Programme Area: Smart Systems and Heat

Project: EnergyPath

Title: EnergyPath Networks Technical Design & User Guide

Abstract:

This Deliverable comprises a report from The purpose of this document is twofold:

• To provide a consolidated technical description of the EnergyPath Networks (EPN) Tool covering both its design and data architecture

• To provide a user guide for the EPN Tool.

It is designed to be a ‘living’ document accessible to an ‘expert’ user, but not the original developer, which should be updated to reflect further developments to the tool. This document reflects R2.1 of the tool. It should be noted that this document does not describe the functional requirements and specification which have driven the development of the tool, this is described in detail in the separate EnergyPath Networks Tool Functional Specification

Context:

Energy consultancy Baringa Partners were appointed to design and develop a software modelling tool to be used in the planning of cost-effective local energy systems. This software is called EnergyPath and will evolve to include a number of additional packages to inform planning, consumer insights and business metrics. Element Energy, Hitachi and University College London have worked with Baringa to develop the software with input from a range of local authorities, Western Power Distribution and Ramboll. EnergyPath will complement ETI’s national strategic energy system tool ESME which links heat, power, transport and the infrastructure that connects them. EnergyPath is a registered trade mark of the Energy Technologies Institute LLP.

(S3D1) Technical Design and User Guide

CLIENT: Energy Systems Catapult

DATE: 20/02/2017

www.baringa.com

Version History

Version	Date	Description	Prepared by	Approved by
V1_1	31/03/2015	Final	JG / KP / FT	OR
V1_2	02/04/2015	Minor updates	JG	OR
V2_0	20/02/2017	Updated to reflect EPN R2.1	AB / LH / KP / FT	JG

Copyright

Confidentiality and Limitation Statement

[1] EnergyPath is the registered trade mark of Energy Technologies Institute LLP

TABLE OF CONTENTS

1. Introduction

1.1. Background

The purpose of this document is twofold:

- To provide a consolidated technical description of the EnergyPath Networks (EPN) Tool covering both its design and data architecture
- To provide a user guide for the EPN Tool.

It is designed to be a 'living' document accessible to an 'expert' user, but not the original developer, which should be updated to reflect further developments to the tool. This document reflects R2.1 of the tool.

It should be noted that this document does not describe the functional requirements and specification which have driven the development of the tool, this is described in detail in the separate *EnergyPath Networks Tool Functional Specification (Deliverable S1D1)*.

1.2. Acronyms and glossary

The following tables provide a list of common acronyms and glossary of terms used in relation to the tool.

Table 1-1 Acronyms

Acronym	Elaboration
API	Application Programming Interface
DB	Database
FK	Foreign Key
GIS	Geographical Information System
GUI	Graphical User Interface
HOM	Household Options Module
I/O	Input / Output
LA	Local Authority
MCF	Master Control Framework
NAM	Network Analysis Module
OS	Ordnance Survey
POM	Pathway Optimisation Module
PK	Primary Key
RDB	Relational Database
SAM	Spatial Analysis Module
SSHP	Smart Systems and Heat Programme
VBA	Visual Basic for Applications

Table 1-2 Glossary of key terms

Term	Description
Data architecture	Data architecture is composed of models, policies, rules or standards that govern which data is collected, and how it is stored, arranged, integrated, and put to use in data systems and in organizations
Data models	Generally divided into 3 layers • *Conceptual data model* is a summary-level representation most often used on strategic data projects to describe an entire enterprise • *Logical data model* is an attributed data representation (typically but not always containing entity relationships) independent of the database (or other data implementation option) which describes data requirements from the business point of view • *A physical data model* is a fully-attributed data model of implementation that is dependent upon a specific version of a database (or other data implementation option)
Modular architecture	An architecture composed of discrete components such that the design of one component depends only on the interface to other components, not on their internal design. It means that it is ideally possible to replace or add one component without affecting the rest of the system as long as the interface specification is retained.
Module	A discrete component of the tool in the modular architecture. Conceptually, it can refer to both functional module and software module.
Sub-module	A smaller piece of a discrete component of which a module consists. One functional sub-module is functionally independent from others.
Software wrapper	A kind of software entity to moderate the behaviour of a component even though the component contains the different types of implementation (e.g. legacy, black box, etc) so that one can enhance the interoperability among the modules and sub-modules.

