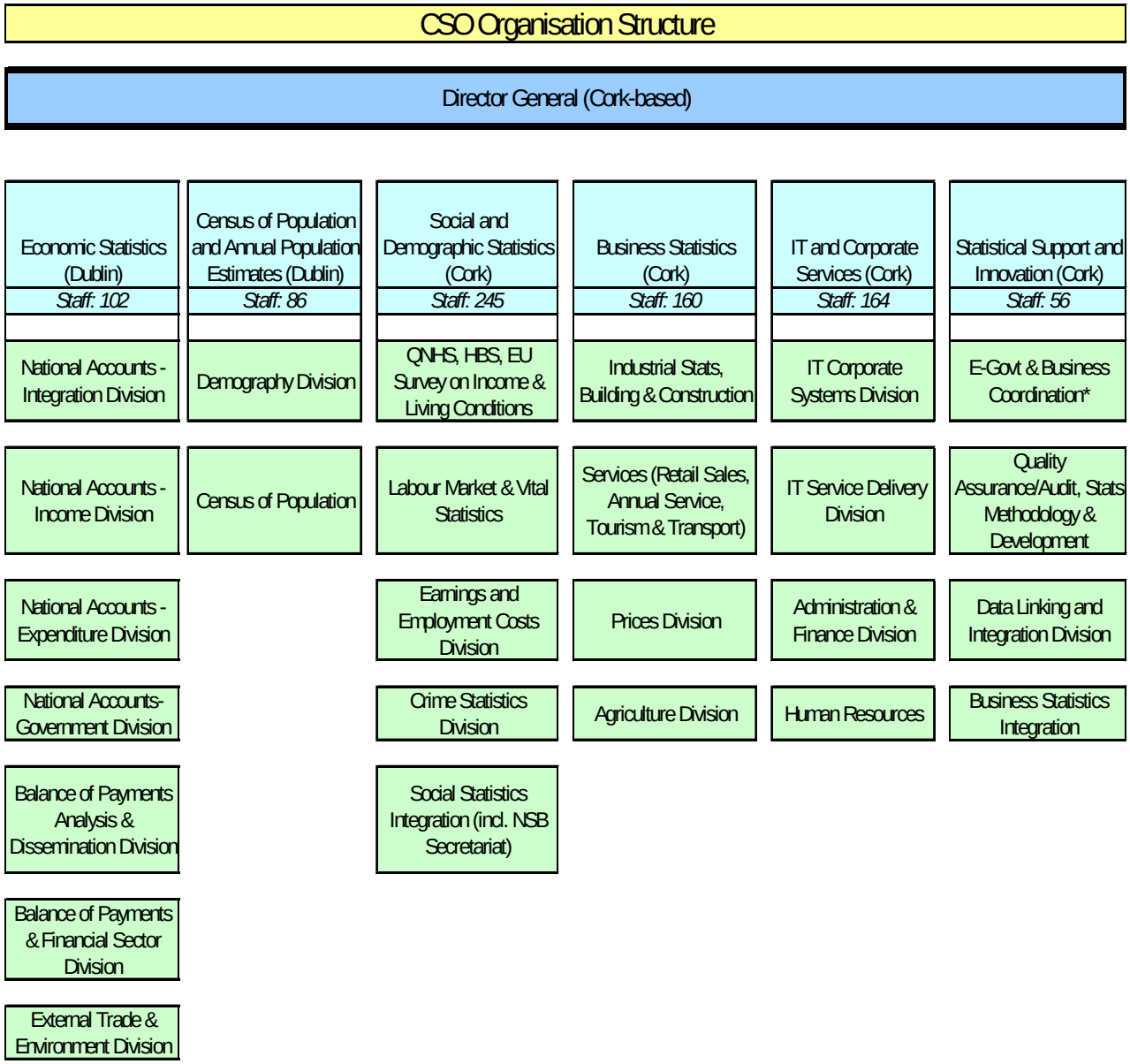
**UNITED NATIONS STATISTICAL COMMISSION**
**ECONOMIC COMMISSION FOR EUROPE**
**CONFERENCE OF EUROPEAN STATISTICIANS**

**UNECE Workshop on the Common Metadata Framework**
(Vienna, Austria, 4-6 July 2007)

# CASE STUDY - IRELAND[1]

## 1. INTRODUCTION

**Organisation Details** Organisation Name: Central Statistics Office (CSO)
Contact: Margaret McLoughlin.
e-mail: margaretmcloughlin@cso.ie




*includes some electronic data collection and web dissemination.

[1] Prepared by Mairead Coughlan, Mairead Coughlan@cso.ie and Margaret McLoughlin, Margaret.McLoughlin@cso.ie

## Overall Strategy

1. The CSO complied I.T. and a Data Management Strategies in 1999. The strategies envisioned a two-phase implementation.

2. The first phase consisted of an infrastructure upgrade to client-server hardware, which was successfully completed in 2001. The infrastructure upgrade consisted of both IT and statistical elements. The IT element included the rollout of group-ware (Lotus Notes) throughout the organisation, which has facilitated document management and considerably improved internal communications. The statistical elements included a new central business register system (CBR) and a classification server (CARS).

3. The second phase, known as the Information Technology Strategic Implementation Programme (ITSIP), includes re-engineering all CSO's software applications, to facilitate the migration from the VAX Open VMS environment. The overall aim of the programme is to establish a framework from which it will be possible to improve and extend the services supplied by the Office to its customers and respondents and provide enhanced service to the public through the availability of online interactive access to statistical data.

