

EPA STRIVE Programme 2007-2013

Building the Database

(2007-S-CD-1-S1)

ISIS Final Technical Report 16

Prepared for the Environmental Protection Agency
by
Teagasc and Cranfield University

Authors:

Hallett, S.H. and Carvalho, J.

ENVIRONMENTAL PROTECTION AGENCY

An Ghníomhaireacht um Chaomhnú Comhshaoil
PO Box 3000, Johnstown Castle, Co.Wexford, Ireland

Telephone: +353 53 916 0600 Fax: +353 53 916 0699

[Email: info@epa.ie](mailto:info@epa.ie) Website: www.epa.ie

ACKNOWLEDGEMENTS

This report is published as part of the Science, Technology, Research and Innovation for the Environment (STRIVE) Programme 2007-2013. The programme is financed by the Irish Government under the National Development Plan 2007-2013. It is administered on behalf of the Department of the Environment, Heritage and Local Government by the Environmental Protection Agency which has the statutory function of co-ordinating and promoting environmental research.

This project involved many staff and students based at Teagasc Johnstown Castle, Teagasc Athenry, Cranfield University and University College Dublin. Many thanks go to all the staff, PhD students, placement students and FAS placements who contributed to this project. We also wish to thank the Steering Committee, Dr Alice Wemaere (EPA), Ms Niamh Connolly (EPA), Dr Mark Gibson (Teagasc), Prof Peter Loveland, Dr Erika Micheli and Dr Arwyn Jones.

DISCLAIMER

Although every effort has been made to ensure the accuracy of the material contained in this publication, complete accuracy cannot be guaranteed. Neither the Environmental Protection Agency nor the author(s) accept any responsibility whatsoever for loss or damage occasioned or claimed to have been occasioned, in part or in full, as a consequence of any person acting, or refraining from acting, as a result of a matter contained in this publication. All or part of this publication may be reproduced without further permission, provided the source is acknowledged.

The EPA STRIVE Programme addresses the need for research in Ireland to inform policymakers and other stakeholders on a range of questions in relation to environmental protection. These reports are intended as contributions to the necessary debate on the protection of the environment.

EPA STRIVE PROGRAMME 2007-2013

Published by the Environmental Protection Agency, Ireland

Details of Project Partners

Teagasc - Environment, Soils and Land-use Centre,
Johnstown Castle, Wexford, County Wexford, Ireland.
Tel.: +353 (0)53 9171200
E-mail: rachel.creamer@teagasc.ie

Teagasc – Spatial Analysis Group, Rural Economy and Development Programme.
Ashtown, Dublin 15, Ireland.
Tel.: +353 (0)1 8059953
E-mail: reamonn.fealy@teagasc.ie

Cranfield University - National Soil Resources Institute
Cranfield Campus, Cranfield, Bedfordshire, MK43 0AL, UK
Tel.: +44 (0)1234 750111
Email: t.mayr@cranfield.ac.uk

University College Dublin
School of Biosystems Engineering, Agriculture and Food Science,
Belfield, Dublin 4, Ireland.
Tel.: +353 (0)1 7162808
Email: nick.holden@ucd.ie

Executive Summary

The Irish Soil Information System (ISIS) project was established in 2008, following a comprehensive inventory of Irish soil data compiled by Daly and Fealy (2007) which highlighted that soil data coverage of Ireland was incomplete in both detail and extent. The ISIS project is funded under the Environmental Protection Agency STRIVE Research Programme 2007-2013 and co-funded by Teagasc. It was led by Teagasc with the participation of researchers from Cranfield University (UK) and University College Dublin. The overall objective of the ISIS project was to conduct a programme of structured research into the national distribution of soil types and construct a soil map, at 1:250,000 scale, which will identify and describe the soils according to a harmonised national legend. This map is now available in digital format and forms the basis of a new soil information system for Ireland (<http://isis.teagasc.ie>).

The ISIS project has utilised existing data and maps from the previous National Soil Survey (NSS) conducted by An Foras Talúntais (forerunner organisation to Teagasc). The NSS produced: mapping at 1:126,720 scale for 44% of the country; a General Soil Map of Ireland and a National Peatland map, both at 1:575,000 scale and other miscellaneous large scale mapping of experimental farms. In addition, more recent map products have been included such as the Indicative Soil and Subsoil mapping (Fealy and Green, 2009) with national coverage using GIS and remote sensing techniques.

Comparison of soil information at European scale has led to the requirement for the harmonisation and coordination of soil data across Europe, and, in light of the demands for soil protection on a regional basis within member states there is a growing need to support policy with a harmonised soil information system. The European Soil Bureau Network (ESBN) Technical Working Group dealing with Soil Monitoring and Harmonisation recommended a soil map of Europe at a scale of 1:250,000 as an economically feasible intermediate scale that can identify specific problems at regional scale (Montanarella and Jones, 1999).

The ISIS project adopted a combined methodology of utilising novel predicted mapping techniques in tandem with traditional soil survey applications. This unique combination at a national scale has resulted in the development of a new national soil map for Ireland. Building upon the detailed work carried out by the An Foras Talúntais (AFT) survey (known as *Terra Cognita*), the ISIS project generated soil-landscape models at a generalised scale of 1:250,000 for the counties of Carlow, Clare, Kildare, Laois, Leitrim, Limerick, Meath, Offaly, Tipperary South, Waterford, Westmeath, Wexford, West Cork, West Mayo and West Donegal. These soil-landscape models (also referred to as soilscape) were used as the baseline data for statistical models (random forests, Bayesian belief networks and neural networks) to predict soil map units in counties where there was no map available (referred to as *Terra Incognita*). To validate the methodology, this work was supported by a 2.5 year field survey, in which 11,000 locations were evaluated for soil type, using an auger bore survey approach. These data were used to check the predicted soil mapping units (associations) for counties: Cavan, Dublin, East Cork, East Donegal, East Mayo, Galway, Kerry, Kilkenny, Louth, Monaghan, Roscommon, Sligo, Tipperary South and Wicklow, where a detailed soil

survey map was not available. Where new soil information was generated, due to previously unknown combinations of soil-landscape units, profile pits were selected at representative locations across the country. These 225 pits were described and sampled in detail and were used to generate a new soil classification system for the country. The final product is a unique combination of new and traditional methodologies and soils data from both the AFT and the ISIS project. The final, soil association map of Ireland consists of 58 associations (excluding areas of alluvium, peat, urban, rock or marsh) that are made up from 213 soil series. Associated representative profile information is available in the online soil information system.