1.3. Units

Units for input data within the primary database are described within the database documentation in section 10 or can be viewed via the table description within MS SQL Server Management Studio[2].

In addition, the POM (via AIMMS) has in built unit conversion and it is possible to switch easily between units (e.g. kWh/MWh/GWh) for results reporting.

1.4. Structure of this document

The main section of the document is structured as follows:

- Section 2 describes the overarching tool architecture
- Section 3 describes the database structure
- Section 4 describe the tool process flow and corresponding sub-module libraries
- Section 5 describes the API (application programming interface)

[2] Right click on the table -> Properties -> Extended Properties

- Section 6 provides instructions for installation and configuration
- Section 7 provides an overview of the user interface (Excel and ArcGIS)
- Section 8 describes the specific Excel UI screens

The appendices are structured as follows:

- Appendix 9 describes sample use cases for different parts of the tool
- Appendix 10 contains documentation for each database
- Appendix 11 contains the ER (Entity Relationship) Diagrams for each database
- Appendix 12 contains code documentation for each module

2. Overarching Tool Architecture

2.1. Mapping to the Functional Specification

The original EnergyPath Networks Tool Functional Specification (Deliverable S1D1) outlined four high-level conceptual modules, integrated via an overarching control framework:

- Household Options Module (HOM)
- Spatial Analysis Module (SAM)
- Network Analysis Module (NAM)
- Pathway Optimisation Module (POM)
- Master Control Framework (MCF)

Within each of these a series of discrete sub-modules was outlined (e.g. HOM-003, SAM-005), all controlled via the MCF, to perform the functional requirements of the tool. The technical design of the tool (covering individual module components, individual databases, schema naming, etc) is based on this terminology.

2.2. Software components

The following software components comprise the overall tool; the rationale for selecting each of these, and for the overall tool design architecture is outlined in the earlier (S1D2) Design Architecture deliverable and is not repeated here:

- Python 2.7[3] (64bit) – this has been used to code key functional requirements across the HOM, SAM, NAM and MCF as well as 'wrap' key 3rd party software. The following python libraries are also used:
  - numpy
  - pandas (includes scipy)
  - SQLAlchemy
  - pyodbc
  - pywin32
  - matplotlib
  - sphinx
  - numpydoc
  - scikit-learn
- EnergyPlus[4] v8.1 (64bit) - used within the HOM for dynamic building energy simulation

[3] https://www.python.org/
[4] http://apps1.eere.energy.gov/buildings/energyplus/

- ETI Optimising Thermal Efficiency SAP model (modified version) – used as part of the simplification of the building archetype representation
- ArcGIS[5] 10.2.2 (64bit) – used within the SAM to support the spatial analysis requirements and spatial elements of the GUI
  - This include the Network Analyst and Spatial Analyst extensions along with the Productivity Suite
- PSS Sincal[6] 10.5 (32bit) – used within the NAM for steady state analysis of electricity and heat distribution networks
  - This includes the Electro and Heating modules
- AIMMS[7] 4.23 (64bit) – used within the POM for generating the optimisation problem
  - This includes the CPLEX 12.6.3 solver
- @Risk[8] 6.2 – used via the Excel GUI for generation the Monte Carlo simulated inputs
- Microsoft Excel[9] 2010 (32bit) or above which is used as the main GUI (along with ArcGIS)
  - It is also used to manage @Risk (via VBA) as this is an Excel plug-in)
- Microsoft SQL Server[10] Commercial 2008 R2 (64bit) and management studio
  - For all databases with the exception of ArcGIS which uses its own geodatabase format to store spatial information (relevant attribute data required from this is still stored within the MSSQL databases)

2.3. Design architecture

An overview of the technical design architecture is shown in the figure below. This highlights the four main sub-modules, the MCF and main UI, location of third party tools and primary / secondary databases. It also illustrates the high level control and data flow interactions between them.