## Information Technology Strategic Implementation Programme – Objectives and Goals Obtained

4. The fundamental goals of the ITSIP project are to:
   - Deliver a set of applications to meet the business and customer needs of the CSO, including survey and respondent management systems and a data architecture supporting the applications and the Office's data management strategy, including a dissemination database containing all publishable CSO data.
   - Develop a technology infrastructure, which will make applications available to all members of staff and will allow external users to use the selected elements of the applications delivered as e-Government services via the Internet.
   - Migrate and reengineer its existing DEC Alpha-based legacy systems to a client server environment
   - Implement the reengineered systems within the parameters of the CSO Corporate Data Model and its associated input, clean unit record, aggregate and disseminate databases and metadatabases and produce the applications necessary to support this
   - Interface the solutions proposed with existing client server systems and Sybase databases, such as the CBR and CARS.

5. This should result in an IT environment which will enable the Office to continue to develop its organisational capability in the context of the wider Information Society and e-Government effort and demands.

6. The main output of the ITSIP project is the Data Management System (DMS) which is scheduled to "GoLive" this September. The system is metadata driven which will enable a move from the current stove type approach to a generic survey processing system. The CSO currently has approximately 150 systems written & maintained centrally and 250 end-user applications written & maintained locally. Most of the end user systems are written in SAS, SAS V6.12 on the VAX or PC SAS V8.02. The DMS has consolidated as much common processing as possible from the current processes into a suite of nine corporate applications in order to promote consistency and re-use across the various survey areas. The corporate applications will reside on a corporate database storing all data and metadata required in the survey-processing lifecycle.

7. The ITSIP/DMS project was divided into two stages. The first stage (Stage A) was a six month contract awarded to Accenture, who complied the Requirements Specifications & High Level Architectural Design, and the second stage (Stage B) was awarded to Cognizant Technology Solutions Ltd. (CTS) based in Chennai, India. Stage B is due for completion in September with an additional 18-month support and maintenance to be provided by CTS.

Project review: Stage A project (Oct 02 – Mar 03).

8. The requirements analysis phase took a "bottom-up" approach and Accenture held over twenty workshops with the CSO business users. These resulted in 51 'As Is' process descriptions and 61 process maps. A data model, including the Swedish Data Model for aggregation and dissemination was drawn up. The 44 'To Be' process descriptions were analysed and resulted in the existing processes being consolidated into nine survey processing applications and one security application.

9. The design phase of Stage A produced:
    - High Level Architectural Requirements
    - High Level Architectural Design
    - High Levl Performance Model
    - High Level Interface Requirements and Design Specification
    - Web Enablement Specification

Project Review: Stage B project (Nov 03 –Sept 07)

10. The design was further validated at the beginning of Stage B, and this validated design became the baseline of the DMS functionality against which the project plan was tracked and the change control process defined. Stage B involved the building of the system, the migration of the historic data & integrity metadata from legacy systems and the migration of the metadata from the UAT environment.

11. The original project plan was scheduled for 30 months and it was to conclude in May 2006. However a number of issues have resulted in a delay of 16 months:
    - delay in initial increment deliveries due to new requirements
    - delay in CSO testing due to underestimation of time required
    - extra functionality in the DMS
    - Quality Assurance Control

## Lessons Learned

- Consider carefully your organisation's capacity for insourcing /outsourcing development work
- Consider the time scale for implementation of the solution
- Manage the change process well
- Understand the complexity of the solution and in procurement stage reject very low bids
- Assume external contractor has no knowledge of your business
- Ensure adequate in house skills in IT Design so IT Partner's assumptions can be validated
- Ensure adequate in-house skills in IT Partner's development tools and proposed application infrastructure from start of project
- Don't accept IT Partner's project plan lightly where your office's resources are concerned
- Don't underestimate the resources needed to (1) manage the project and (2) keep abreast of all project documentation
- Consider carefully the items that are for sign-off, review and for information by you - these will have financial implications later
- QA is more important than just ticking boxes but throughout the software development lifecycle should include:
    - reviewing the decisions taken to obtain technical solutions
    - examining the underlying deliverable
    - adherence to agreed standards
- Allocate adequate time to reviewing the test process and test cases
- Have the big picture in view at all times
- Managing the contract requires high-level expert resources with project management, statistical and IT skills
- Organisational support and commitment from top management critical

### Future challenges

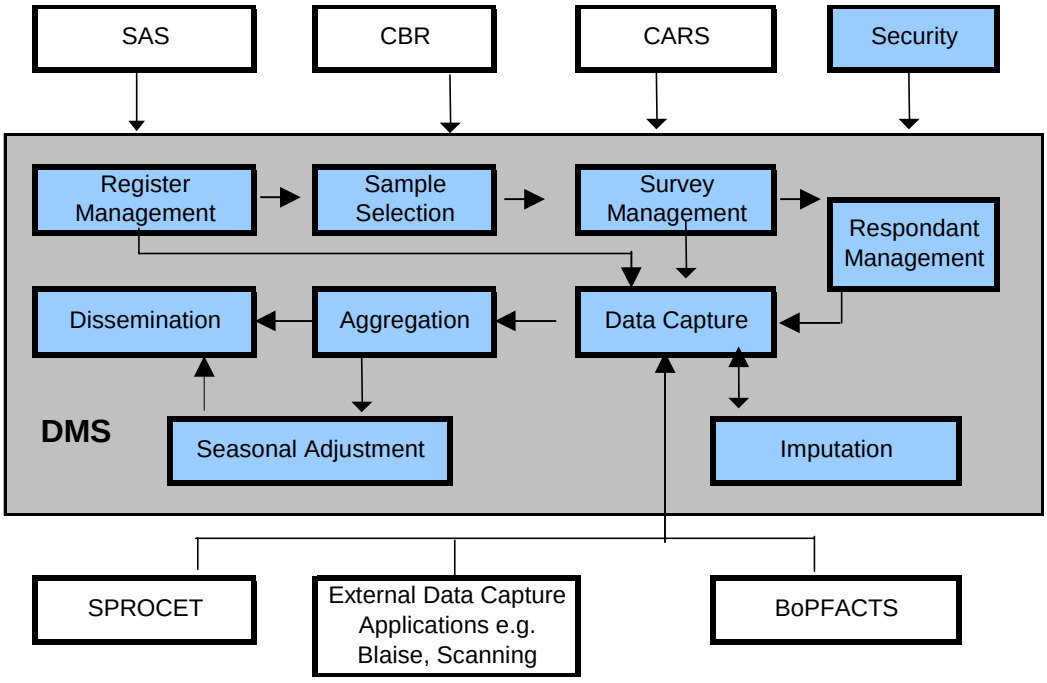
- DMS is to GoLive in September and over the following six months surveys will gradually be moved onto the new system.
- New SAS environment will be required as we move from SAS V6 on the VAX and PC SAS V8.02.
- New IT Strategy is required.