A key component of the ISIS project has been the development of a soil and land information system and associated public web site. This system has been designed to hold the complete set of information deriving both from the ISIS field programme and modelling activity, as well as the previously existing legacy soils information available for Ireland. Drawing on this information system, the web site is designed to hold and disseminate this information online both in cartographic and tabular form to stakeholders. Prior to this development, there has been no harmonised computerised system in place to hold and manipulate national Irish soils data. The information system therefore addresses the pressing need and requirement for a publicly-accessible, integrated IT framework based upon contemporary informatics standards to serve the many and varied stakeholders having an interest in soils information in Ireland.

This work is part of the Irish Soil Information System (ISIS) Project, managed by Teagasc (the Irish Agriculture and Food Development Authority) and co-funded by the Environmental Protection Agency (EPA) of Ireland through their Science, Technology, Research and Innovation for the Environment (STRIVE) Programme, as part of the National Development Plan 2007-2013.

Contents

OVERVIEW.....	4
SCHEMA DESIGN USING THE ENTERPRISE ARCHITECT CASE TOOL	4
DATA IMPORT TOOL	7
RUNNING THE DATA IMPORT TOOL.....	9
IMPORTING THE SQL	9
CONFIGURING AND RUNNING SCRIPTS ON MS SQLSERVER.....	10
BUILDING THE ESRI GEODATABASE.....	13
UPDATING DATA IN SIS CORE	15
EDITING DATA IN SQL SERVER.....	15
SIS CORE AND SIS PUBLIC	15
MAP EDITING	15
APPENDIX ONE – DIGITAL INCLUSIONS	17
APPENDIX TWO – TRANSCRIBING MS ACCESS DATA TYPES TO MS SQLSERVER .	19

This report is submitted as part of the deliverable by Cranfield University to Teagasc Ireland for the provision of technical support and assistance during the Irish Soil Information System ISIS project.

Citation. This report may be cited as follows:

Hallett, S.H., Carvalho, J. and Fealy, R. (2013) Building the Databases. 19pp. NSRI, Cranfield University, UK, and Teagasc, Ireland.

Report cover, shows the provisional database SIS schema.

Data Schema Construction Tool

Overview

'SIS Core' is the database instance of SQLServer which holds Teagasc-side data emanating from the ISIS project. SIS Core data will not be accessible directly by the public – examples being datasets holding georeferenced tags, or land owners details etc. A subset of the SIS Core data, named 'SIS Public' will comprise those data tables that will be placed in the public domain. The SIS Public database will be placed both in a SQL Server Light edition instance, and in an ESRI file geodatabase both on the public webserver. The web mapping tools will draw on the latter data.

This document describes how the databases were implemented physically, based on the design guidance notes from the earlier Systems Analysis document 'Document:WP5_D1.1'. Building the databases incorporates a number of stages, as described in this report.

- 1.) The design structure was developed using the CASE tool, 'Enterprise Architect', used to develop the data schema for both the 'SIS Core' and 'SIS Public' databases;
- 2.) Data was migrated from the transition databases to the SQLServer database using a custom-developed Java-based construction tool designed to facilitate the migration of both non-spatial and spatially-enabled data from the field recording databases of the ISIS project arising from WP3, as well as legend and related data tables from WP1, and
- 3.) Spatial data was migrated using a constructed ESRI file geodatabase, used to hold the spatial map data arising from WP2.

Schema Design using the Enterprise Architect CASE tool

The design of the schema for the non-spatial and spatially-enabled data in the ISIS databases was undertaken using the 'Enterprise Architect' CASE (Computer Aided Software Engineering) tool from Sparx Systems <http://www.sparxsystems.com/>.

The schema design was developed and laid out in a graphical format (Fig 1) from which is generated the database-specific SQL DDL (data definition language) script to build the database (Fig 4).

All digital files output from Enterprise Architect, including the full graphical schema and the digital files for loading into Enterprise Architect., are presented in Appendix One.

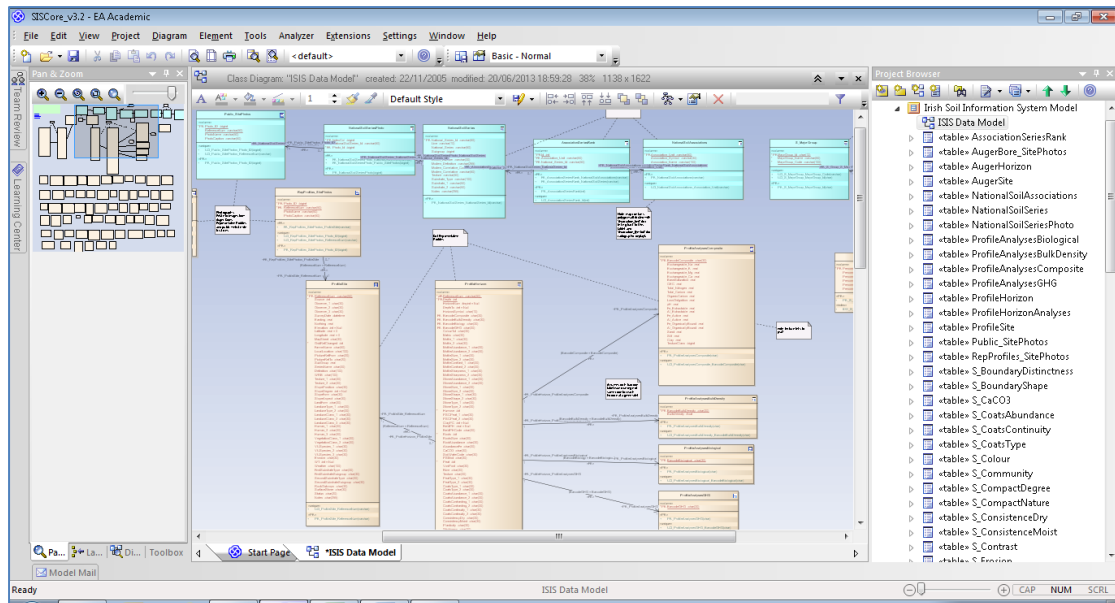


Figure 1 Schema design using the Enterprise Architect CASE tool

Once the design is created graphically, the DDL may be generated for the specific target relational database management system (RDBMS), (Fig 2), for ISIS this is 'MS SQLServer 2008'.