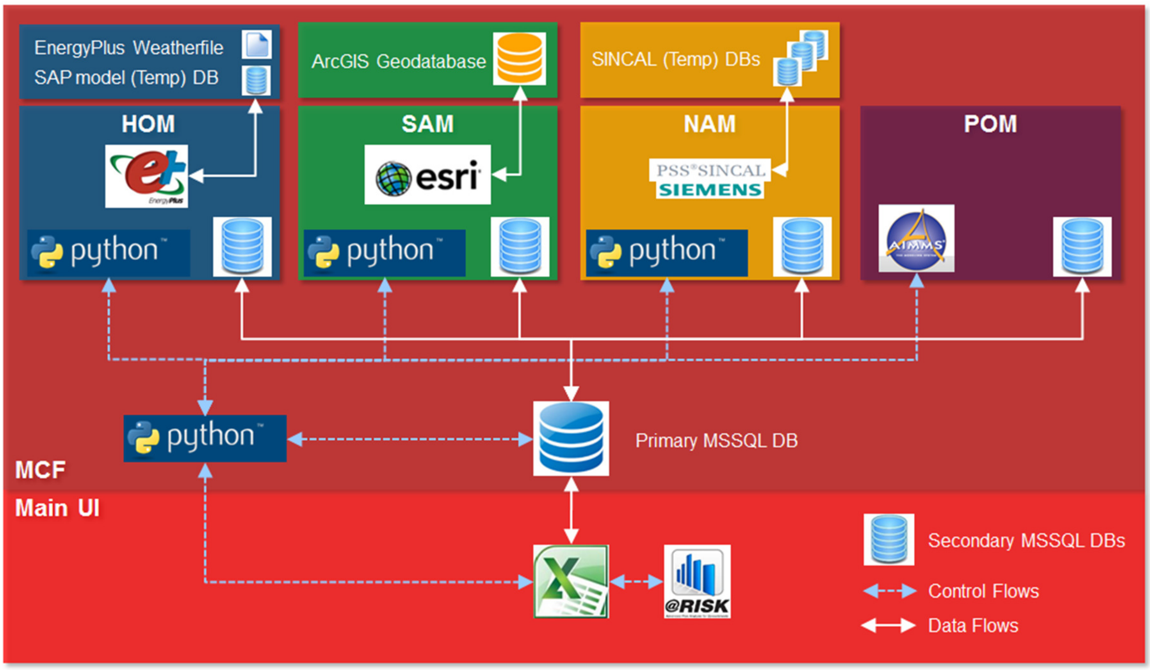
An Excel UI is used to enter and interrogate data and results as well as indirectly execute the process steps within the tool. The ArcGIS UI is also used to support the input of spatial data and visualization of interim and final spatial results.

---

[5] http://www.esri.com/software/arcgis/arcgis-for-desktop

[6] http://w3.siemens.com/smartgrid/global/en/products-systems-solutions/software-solutions/planning-data-management-software/planning-simulation/pages/pss-sincal.aspx
[7] http://www.aimms.com/
[8] http://www.palisade.com/risk/
[9] http://products.office.com/en-us/excel
[10] http://www.microsoft.com/en-gb/server-cloud/products/sql-server/

Figure 2-1 Overview of technical design architecture



The architecture reflects a highly modular structure with a hybrid centralised / decentralised DB configuration, supported by the MCF orchestrator. As outlined in the original *(S1D2) Design Architecture deliverable* this helps to:

- Enable flexible, parallel development across the tool
- Isolate 3rd-party components and the main UI so that they are easier to update or replace in future, if needed
- Centralise the management of primary input and results data to facilitate an audit trail
  - The secondary databases only store intermediate calculation data which is needed across other (sub)-modules
- A main UI which is decoupled from the core tool (with the exception of @Risk), which means that it is easier to move to a separate UI platform at a later date, if required

2.4. IT infrastructure

2.4.1. Operating system

A 64bit Windows operating system architecture is required for compatibility with all 3rd party software within the tool and due to memory requirements within particular sub-modules. 64bit operating systems overcome the memory limit of 4GB seen in 32bit operating systems. A single POM optimization problem typically requires >>4GB of memory.

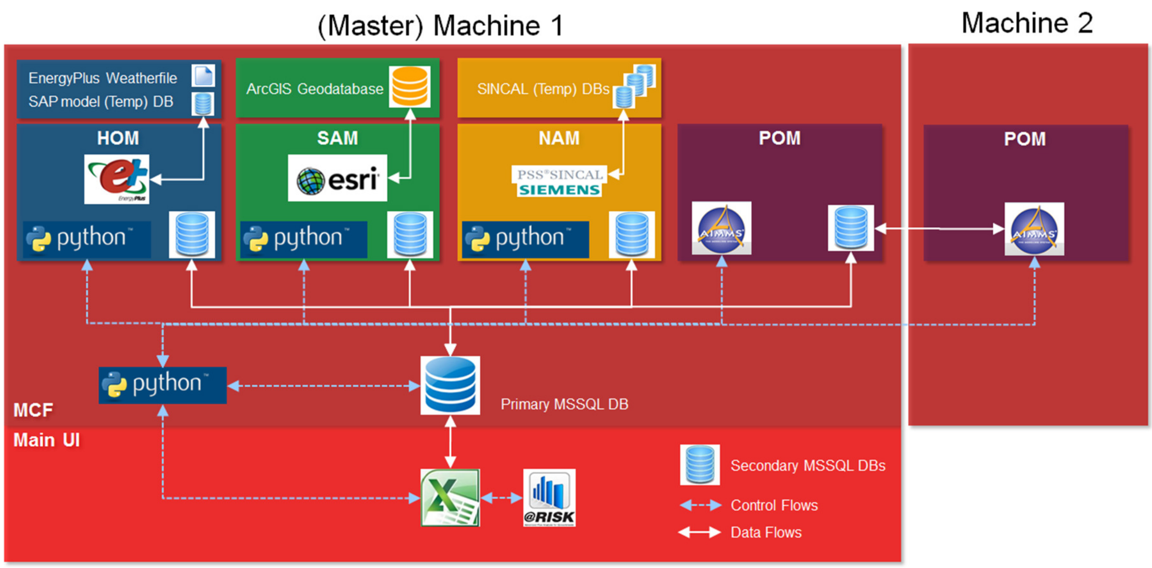
2.4.2. Hardware

The tool has been designed to work on a dedicated, sufficiently powerful desktop computer. With some adaptation (and subject to licensing requirements for the third party software) it would be possible to adapt this to a cloud-based environment.

2.4.3. Parallel computing

From EPN R1.0 onwards the tool has been designed to enable parallel computing of simulations within the POM module – across multiple machines. In a simple two machine example this is enabled by a master control machine (1) controlling the execution of the POM on both machine (1) as normally and temporarily calling machine (2). The POM secondary database is stored on machine (1) only, but the POM AIMMS model on both machine (1) and (2) connect directly to it. This is illustrated in the figure below. It requires that both machines are able to connect to the same SQL Server Instance to access the POM database.

Figure 2-2 Overview of architecture for POM parallel computing



Hence machine (2) only acts as a temporary processor for the optimization and returns the completed results directly to the master machine (1). In this manner if 100 simulations are required 1-50 can be sent to run within machine (1) as normal and 51-100 can be sent in parallel to machine (2). Further machines (subject to available hardware and software licenses for the POM) can be added relatively easily within this configuration.