### Basic Metadata Management Principles Used

12. The Swedish Data model is used for Dissemination. Metadata tables are shared across all surveys and, because of concurrency issues with Sybase IQ, all the metadata tables are held in Sybase ASE.

## 2. THE STATISTICAL METADATA SYSTEMS AND THE STATISTICAL CYCLE

**Process Model**




13. The DMS has ten applications as set out in the diagram above: Register Management, Sample Selection, Survey Management, Data Capture, Imputation, Aggregation, Seasonal Adjustment, Dissemination, Respondent Management and Security.

14. The existing organisational systems interacting with the DMS are:
    CBR – Central repository for all sole traders, partnerships, private & public companies and co-operatives engaged in ecomonic activity in the state.
    CARS – Database containing all classifications and concordances.
    SPROCET (Survey Processing Template) – A re-usable survey processing template currently used by the CSO's Industrial surveys.
    BoPFACTS – Data processing system and survey management system used by Balance of Payments section.

15. The process model was developed during Stage A of the ITSIP/DMS project. Each business user documented the survey process for every survey undertaken in the CSO. These processes were analysed by Accenture and the ITSIP team and were consolidated into nine distinct survey applications.

16. The DMS will be the main survey processing tool post-September for the business and household surveys. The survey areas will use the DMS functionality in the different applications, as they require it. SAS may be used for specialised functionality not available in the DMS but the main use of SAS will be for analysis. Each application has at least one entry and exit point for importing and exporting data. However, users will not be allowed to write directly to the database tables.

## DMS Application Functionality

Register Management:

This application allows users to create and maintain registers and allows read-access to the CBR. Unit data can be imported to, and exported from, the DMS registers in a number of different file formats. The Register Management module uses SSA-Names3 to search all the registers and uses CARS to code register variables.

Sample Selection:

This DMS application allows users to select censuses or random samples linked to the Register Management application. The random samples can be stratified using CARS classifications or manually entered stratification groups based on register variables. Sampling proportions can be imported or manually entered. The Sample Selection application interfaces with CARS and the CBR.

Survey Management:

The Survey Management application enables users to maintain a record of the questionnaires and reminders sent to the participants of a survey. Detailed statistics of the respondents and non-respondents of a survey can be viewed. A response code can be entered in this application to record the return of survey forms. This application is linked to the Sample Selection and Register Management applications.

Data Capture:

The Data Capture application can be used to capture data through manual keying or from external files which can be imported in a number of formats: CSV, Delimited, Excel, Fixed Width, SAS, Sybase table and XML. The application allows the user to set up a GUI based on the layout of the survey questionnaire to facilitate manual data entry. After the respondent data is captured, edits, derived variables, coding and imputation are run. Live ‘dirty’ data is stored in the Input database and moved to the Clean Unit database (CUR) once errors have been corrected. The CUR holds many different versions of the data and the aggregation application runs on the CUR data. Within the DMS the Data Capture application interacts with Register Management, Survey Management and Imputation. Data capture also interfaces with CBR, CARS, Trigram and Precision to code variables. Data can be exported from both the Input and CUR databases.

Imputation:

This application comprises five standardised imputation methods, namely Last Value Carried Forward, Nearest Neighbour Imputation, Direct Imputation, Stratum Average Imputation and Time Series Imputation. Multi-stage imputation is possible to ensure the best estimate is obtained. Imputation is dependent on the Data Capture application and also utilises CARS.

Aggregation:

The Aggregation application enables the user to define the classifications, aggregates, weights, tables and rules required for aggregation. Throughout the DMS, metadata is set up prior to any process being run. The Aggregation application allows the user to set up and run macro edit rules and confidentiality

rules. User can “drill-down” to the unit level once the aggregation has been performed. The aggregation results can be viewed and exported. The Aggregation application interacts with the Register Management, Data Capture and Seasonal Adjustment applications and can access both the CBR and CARS.

Seasonal Adjustment:

The Seasonal Adjustment application depends on the Dissemination and Aggregation applications for the data to be seasonally adjusted. Seasonal adjustment is performed on aggregates (data columns) selected from a disseminate table. The seasonally adjusted estimates generated are used to update seasonally adjusted aggregates in the aggregate table of the unadjusted aggregates. The seasonally adjusted aggregates can then be disseminated through the Dissemination application.