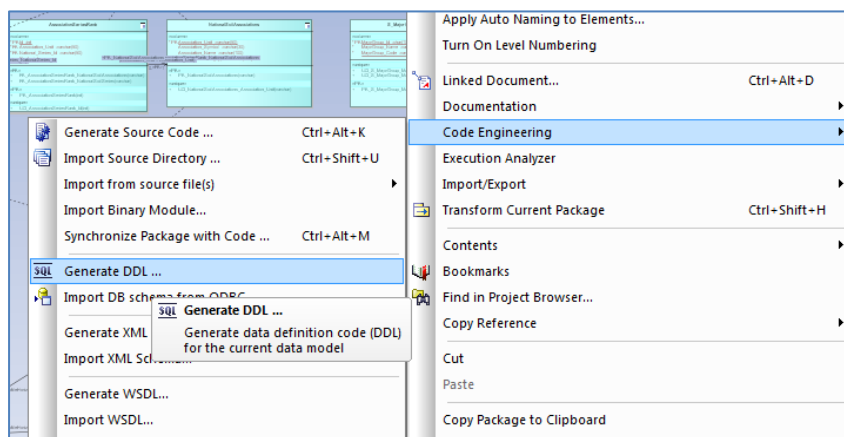


Figure 2 Generating a DDL script

In generating the DDL itself, Enterprise Architect offers a number of options. The final options used are shown in Figure 3. All code was output to one file. Note the use of the `--` symbol for comments, and inclusion of table constraints. The database name was also included in the export DDL. This was varied respectively for the 'SIS Core' and 'SIS Public' exports.

In developing the schema, inevitably there were occasions where the database itself needed to be reset, and all tables and constraints etc. deleted. As this is not a simple task, and is also a task of many stages, a dedicated script was developed to achieve this, 'ISIS_DropAllTables.SQL' (Appendix One). Running this script between test implementations ensured the database was fully cleared in advance.

The resultant DDL SQL script could then be executed in the RDBMS itself to construct the schema in the MS SQL Server administration utility. Initial testing was

undertaken on a 'SQL Server Light Edition' instance at Cranfield ready for the porting of the data structure and data to the ISIS servers in Ireland.

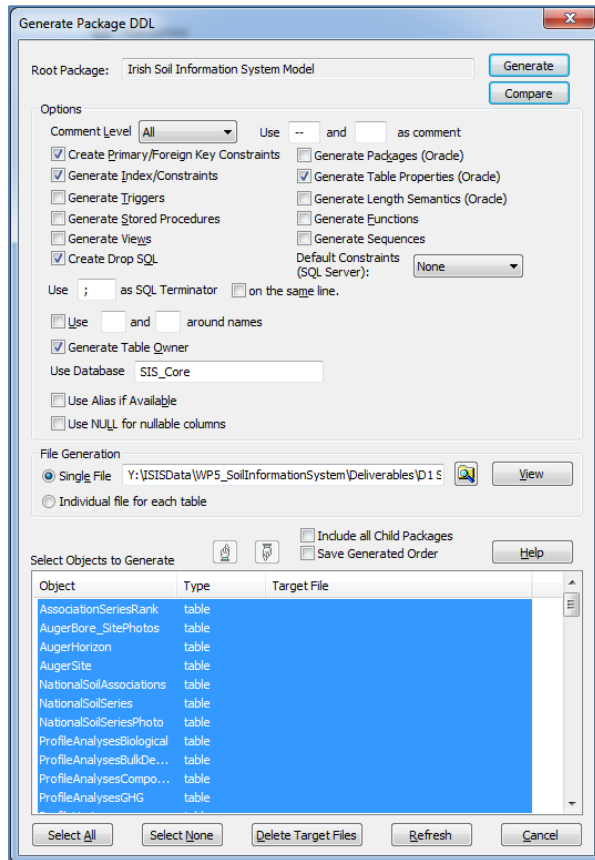


Figure 3 Settings used for exporting schema to DDL

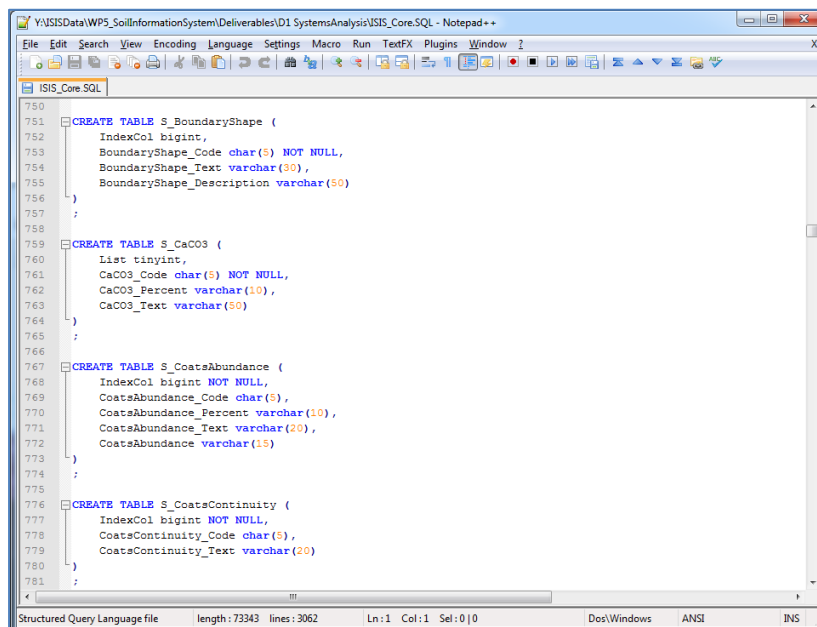


Figure 4 SQL construction DDL script exported from Enterprise Architect

In creating the schema, the tables were designed to match the fields of the originating tables in MS Access. It was noted that the data types are not identical

between Access and SQL Server, so a mapping was developed to aid the transcription of the database design (Appendix Two).

Two schemas were initiated, 'SIS Core' and 'SIS Public', respectively the internal Teagasc database, and the public-facing database used by the web viewer app. On the graphical schema (in file 'ISIS Data Model 3_2.pdf', see Appendix One), the tables are all shown in one diagram, with the public tables coloured in green. To export that public subset, the necessary tables were selected and a right mouse click showed the export DDL menu. The export options dialogue (Figure 3) needed to be adapted for this, e.g. with the new database name. All resultant SQL files are presented in Appendix One.

Once the structural components of the database design was implemented, the next stage was to import the various datasets into the database.

Data Import Tool

Field data from the ISIS project field teams is captured in a series of MS Access databases – for auger bores and for representative profiles, as well as for photographic references etc. Further to this is the national soil legend, the soil association listings and the membership rankings of series in associations. A range of related look-up tables are also required to service these tables. The ultimate residing place for these data is the centralised RDBMS 'SIS Core' and 'SIS Public' databases, described in the sections above.

A utility tool is therefore required to facilitate the transfer of schemas and data from this collection of 'feeder' applications to the core database. The schemas may vary between those adopted in the source standalone applications and the core information system. Therefore a 'mapping' is required to allow transcription of one schema to the other, allowing for changes in data table design. Due to differences in the respective database systems and the aims of the applications, the schemas may be different by design from each other and so necessitate such a mapping. Further to this, other edits can be accommodated on an iterative basis between the systems – such as additional columns and even tables. Thus flexibility is designed into the data migration process.

Furthermore, as the process may require being undertaken more than once, the means to establish a clear audit trail is necessitated. Accordingly a powerful 'data import' tool was developed within the project to allow the mapping of data fields from each respective child database (usually in MS Access) into the core SIS databases ('SIS Core' and 'SIS Public').