2.5. Technical configuration of 3rd Party Tools

Installation and initial configuration of the EnergyPath Networks Tool (including 3rd party software) is described in section 6. Global configuration parameter values and settings for various parts of the tool (including commercial third party packages) are stored and described in the following *primary* database tables, as it is necessary to store these as part of an audit trail to be able to recreate a set of results:

- Mcf.pi_GlobalParameters

- Mcf.si_POMGlobalParameters

3. Database Structure

The primary MSSQL database, subject to two key exceptions[11], stores all

- Primary input data necessary to run the tool
- Intermediate output calculations from one module which are necessary for another, the key example being all outputs generated by the HOM / SAM / NAM necessary to run the pathway optimisation process in the POM
- Pathway results from the POM

As part of the modular design, the secondary databases are effectively transient, containing only the data necessary to generate the intermediate output calculations which are passed to other modules.

However, to create these intermediate outputs each of the HOM / SAM / NAM modules (in reality containing a series of sub-modules) and corresponding secondary databases will require input information, which is pushed by the MCF from the primary to the secondary database.

Each secondary database may also store transitional calculations for use *between* sub-modules in the same overarching module. These calculations only serve to support the calculation of the final intermediate outputs from a module and hence are not pulled back from the secondary database to the primary database in the same manner as intermediate output calculations.

The modularity of the tool is reflected within the physical data model via a combination of specific database *schema* (i.e. to compartmentalise related sets of data tables, or views/stored procedures) and a clear naming convention. This is important to navigate the large number of tables required across the entire tool.

A summary of the database structure and an illustrative example is provided below. The full set of tables for each primary and secondary database (including descriptions) and their Entity Relationship diagrams (from the primary and foreign key structure) are described in more detail in sections 10 and 11, respectively.

[11] The EnergyPlus Weatherfile (EPW) and additional spatial data necessary for displaying ArcGIS layers, which is stored in a native ArcGIS geodatabase format.