Dissemination:

The Disseminate application is where aggregates that are to be disseminated are stored. The aggregate data that is to be disseminated is copied from aggregate tables to disseminate tables. Only aggregate tables that use standard classifications stored on CARS can be copied to the Disseminate database. Additional metadata is entered here including footnotes, data custodians and contact details. The Disseminate application also features a timeliness monitor and data can be exported in many formats including PC Axis.

Respondent Management:

The Respondent Management application provides the functionality to view the compliance details and data returned by a respondent across survey periods. These details can be exported or printed to facilitate analysis as required. The application is linked to the Register Management, Survey Management and Data Capture applications.

## Mapping the DMS to the CMF Lifecycle Model

| | | |
|---|---|---|
| Register Management | → | Survey Preparation (2) |
| Sample Selection | → | Survey Planning and Design (1) |
| Survey Management | → | Survey Preparation (2) |
| Data Capture | → | Data Collection (3) |
| | → | Input Processing (4) |
| | → | Derivation (5) |
| Imputation | → | Estimation (5) |
| Aggregation | → | Aggregation (5) |
| Dissemination | → | Dissemination (7) |
| Respondant Management | → | Post Survey Evaluation (8) |

17. The DMS is a processing and not an analysis tool, therefore CMF LifeCycle Model “(6) Analysis” cannot be linked to the DMS. In general SAS will be the tool of choice for analysis in the CSO. Data can be exported from the DMS from every application in many formats whenever nessecary.

## 3. STATISTICAL METADATA IN EACH PHASE OF THE STATISTICAL CYCLE

### Application Metadata

18. The system has been programmed to perform many statistical processes. However, to customise the system to suit an individual survey the user must enter his/her application metadata. This means that the system, or the required parts of the system, will be tailored to each individual survey.

19. This application metadata is stored in a large array of interconnecting metadata tables in the Sybase database, along with the data, which is also stored in Sybase databases (a mixture of Sybase ASE and Sybase IQ).

### Core Metadata

20. This is the traditional metadata that is minimally required to describe the data. For example, when variables are being set up for keying into the Data Capture application, the unit of measure associated with the variable will be set up too. If the variable is to be coded, the link to the classification will be established here. In the Aggregation module, the groups (typically classification values) must be created and saved.

21. The aggregate methodology (formulae) are constructed in the system and stored. However no methodological documentation is associated with this, but this can be linked in at a later phase.

22. When data is moved to the Disseminate database, additional metadata fields are completed so that the data can be disseminated on the Web. However, the CSO website is currently populated from another database, but the intention is to streamline the DMS so that it will feed directly into this database when the project has been tested fully and any required amendments are made.

23. The way the Aggregation and Dissemination modules have been programmed ensures that groups/classifications have to be used to facilitate the cubed data structure.

### MetaData Inputs

*Metadata tables are shared across the DMS. Below is an example of the input metadata in use.*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Register Name | Lists the Register Names stored in the DMS | R_RSI: The register for the Retail Sales survey. | DMS Table: Register |
| Survey Name | Lists the survey in the DMS | RSI: The Retail Sales Survey | DMS Table: Survey |
| DMSCommonCodes | Holds a list of metadata which may be reused across surveys (See Appendix A) | Data Type: String, Int, Boolean, Varchar, Numeric. | DMS Table: DMSCommonCodes |
| Classifications | Classifications used to code variables in RM. | NACE Rev 1.1 | CARS |
| Survey Periodicity | | Monthly 2007M06 | DMS Table: SurveyInstance |
| Sample Method | | Census | DMS Table: Sample |
| Unit of Measure | | Euro | DMS Table: DMSCommonCodes |
| PostOut Item Type | | Questionnaire or Reminder | DMS Table: DMSCommonCodes |

## Register Management

*Initial Application Metadata Set-ups*

- Create Register
- Define Register variables
- Set-up Register coding

*Basic Functionality*

- Upload Register
- Search the Register & View Possible Dup
- Add & View Unit Details & Upload New Births File
- Export Register
- Run Coding

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Allocated Surveys | Lists the surveys in the DMS allocated to the register in question | RSI: The Retail Sales Survey | DMS Table: SurveyRegisterLink |
| Coding Details | | | CARS |
| Variable Details | Name, DataType, Coded, | Number of Employees, Numeric, Not Coded | DMS Table: M_RegisterVariable |

## Sample Selection

*Initial Application Metadata Set-ups*

- Set the Sample Selection Criteria &
- Define Stratification Groups

*Basic Functionality*

- Import or Key Sampling proportions or sample size
- Run Census or Random Sample
- Exclude or Include Units
- Export the final sample for analysis

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Sample Name | | RSI2006M01 | DMS Table: Sample |
| Sample Details | Sample type, | Sample | DMS Table: Sample |
| | method | Matched with previous respondants | DMS Table: Sample |
| | rules | Include where employment > 5 | DMS Table: SampleRetrievalRules |
| | stratification groups | NACE Rev 1.1 | DMS Table: M_RegisterTableSampleGroup |
| Sample Size | Number of units in the sample | | DMS Table: Sample |

## Survey Management

*Initial Application Metadata Set-ups*

- Set Up Post Out Items

*Main Functionality*

- Choose Post Out Details
- Issue Post-Out units to Print Shop Mail
- Receipt Respondents
- Response Tracking & Follow Up Details

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Post Out Rules | Applicable if survey has multiple forms. | If employee > 5 | DMSTable: QuestionnaireRules |
| Post Out Details | Accesses register for contact details | Company Name, Address, Unit ID | |
| Total Forms Issued | A list of the units receiving each form | | |
| Response Rates | Number of survey forms returned | | |