The tool was written in the programming language 'Java' using the Integrated Development Environment (IDE) 'NetBeans' (Fig 5). The source code for the tool is also presented in Appendix One.

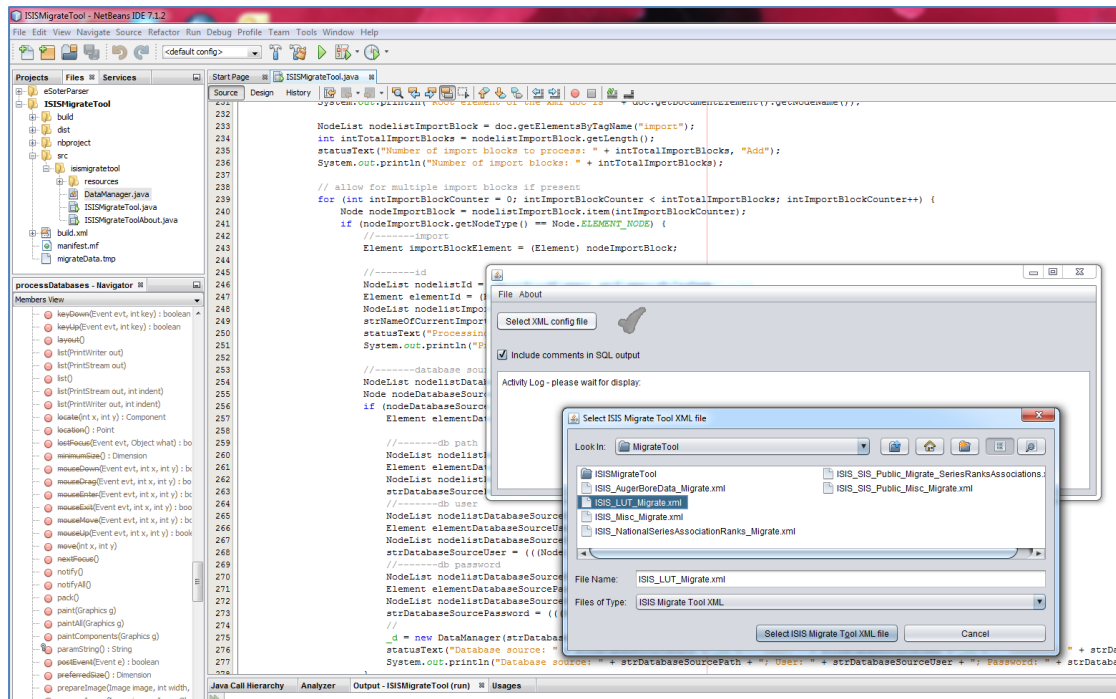


Figure 5 NetBeans Java-based 'Data Import' tool

The operation of the tool is that the data is sourced from the originating MS Access database and written out to the appropriate SQLServer SQL statements under the control of a set of XML configuration files. These output files (e.g. 'ISIS_NationalSeriesAssociationRanks_Migrate.xml') are all presented in digital form in Appendix One. The xml files each follow a prescribed format (Fig 6 shows the XML standard header comment), which allows fine control over the process.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
    ISISMigrateData Data Migration XML Configuration file, v1.0
    =====

    Purpose: Organise transfer of data from various MS Access databases
    to the SIS Core SQL Server database
    Author: Stephen Hallett, NSRI, Cranfield University

    XML File Structure
    =====

    Whole structure must be contained within <SISCore> tag
    Each component for processing is then held in an <import> tag
    The <import> tag holds:

    <id> Descriptor for component (used for labelling only) - free text
    <database_source> Location of the MS Access (mdb or accdb) file
    holding the data to be migrated, contains:
    <path> Full path to database, including database name, in
    format C:\\folder\\folder\\database.mdb
    <user> user name required to access the source database.
    <password> password required to access the source database.
    <database_destination> Name of the output SQLServer database, contains:
    <db> Name of the SQLServer database
    <schema> Name of the SQLServer database schema
    <table> database table to migrate. The table tag has two identifiers:
    source="xxx" name of database table in source database.
    dest="xxx" name of database table in destination database
    (i.e. they tables do not have to have same name).
    <column> database table attribute column to migrate. The column
    tag has several potential identifiers:
    source="xxx" name of field in source table.
    dest="xxx" name of field in destination table (i.e.
    they field do not have to have same name)
  
```

```
autoincrement="true" the column will be an integer
incrementing by 1, starting at 1. Note you still
require a 'dest' field for destination field name.
number="true" the column is numerical (integer or real)
and will not have quotes around it in insert statement
sourceSQL="xxx" MS Access SQL statement returning a
recordset - must include all columns returned.
The SQL is executed in the Access database and must
include all columns for the current table.
```

Example:
=====

```
<SISCore>
  <import>
    <id>National Soil Series</id>
    <database_source>
      <path>C:\\folder1\\folder2\\NationalSoilSeries.mdb</path>
    </database_source>
    <database_destination>
      <db>sis_core</db>
      <schema>dbo</schema>
    </database_destination>
    <table source="NATIONAL_SOIL_SERIES" dest="NATIONAL_SOIL_SERIES">
      <column autoincrement="true" number="true" dest="key"/>
      <column source="Id" number="true" dest="Sample_No"/>
      <column source="National_Series" dest="National_Soil_Series"/>
    </table>
  </import>
</SISCore>

-->
```

Figure 6 Format for the XML configuration file

Running the Data Import Tool

In order to run the data import tool the following steps should be undertaken. First an XML configuration file is required to control the sequence of the operation of the tool (Fig 6). A number of examples are presented in Appendix One. Next the local databases referenced must be visible – for example it may be best to locate these databases in a local folder.

Next the Java programme must be run. The source and compiled code for the programme is presented in Appendix One as file 'ISISMigrateTool.zip'. There are two choices for running the code. The code may either be run directly on the compiled 'jar' (Java Archive) file, thus (Fig 7), or directly via the NetBeans IDE (Fig 5):

```
c:\
cd data_import\dist
java -jar ISISMigrateTool.jar
```

Figure 7 Running the Data Import tool from the command line

Note: assuming you have unpacked the data import tool to c:\data_import

Importing the SQL

Once the process has run, you should have created alongside the XML file a new SQL file called a file named "ISISMigrateToSISCore.sql". This file contains an SQL script able to construct the mirror data tables in the 'SIS Core' or 'SIS Public' implementations (Fig 8). Running the SQL commands in MS SQL Server will create the data structures required.