Table 3-1 Database naming convention and structure

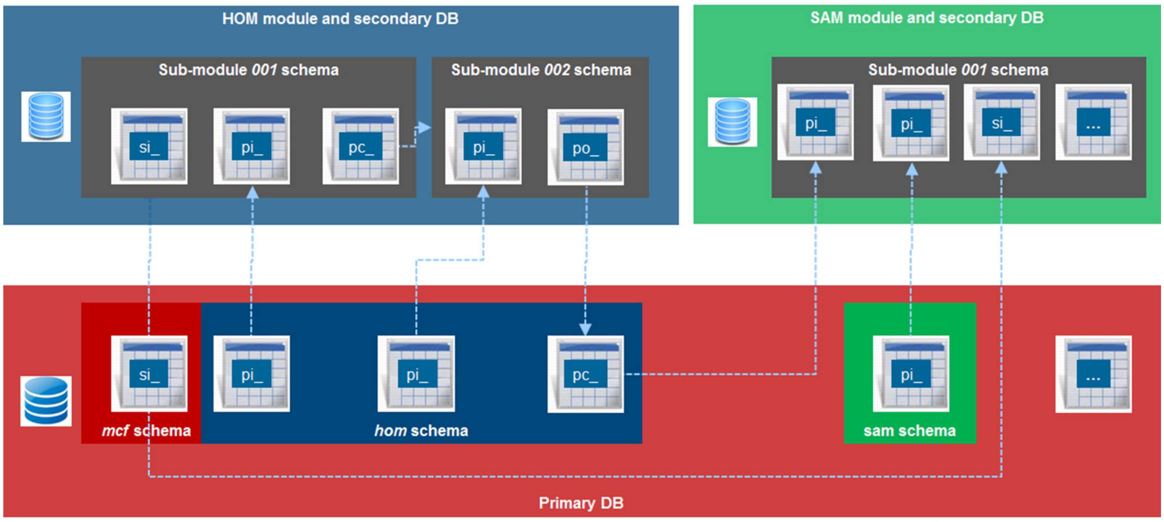
Database	Schema	Description		Table / view / Stored procedure prefix	Description
Primary	*mcf*	Relevant for >1 module		*si_*	Table containing a set of user defined input elements e.g. all time periods contained within the tool. These are used to maintain data consistency. These are pushed into 'mirror tables' within the relevant secondary database (based on the schema). E.g. a hom.si_XXXX table contains set information only relevant to the HOM module.
	hom	Relevant to HOM secondary DB only		*sg_*	Table containing a set of elements generated by a process within a tool module (e.g. the set of final network technologies generated within the NAM to be used in the POM). These are pulled back from a secondary DB into the primary DB These are used to maintain data consistency.
	sam	Relevant to SAM secondary DB only		*pi_*	Parameter table containing user defined input data. E.g. the capex costs of individual network components. These are pushed into 'mirror tables' within the relevant secondary database (based on the schema). E.g. a hom.pi_XXXX table contains parameter data only relevant to the HOM module.
	nam	Relevant to NAM secondary DB only		*pc_*	Parameter table containing data calculated by one module and pulled back to the primary database for use in another. E.g. the cost and losses for the final network technology options from the NAM for use in the POM. These are pulled back from the secondary DB (from a mirrored **po_** table) to the primary DB
	pom	Relevant to POM secondary DB only		*r_*	These contain results generated by the POM which are pulled back to the primary DB
	prepom	Used as part of pre-processing calculations for POM secondary DB		*vs_ (resp. sg_, vp_, pc_)*	These are views used to create sets of elements (resp. intermediate calculated sets, final calculated parameters or intermediate calculated parameters), primarily as part of the pre-processing to generate data for the POM

	monte_carlo	Used as part of calculating simulated input parameters used across tool		*Vp_*	These are views used to create data parameters, primarily as part of the POM pre-processing
	-	-		*sp_*	This is a stored procedure used to create data parameters as part of the POM pre-processing
	Prehom / presam / prenam	Intermediate pre-processing to create input data for HOM / SAM / NAM modules		*pc_ (resp vp_, sg_, vr_)*	Table containing an intermediate calculated parameters e.g. domestic buildings' demands (resp. final calculated parameters, calculated set of elements, calculated parameters for results review) contained within the tool. These are pushed from the primary DB into the equivalent secondary DB tables for use in the relevant sub-modules
				vp_ / vr	Intermediate views used to support the processing of results into the required output format for the UI or to be processed for display in ArcGIS
Secondary	*000*	Tables relevant to >1 sub-module within a single secondary database		*sg_*	Table containing a set of elements generated by a process within the tool module (e.g. the set of final network technologies generated within the NAM to be used in the POM). These are pulled back from the secondary DB into a mirrored table in the primary DB
	001 to 00X	Tables related to a specific sub-module within the secondary database		*pi_*	Table containing a set of user defined data parameters contained within the tool. These are pushed from the primary DB into the equivalent secondary DB tables for use in the relevant sub-modules
	-	-		*po_*	Parameter table containing data calculated by one module and pulled back to the primary database for use in another. These are pulled back from the secondary DB to (a mirrored **pc_** table) the primary DB
	-	-		*pc_*	These are parameter tables containing data calculated by one **sub**-module, for use by another sub-module in the same overall module. Ie they only exist in the secondary DB and are **not** pulled back into the primary DB
	-	-		*p_*	The equivalent of data parameter for use in the POM, generated as part of the pom-preprocessing
Secondary – POM only	*Inputs*	All input data necessary to perform the pathway optimisation		*s_*	The equivalent of a set for use in the POM, generated as part of the pom-preprocessing

	Results	All results generated *and* saved from the pathway optimisation process		*va_*	A view used to create supplementary input data necessary for use in the POM after the pre-processing has transferred data from the primary DB into the POM secondary DB
	-	-		*v_ / p_*	A user defined results variable / parameter in the POM (AIMMS model) which is exported and saved in the POM secondary database. It is then pulled back to the primary DB via the views below
	-	-		*vr_*	Intermediate view to process the results of the optimiser in the POM (AIMMS model) which are exported and saved into the primary DB in an **r_** table

An illustration of the data transfer between tables and schema structure / naming convention is provided in the table below.