## Data Capture

*Initial Application Metadata Set-ups*

- Create Data Capture form
- Define variables characteristics (e.g. derived, coded, type, expression)
- Set-up coding rules
- Set up Import Details
- Set-up Edit rules and Validations

*Main Functionality*

- Run Derived Variables, Edit Rules and Validations
- Run coding
- Import data files, Key data
- Take snap-shots of data to the CUR database

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Variable Details | DataType, Name, FieldType, Unit of Measure | Numeric, Turnover, TextBox, Euro | DMSTable: MI_Variable |
| Edit Rule | The edit rules. | Turnover > 0 | DMSTable: EditRuleTerm |
| Edit Failures | List of units failing each edit. | | DMS Table: I_EditFailure |
| Derived variable formulae (period specific) | | | DMSTable: MI_Variable |
| Coding (period specfic) | | | DMSTable: CodingType |

## Imputation

*Initial Application Metadata Set-ups*

- Set Up Imputation Groups
- Set Up Imputation Rules

*Main Functionality*

- Run Imputation & View Results
- Exclude Records

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Imputation Rule | Last Value Carried Forward | If n (t) = '.'then n (t) = n (t-1) | DMSTable: ImpRule |
| Imputation Cell | Stratification groups for imputation | | DMSTable: ImpRule |

## Aggregation

*Initial Application Metadata Set-ups*

- Define Aggregates, Complex Aggregates, Groups & Aggregate Tables
- Set-up Macro Edit Rules & Confidentiality Rules
- Create Weights & Weight Tables

*Main Functionality*

- Run Aggregation on CUR data or other aggregate data
- Define Relative and Index Component Updating
- Re-Scaling and Splicing
- Import Aggregate data & Weight data
- View, Drill-down and Export Aggregation Results

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Aggregate Definition | The expression used to derive the aggregate result | Total Turnover = sum(turnover) | DMS Table: MA_AggregateDefinition |
| Groups | The are the classifications used to group the data for aggregation. | Nace Rev 1.1 | DMS Table: DMSVarClassifications |
| Aggregate Table | The table definition links the classifications and the aggregates. | Total Turnover by Nace. | DMS Table: MA_TableDimensions |
| Macro Edit Rules | Edit rules that are run on the aggregate data | Total Turnover (T) compared with Total Turnover (T-1) | MA_MacroEditRules |

*The version of CUR data used to populate the aggregate tables is stored in the metadata tables.*

## Dissemination

*Initial Application Metadata Set-ups*

- Create disseminate tables
- Define the data column (aggregate) attributes
- Set-up footnotes and link footnotes to tables.

*Main Functionality*

- Copy aggregate data to the disseminate table
- Export disseminate data

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) | Quality issues |
|---|---|---|---|---|
| Data Column Description | This contains all metadata relating to aggregates to be disseminated. | Name, Presentation text, Data Custodian, MeasureUnit, UpdateDate | DMSTable: MD_DataColumn | Ensuring that presentation texts and classifications are to a publishable standard. |
| Footnotes | Footnote links and content | | DMSTable: MD_FootNoteAllText | Ensuring that footnote texts are to a publishable standard. |

## Seasonal Adjustment

*Initial Application Metadata set-ups and Main Functionality*

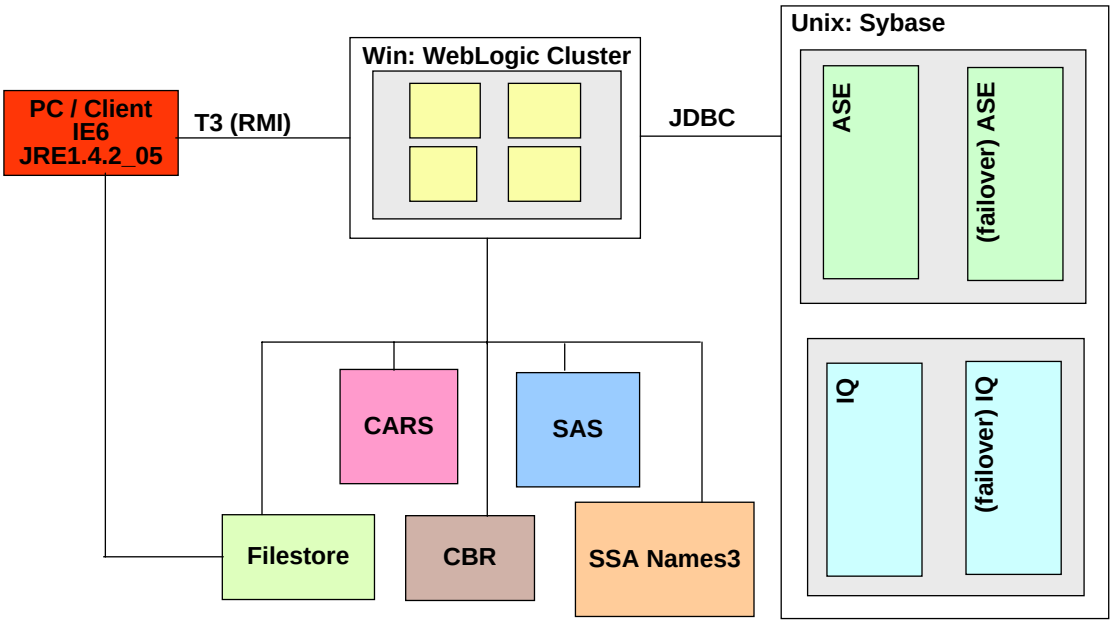
- Setup Seasonal Adjustment
- Run Seasonal Adjustment
- View Seasonal Adjustment Results

*Metadata Outputs*

| Group | Description | Examples | Source (if reused) |
|---|---|---|---|
| Seasonal Adjustment Parameters | | | MA_SeasonAdjust |

## 4. SYSTEMS AND DESIGN ISSUES

### Technical Overview




### System Summary

24. The DMS is a classic 3-tier system, consisting of a Java Client interacting with a J2EE Weblogic Server while the data is stored in Sybase ASE and Sybase IQ databases. All interactions with the databases are controlled through Weblogic.