```

1  /*-----*/
2  * ISISHigrateTool Output SQL file
3  * Version 1.0
4  * Developer: S.Hallett, NSRI, Cranfield University
5  * Purpose: Export SQL file of transactional data files to load into ISIS SIS Core database
6  * Assumption: Data tables are already built (SQL script originates from Enterprise Architect)
7  * File created: Thu Jun 20 18:26:31 BST 2013
8  /*-----*/
9
10 /* Processing: ISIS Miscellaneous files
11 * Database (source): Y:\ISISData\WPS_SoilInformationSystem\Data\Import\MigrateTool\ISIS_Misc_MigrateToSISCore.sql
12 * Table source: S_ISIS_Personnel
13 /*-----*/
14 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
15 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
16 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
17 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
18 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
19 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
20 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
21 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
22 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
23 INSERT INTO [SIS_Core].[isis_core].[S_ISIS_Personnel] ([Personnel_Id], [Personnel_Forename], [Personnel_Surr
24 /*-----*/
25 /* Processing: ISIS Miscellaneous files
26 * Database (source): Y:\ISISData\WPS_SoilInformationSystem\Data\Import\MigrateTool\ISIS_Misc_MigrateToSISCore.sql
27 * Table source: Public_SitePhotos
28 /*-----*/
29 INSERT INTO [SIS_Core].[isis_core].[Public_SitePhotos] ([Photo_ID], [ReferenceNum], [PhotoName], [PhotoCapti
30 /*-----*/
31 /* Processing: ISIS Miscellaneous files
32 * Database (source): Y:\ISISData\WPS_SoilInformationSystem\Data\Import\MigrateTool\ISIS_Misc_MigrateToSISCore.sql
33 /*-----*/

```

Figure 8 SQL output from the Data Import tool

Configuring and Running scripts on MS SQLServer

As described, a series of SQL scripts are generated for execution on the SQLServer instance. Test trials were undertaken to implement these scripts on a prototype SQLServer Light installation at Cranfield. The settings used should be similar to those on the full SIS implementation in Teagasc.

The following database entities were created (Fig 9):

Database: SIS_Core
User: isis_core
Role (Database role): db_owner
Schema: isis_core

Database: SIS_Public
User: isis_public
Role (Database role): db_owner
Schema: isis_public

Figure 9 SQLServer configuration

The first step was to create two separate databases 'SIS Core' and 'SIS Public' respectively (Fig 10).

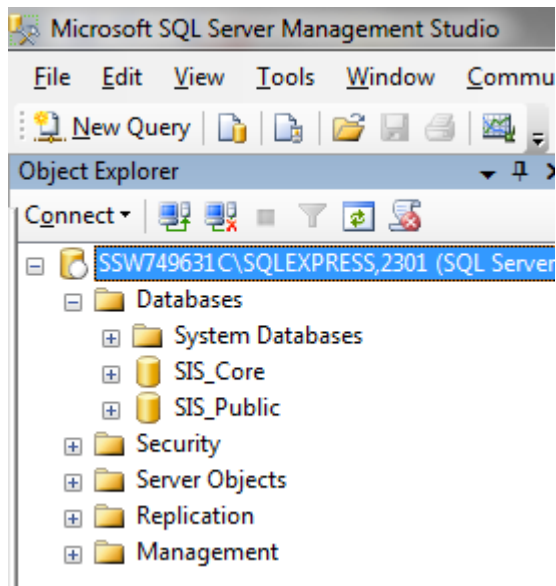


Figure 10 Databases created in SQL Server

Next, schemas and roles were established and assigned to users as noted, thus (Figs 11 and 12).

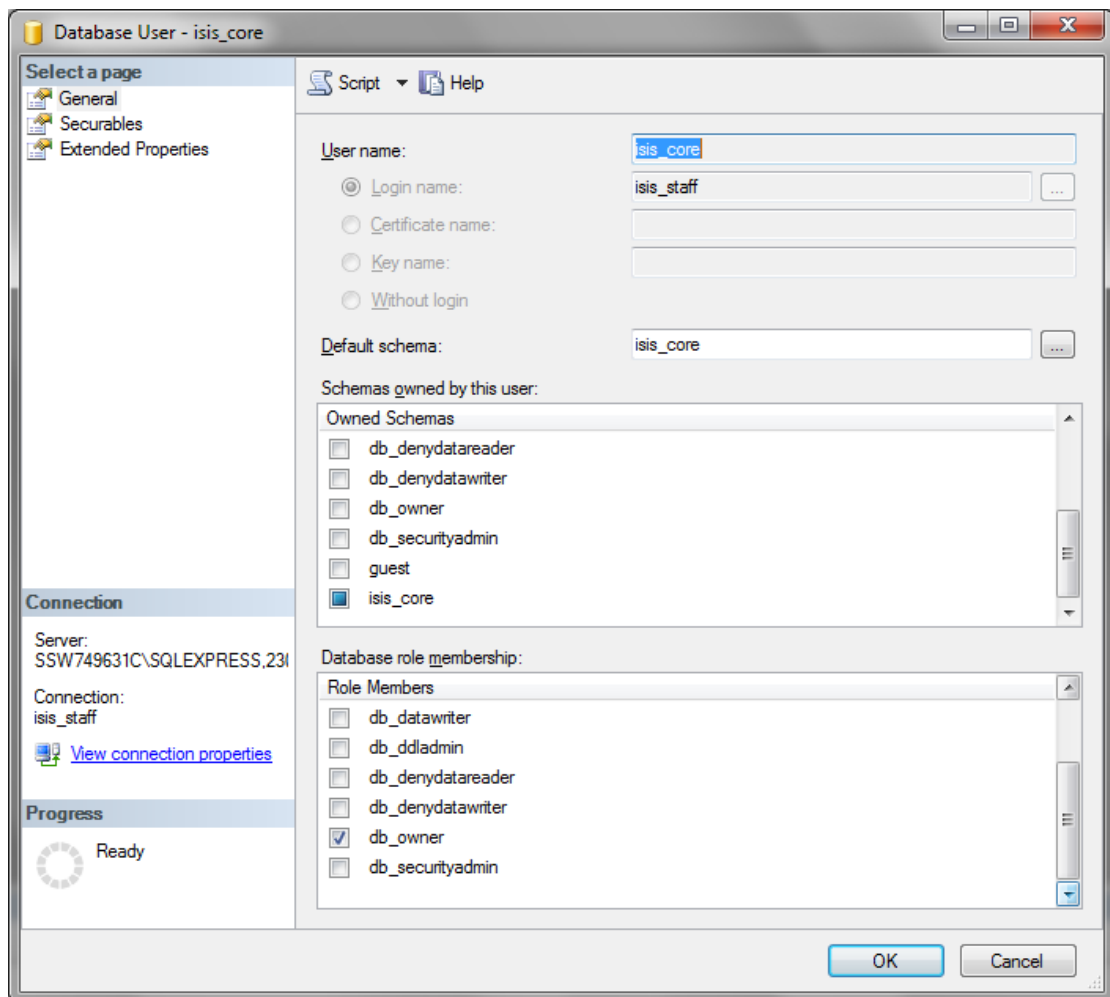


Figure 11 Database user settings for the isis_core user

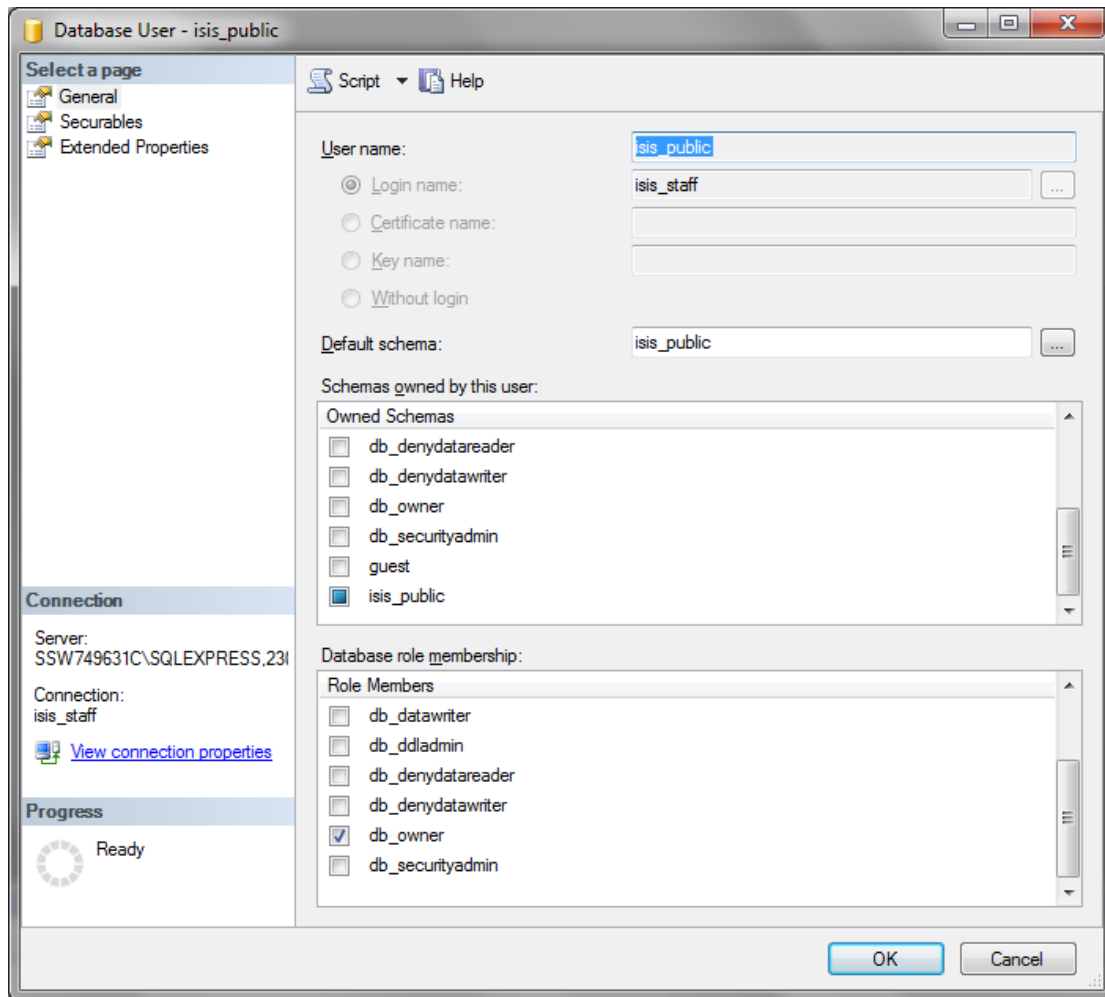


Figure 12 Database user settings for the isis_public user