25. The User interacts with the DMS Filestore through a network drive mapping on their PC. Interaction with all other applications is performed through Weblogic.

### DMS Client

26. The DMS Client is a 'fat client' developed utilising Java Technologies.

27. The DMS is a complex system, and while every effort has been made to simplify the design of the graphical user interface (GUI) it remains a complex GUI to develop programmatically. In the design phase of the project the CSO looked at the possibility of developing the application with a 'thin client' / web based front-end, however the decision was made to develop the GUI in Java JFC/Swing to give developers and users better control over the processing within the system.

28. The DMS Client has been designed to work with a Business Delegate design pattern. All interaction with the J2EE / Weblogic server uses Java RMI protocol. The DMS is designed using Weblogic T3 / RMI Protocol.

29. The original design for the DMS was the use of the Session Façade design pattern however during the initial System Testing and Performance iterations it was found that this was not performing adequately and as a result we moved to the Business Delegate pattern. For more information:
(http://java.sun.com/blueprints/corej2eepatterns/Patterns/BusinessDelegate.html)
(http://java.sun.com/blueprints/corej2eepatterns/Patterns/SessionFacade.html)

30. The CSO has implemented a 'single sign-on' solution for the DMS Client and where by the CSO Active Directory Service (ADS) is interrogated for user authentication. Once the user has been authenticated, the user privileges are then pulled from the DMS Databases and access within the DMS is granted accordingly.

31. The DMS is deployed though the use of Java Web Start Technology which allows both a quick deployment and centrally managed releases to the organisation. This will significantly reduce the administrative burden on the CSO IT Teams in supporting the DMS on the Client PC.

## J2EE Weblogic Server

32. The CSO have implemented the Weblogic8.1 J2EE Server for the mid-tier of the DMS solution. This Weblogic Server provides:
    - Product maturity / stability
    - Performance
    - Excellent customer support
    - Excellent administration interface

33. The DMS utilises the Weblogic Server to host the DMS EJB Container. The Client interacts with the Weblogic Server through the use of either Stateless Session Beans or Java Messaging Service (JMS).

34. Users standard interaction with the DMS is through synchronous transactions. These requests are passed to the DMS Stateless Session Beans. The DMS also facilitates users who want to run jobs in batch mode, asynchronous processing, and these interactions are routed to the JMS Queues where they are processed.

35. All DMS interactions to the Sybase Database Layer are controlled through the use of JDBC Connection Pools defined within the Weblogic Server. User requests requiring access to the Sybase ASE or Sybase IQ databases take an available connection from the pool for the duration of their transaction.

36. The Weblogic servers will work in a clustered environment ensuring that there is equal distribution of work across the available server resources and also provide for failover on the mid-tier.

## Database Layer

37. Prior to commencing the ITSIP project the CSO IT Team had significant experience working with Sybase ASE. In the analysis phase of the project the consultants recommended that the CSO should also employ the use of data warehouse technologies, Sybase IQ, for the aggregation of survey data and as the store for disseminated data.

38. At a high level, the data within the DMS is stored as follows:
    - Register Management, Data Capture (Input), Imputation: **Sybase ASE**
    - Data Capture (CUR), Aggregation, Dissemination: **Sybase IQ**

39. The DMS stores two types of data:
    Survey Metadata - Core DMS Tables
    Survey Specific Data - Survey Tables created from the data definitions stored in the Core DMS Tables.

40. As part of the overall design of the system it was agreed that all complex numerical processing be performed within the database layer through the use of Sybase Stored Procedures. This improves performance as data is processed within the RDBMS and will ease the long-term maintenance of the DMS, which will be undertaken by the CSO Sybase and Java teams.

41. Power users within the DMS will have the facility to read their data directly from the RDBMS. As surveys are created within the DMS, specific database views will be created on the inputed data. The use of the database views will allow users read-only access to the data within the system and enable them to analyse and process this data through the use of other statistical tools.

42. In implementing the database solution for the DMS it was originally intended that the database system would run on the Wintel platform. However, initial rounds of system testing, and performance during the System Testing phases, indicated performance issues with this platform. As a result it was decided to move from the Wintel platform to a Sun Solaris 9 running on Sun Microsystems V890 servers.

43. The DMS will be the critical system within the CSO for survey processing. When rolled out within the organisation it is envisaged that there will be 100+ surveys processed by the DMS at monthly, quarterly and yearly intervals. As a result we have implemented Veritas Clustering software on database layer to facilitate database failover and ensure that any downtime due to server outage is minimised.