Once the configurations were established correctly, the SQLServer management Console could be used to execute the respective SQL scripts (Fig 13).

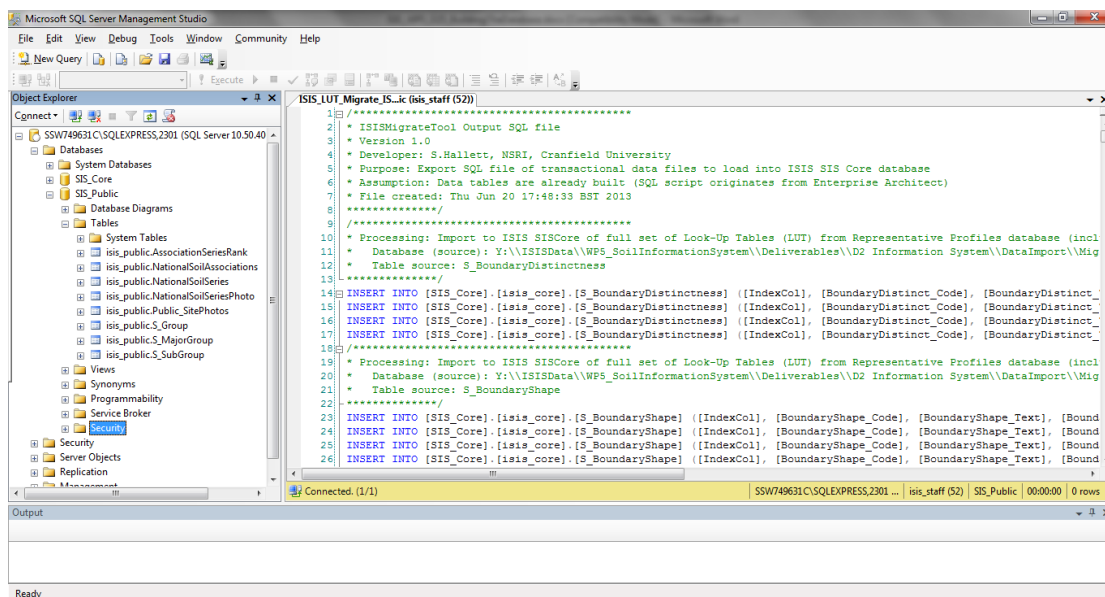


Figure 13 Running an SQL script in the SQLServer Management Console

Building the ESRI Geodatabase

The sections above have covered the construction of the non-spatial and spatially-enabled data tables. The other key element of the wider soil information system is the spatial data component. This is implemented in an ESRI Geodatabase (Fig 14). The geodatabase was delivered as described in document 'ISIS_WP2_D44_NewSoilMapofIrelandMapv1DataDelivery'.

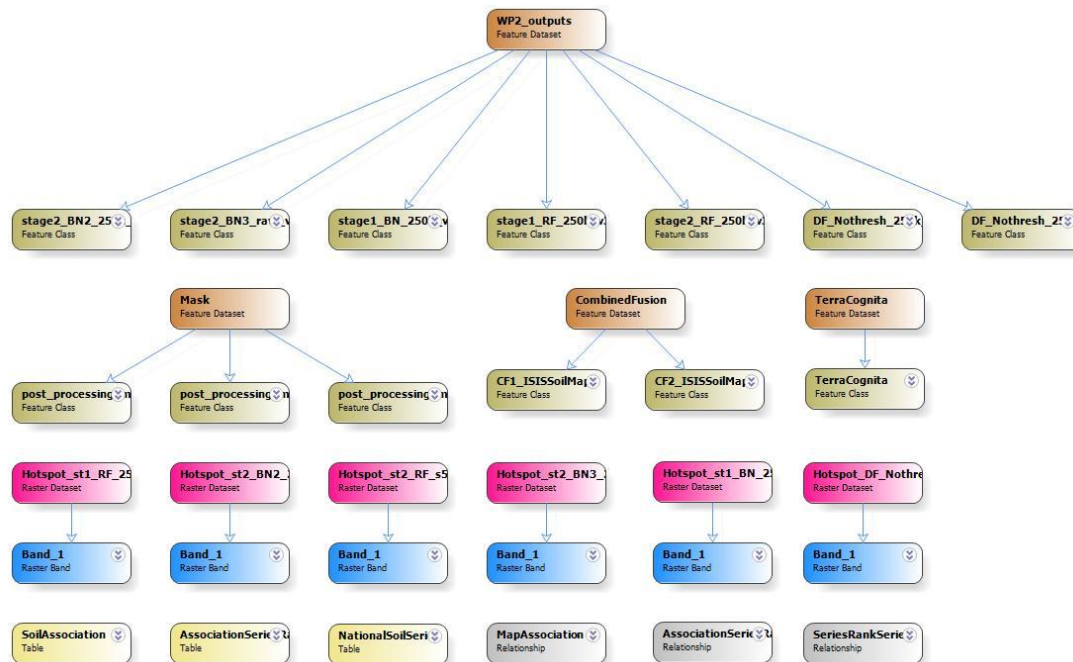


Figure 14 Entities contained in the Geodatabase
(Prepared using ArcGIS Diagrammer)

The geodatabase format is able to hold both spatial and non-spatial data formats in the one database format. As a final decision about the topology of the final database was still in flux, it was decided to replicate the relevant non-spatial tables 'NationalSoilAssociations', 'AssociationSeriesRank' and 'NationalSoilSeries', together with the respective Relationship classes 'Association_SeriesRank' and 'Series_SeriesRank' in the geodatabase (Fig 15).

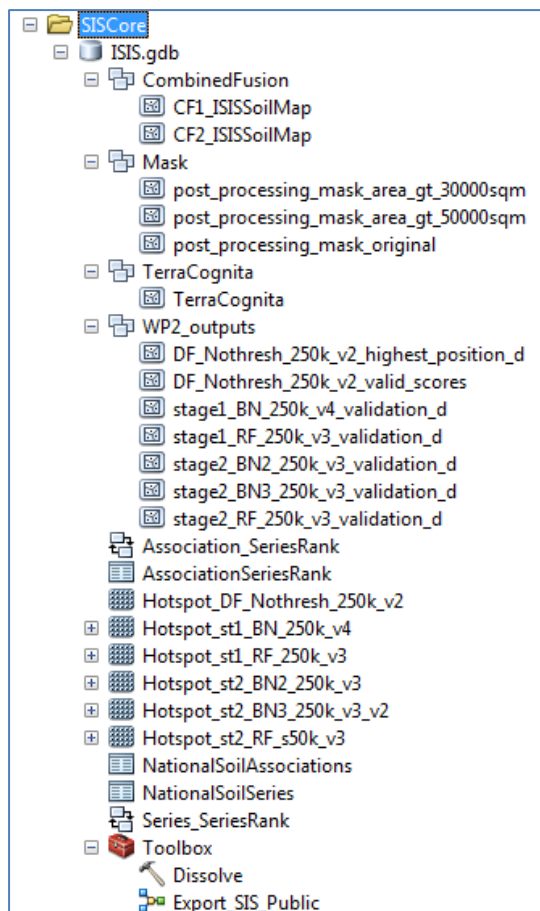


Figure 15 The Geodatabase structure

As a result, the v1 Soil Map has relational integrity and concurrence from the map through to the soil series tables. The geodatabase is also accompanied by an ESRI ArcGIS project 'mxd' file allowing the data to be explored (Fig 16).

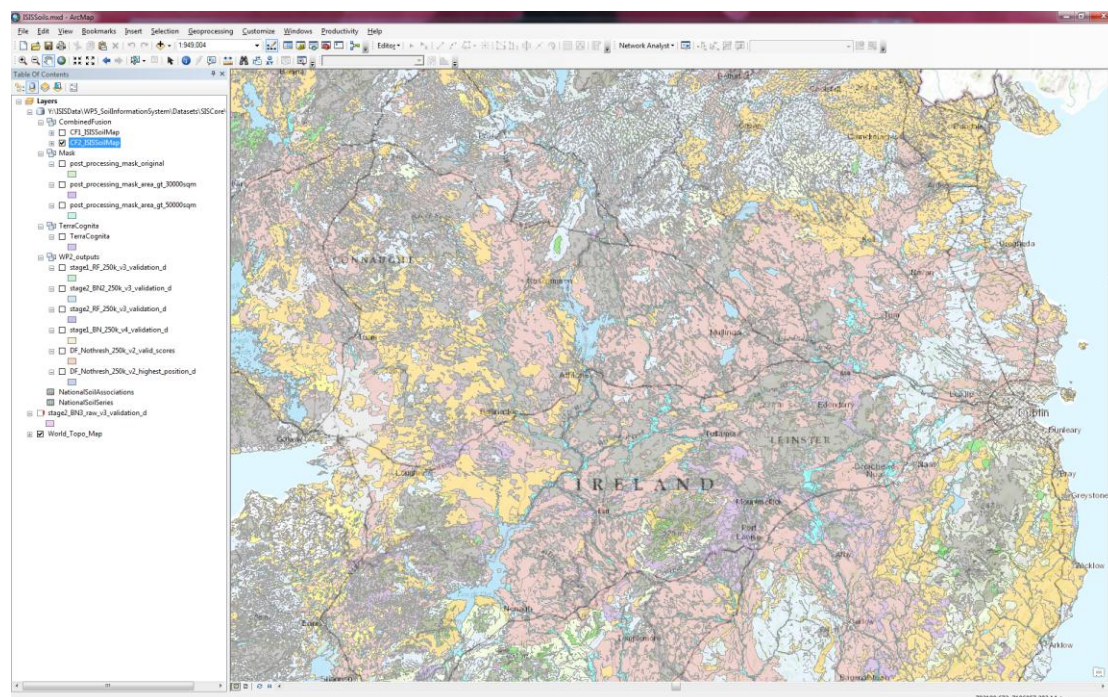


Figure 16 The ArcGIS project file accompanying the geodatabase

Updating data in SIS Core

This report has indicated the means by which both the MS SQL Server data tables and the ESRI geodatabase tables have been constructed and populated. The result is a 'version 1' of the 'SIS Core' and 'SIS Public' databases. It is recognised that the data in the databases is likely to remain in flux, even after the end of the official project. Instructions are therefore required as to how to maintain and manage edits to the datasets.

Editing datasets ...

From this point on, all edits should be made to the SIS Core database, and not to the assortment of spreadsheets and Access databases that have characterised the work to date.

Editing data in SQL Server

Réamonn Fealy has prepared a test demonstrator using the 'ODBC' drivers for MS SQL Server, which allow networked access to the core tables on the SQL Server computer from desktop computers elsewhere in Teagasc.

A networked ODBC join can now be made from a compliant desktop programme such as Access or Excel to the data tables in the SQL Server database. Edits can then be made directly into the database. Every effort has been made in the design to enforce referential integrity within the data schema – meaning that relational joins and cascading relationships are maintained between the related tables.

In addition to editing data in this manner in SQL Server, editing can also be undertaken in an 'editing session' in ArcGIS on the tables in the geodatabase. Each table here uses relationship classes (noted above) to control the referential integrity between tables.

Careful editing ...

As with any tabular editing in a relational database, care should be taken to preserve the referential integrity of the data, and the relationships between data elements between tables (through the use of common primary and foreign keys). In summary, new rows should only be added, or existing rows edited whereby the relational linkages remain intact between the tables. As the data is delivered, there is 'structural integrity' between the tables and concurrence with the map polygons.

SIS Core and SIS Public

All formal editing should be conducted in the SIS Core database. The SIS Public tables are then solely a subset of the SIS Core tables, with no additional editing. The tables required can therefore be literally mirrored across from the SIS Core to the SIS Public database. In Appendix One, and in document 'ISIS_WP5_D1.4', the database schema is shown defining which tables belong to SIS Public.

Map editing

The maps can be edited, post-delivery of the first version of the map, as required either to the geometry or the attributes. However, certain points should be borne in mind. Document 'ISIS_WP2_D44_NewSoilMapofIrelandMapv1DataDelivery' notes how the final 'Combined Fusion Map 2', or 'CF2' map is the product of a series of stages of processing, from the stage one and two sub maps, to the Fusion 'F1' map,

to the CF1 map combining the mask and terra Cognita data and ultimately being dissolved out to the CF2 map. From a GIS perspective the following is true (Table 1):

Table 1 Characteristics of feature classes in geodatabase

Map	Character
F1	Single element
CF1	Multi-part polygon
CF2	Single part polygon

Edits should therefore be made in the light of this sequence. Edits to CF2 alone may have repercussions were edits then required to the earlier versions. Thus ideally edits should be made as 'high up the tree' as possible. The implication of this, however, is the attendant processing is then required to rebuild the intermediate map stages.

A final point to note is that map CF2 is not the 'end of the trail' for the maps. Beyond the scope of this document, a further 'CF3' maps is made which is an exploded 'multi-part polygon' dataset, optimised for the web viewer application. From this map, projected in the Irish National Grid, a fourth and final 'CF3_EPSG3857' map is created reprojected to the 'WGS_1984_Web_Mercator_Auxiliary_Sphere', WKID: 3857 Authority: EPSG. The latter being the required projection for web mapping.