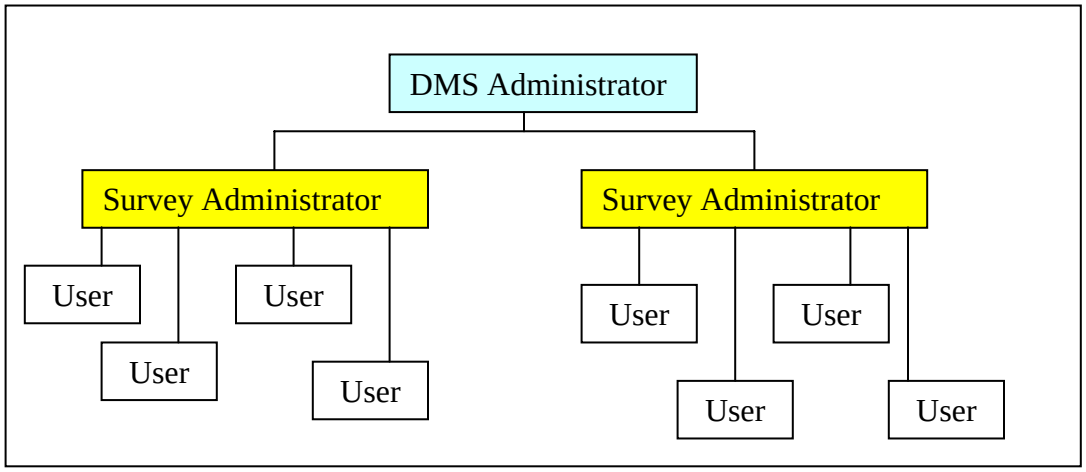
## Other Components

44. A brief description of the applications that interact with the DMS to contribute to the overall solution is included below.

| Application | Description |
|---|---|
| DMS Filestore | The DMS Filestore is a shared network drive onto which both the DMS User and the DMS Weblogic Server have access.<br><br>Users have restricted access to specific Survey and Business Areas folders. All data to be Imported / Exported to the DMS must be stored on the DMS Filestore. |
| SAS | The Seasonal Adjustment processing with the DMS is performed using SAS.<br><br>The Weblogic mid-tier makes a remote call to the SAS server to facilitate the Seasonal Adjustment calculations. |
| CARS<br>[Statistics New Zealand] | The contents of the CARS application are mirrored within the DMS.<br><br>All data that is disseminated in the DMS must utilise a CARS classification, as a consequence the DMS Survey Metadata links with the CARS data for survey definition and processing. |
| CBR<br>[Statistics New Zealand] | The CBR is a hierarchical database of businesses within Ireland.<br><br>Surveys within the DMS can either use the CBR for survey processing. |
| SSA Names3 | The SSA Names3 Server is used for data matching and duplicate identification within the DMS.<br><br>If a user wished to search a Register the SSA Names3 server will allow them specify the ‘accuracy’ of the search which they require and display the results accordingly.<br><br>When users are entering Register Data into the DMS the data is passed through the SSA Names3 Server facilitating the identification of potential duplicates in the data stream. |

## 5. ORGANISATIONAL AND CULTURAL ISSUES

45. The introduction of the DMS will create 2 new roles within the CSO
    - DMS Administrator: DMS Support Staff who have the highest level access to the DMS System and will manage the day to day interaction with the system.
    - Survey Administrator: CSO Statisticians who will process their surveys through the DMS.




46. The Security Application within the DMS allows for end-user access to a survey processed in the DMS to be closely defined.

47. The process works as follows:
    1. Survey is defined within the DMS by the DMS Administrator.
    2. The DMS Administrator assigns a Survey Administrator to the survey.
    3. The Survey Administrator then assigns the staff in their area access to the DMS application based on their specific requirements.

48. The security definitions within the system are at a granularity to allow access to specific functionality within the applications that make up the DMS, e.g. Create a register in the Register Management Application.

49. The Survey Administrator defines the metadata for the survey and has control over how the survey will be defined and executed within the DMS. All metadata defined being stored in the Core DMS / Process Metadata tables on the Sybase ASE database.

50. The DMS was developed by Cognizant Technology Solutions (CTS) in partnership with the CSO. The maintenance contract for the DMS has been given to CTS for the first year post rolling out the DMS, with the provision for the CSO to continue with CTS support for another 5 years.

51. The introduction of the DMS within the CSO has resulted in significant change within the I.T. Department. As a result there has been a need to establish a Java Development Team and a Weblogic team, with the intention of the CSO teams taking over the maintenance of the DMS post the completion of first year maintenance agreement with CTS. There has been knowledge transfer processes put in place to facilitate the CSO team in taking over the maintenance of the DMS going forward.

52. End-user training has been facilitated through the development of E-Learning Applications, DMS Application Manuals. Power Users / Survey Administrator have also had a week long training course covering all functions within the DMS. Prior to surveys going live on the system they will also undergo a refresher course in the DMS.

## Appendix A: Extract from DMS Common Codes Table

| CommonCode_PK | SectionType | FieldCodeNbr | FieldCodeInd | FieldValueText | FieldCodeDesc | FieldCodeOrder |
|---|---|---|---|---|---|---|
| 1 | Common.AggregationMethod | 1 | [NULL] | Total | [NULL] | 1 |
| 2 | Common.Application | 1 | [NULL] | Register Management | [NULL] | 1 |
| 3 | Common.Application | 2 | [NULL] | Sample Selection | [NULL] | 2 |
| 4 | Common.Application | 3 | [NULL] | Survey Management | [NULL] | 3 |
| 5 | Common.Application | 4 | [NULL] | Respondent Management | [NULL] | 4 |
| 6 | Common.Application | 5 | [NULL] | Data Capture | [NULL] | 5 |
| 7 | Common.Application | 6 | [NULL] | Imputation | [NULL] | 6 |
| 8 | Common.Application | 7 | [NULL] | Aggregation | [NULL] | 7 |
| 9 | Common.Application | 8 | [NULL] | Seasonal Adjustment | [NULL] | 8 |
| 10 | Common.Application | 9 | [NULL] | Dissemination | [NULL] | 9 |
| 11 | Common.CFPricesInd | 1 | [NULL] | Current | [NULL] | 1 |
| 12 | Common.CFPricesInd | 2 | [NULL] | Fixed | [NULL] | 2 |
| 13 | Common.CodingType | 1 | [NULL] | Precision | [NULL] | 1 |
| 14 | Common.CodingType | 2 | [NULL] | Death Coding | [NULL] | 2 |
| 15 | Common.ComparisonOperators | 1 | [NULL] | = | [NULL] | 1 |
| 16 | Common.ComparisonOperators | 2 | [NULL] | != | [NULL] | 2 |
| 17 | Common.ComparisonOperators | 3 | [NULL] | < | [NULL] | 3 |
| 18 | Common.ComparisonOperators | 4 | [NULL] | <= | [NULL] | 4 |
| 19 | Common.ComparisonOperators | 5 | [NULL] | > | [NULL] | 5 |
| 20 | Common.ComparisonOperators | 6 | [NULL] | >= | [NULL] | 6 |
| 21 | Common.ComparisonOperators | 7 | [NULL] | Is Between | [NULL] | 7 |
| 22 | Common.ComparisonOperators | 8 | [NULL] | In | [NULL] | 8 |
| 23 | Common.CriteriaOperators | 1 | [NULL] | SELECT | [NULL] | 1 |
| 24 | Common.CriteriaOperators | 2 | [NULL] | AND | [NULL] | 2 |
| 25 | Common.CriteriaOperators | 3 | [NULL] | OR | [NULL] | 3 |
| 26 | Common.CriteriaOperators | 4 | [NULL] | END | [NULL] | 4 |
| 27 | Common.DataType | 1 | [NULL] | Int | [NULL] | 1 |
| 28 | Common.DataType | 2 | [NULL] | Varchar | [NULL] | 2 |
| 31 | Common.DataType | 5 | [NULL] | Char | [NULL] | 5 |
| 36 | Common.DataType | 10 | [NULL] | TinyInt | [NULL] | 10 |
| 37 | Common.DataType | 11 | [NULL] | Numeric | [NULL] | 11 |
| 38 | Common.DataType | 12 | [NULL] | Datetime | [NULL] | 12 |
| 39 | Common.DataType | 13 | [NULL] | SmallInt | [NULL] | 13 |
| 40 | Common.DataType | 14 | [NULL] | Bit | [NULL] | 14 |
| 41 | Common.Delimiter | 1 | [NULL] | Tilda | [NULL] | 1 |
| 42 | Common.Delimiter | 2 | [NULL] | Comma | [NULL] | 2 |
| 43 | Common.Delimiter | 3 | [NULL] | Tab | [NULL] | 3 |
| 44 | Common.Delimiter | 4 | [NULL] | Semi-Colon | [NULL] | 4 |
| 45 | Common.Delimiter | 5 | [NULL] | Pipe | [NULL] | 5 |
| 46 | Common.EditRuleType | 1 | [NULL] | Range | [NULL] | 1 |
| 47 | Common.EditRuleType | 2 | [NULL] | Comparison | [NULL] | 2 |
| 48 | Common.EditRuleType | 3 | [NULL] | Sum | [NULL] | 3 |
| 49 | Common.EditRuleType | 4 | [NULL] | Ratio | [NULL] | 4 |
| 50 | Common.FieldType | 1 | [NULL] | Text Field | [NULL] | 1 |
| 51 | Common.FieldType | 2 | [NULL] | Radio button | [NULL] | 2 |
| 52 | Common.FieldType | 3 | [NULL] | Combo box | [NULL] | 3 |
| 53 | Common.FieldType | 4 | [NULL] | Check box | [NULL] | 4 |
| 60 | Common.FilePurposeInd | 1 | [NULL] | Import | [NULL] | 1 |
| 61 | Common.FilePurposeInd | 2 | [NULL] | Export | [NULL] | 2 |

## Appendix B: Example MetaData Table from the DMS

| Table Name | Creation type | Column Name | DataType | Relationship | Description |
|---|---|---|---|---|---|
| **M_RegisterVariable** | **Static** | | | | This table contains all the Register Variables information for a Register. |
| | | VariableID_PK | int | Primary Key | Variable ID |
| | | RegisterID_FK | int | | Register ID of the Register in which the Variable is present |
| | | VariableNameText | varchar(30) | | Variable Name |
| | | LabelText | varchar(40) | | Variable Label |
| | | FormatID | int | | Format type ID<br><br>0 - Default<br>1 - Herd Number<br>2 - PPS Number<br>3 - MVSerNo |
| | | DataTypeID | smallint | | Data type ID<br><br>1 - Numeric<br>2 - String<br>3 - Date<br>4 - Boolean |
| | | ApplicableInd | bit | | Flag to indicate if the variable is applicable in the Register. |
| | | SearchVarInd | bit | | Flag to indicate if the variable is Search applicable in the Register. |
| | | CommonInd | bit | | Flag to indicate if the Variable is a generic Variable. |
| | | MandatoryInd | bit | | Flag to indicate if the Variable is Mandatory. |
| | | UpdateUser | varchar(20) | | The login name of the user who last updated the row. |
| | | UpdateDate | datetime | | This is the date and time at when the user last updated the row. |
| | | CodedInd | bit | | Flag to indicate if the Variable is coded. |
| | | HoldsCodedValueInd | bit | | Flag to indicate if the Variable holds codes directly.<br><br>0 - Does not Holds the coded Values<br>1 - Holds the coded Values |
| | | Length | int | | Length of the Variable |
| | | Prec | int | | Precision of the Variable |
| | | CapsOnInd | bit | | Flag to indicate if the CAPS ON property is turned on for the Variable. |
| | | | | | |