~oOo~

Appendices

Appendix One – Digital Inclusions

Included with this report are a number of digital files, comprising as follows:

File name	Purpose
SISCore_v3.2.eap	'Enterprise Architect' schema file. Note this is document 'ISIS_WP5_D1.4'
ISIS Data Model 3_2.pdf	PDF output of the 'Enterprise Architect' graphical schema. Note this is document 'ISIS_WP5_D1.4'
ISIS_Core.SQL	'Enterprise Architect' generated DDL for the 'SIS Core' database in MS SQLServer
ISIS_Public.SQL	'Enterprise Architect' generated DDL for the 'SIS Public' database in MS SQLServer
ISIS_DropAllTables.SQL	Script to drop (delete) all tables in a given MS SQLServer database
ISISMigrateTool.zip	Data Import tool, written in Java. Unpack into the project file format required by Netbeans.
ISIS_LUT_Migrate.xml	XML control file governing export of lookup tables in 'Java data import' tool
ISIS_LUT_Migrate_ISISMigrateToSISCore.sql	SQL output file created by 'Java data import' tool using the ISIS_LUT_Migrate.xml control file
ISIS_Misc_Migrate.xml	XML control file governing export of miscellaneous tables in 'Java data import' tool
ISIS_Misc_Migrate_ISISMigrateToSISCore.sql	SQL output file created by 'Java data import' tool using the ISIS_Misc_Migrate.xml control file
ISIS_NationalSeriesAssociationRanks_Migrate.xml	XML control file governing export of national soil series, rankings and associations tables in 'Java data import' tool
ISIS_NationalSeriesAssociationRanks_Migrate_ISISMigrateToSISCore.sql	SQL output file created by 'Java data import' tool using the ISIS_NationalSeriesAssociationRanks_Migrate.xml control file
ISIS_SIS_Public_Migrate_SeriesRanksAssociations.xml	XML control file governing export of national soil series, rankings and associations tables in 'Java data import' tool
ISIS_SIS_Public_Migrate_SeriesRanksAssociations_ISISMigrateToSISCore.sql	SQL output file created by 'Java data import' tool using the ISIS_SIS_Public_Migrate_SeriesRanksAssociations.xml control file

ISIS_SIS_Public_Misc_Migrate.xml	XML control file governing export of all SIS Public miscellaneous tables in 'Java data import' tool
ISIS_SIS_Public_Misc_Migrate_ISIS MigrateToSISCore.sql	SQL output file created by 'Java data import' tool using the ISIS_SIS_Public_Misc_Migrate.xml control file

Appendix Two – Transcribing MS Access data types to MS SQLServer

The following guidelines were used in transcribing MS Access data types to MS SQL Server.

Access	SQL Server
Text	nvarchar
Memo	nvarchar(max)
<i>Number:</i>	
Byte	tinyint
Integer	smallint
Long Integer	int
Single	real
Double	float
Replication ID (guid)	Uniqueidentifier
Decimal	float
Date/Time	datetime
Currency	money
AutoNumber	int
Yes/No (Boolean)	bit
OLE Object	varbinary(max)

See <http://blogs.msdn.com/b/ssma/archive/2011/03/06/access-to-sql-server-migration-understanding-data-type-conversions.aspx>