Figure 3-1 Illustration of database naming convention



.

3.1. Management of multiple scenarios in data structure

From EPN R2.0 onwards the underlying data structure of the tool has been extended to capture input and processed data associated with multiple scenarios/sensitivities in a number of key areas across the HOM/NAM/SAM. These have been targeted initially to a small number of areas which can take substantial time to configure or process, thus allowing the user to undertake these steps across multiple scenarios in a batch and switch between the resulting data for use in the POM. The data structure has also been configured to allow storage of multiple sets of results from the POM in the primary database.

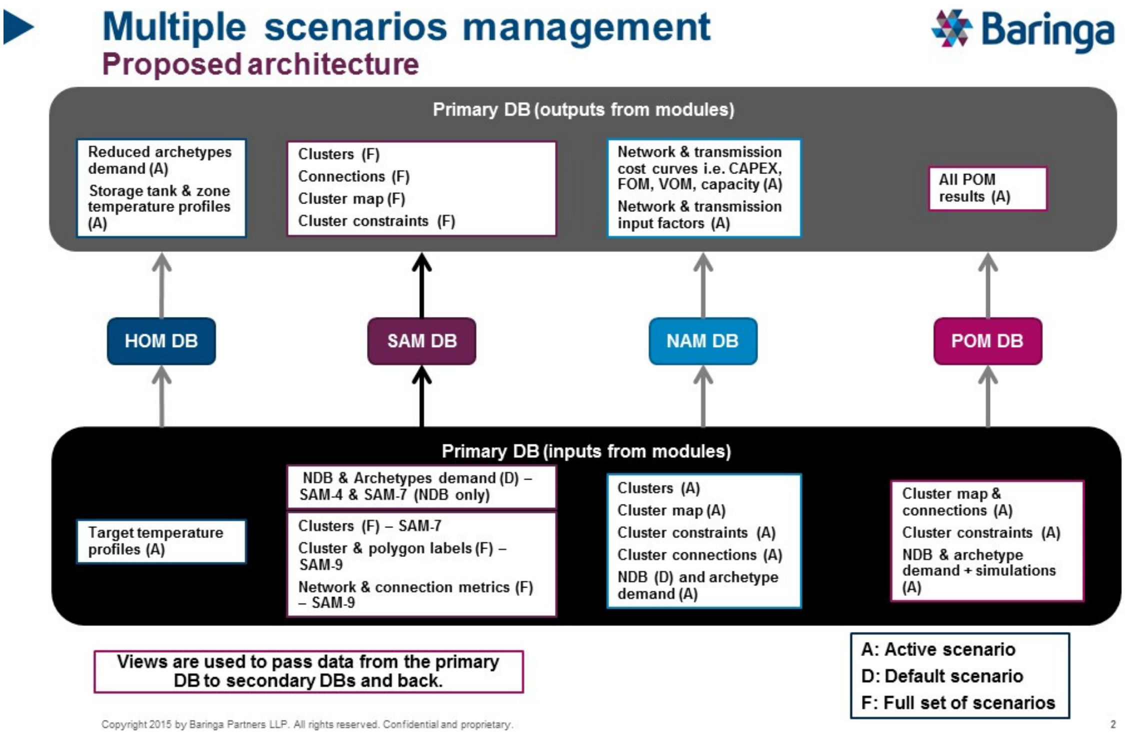
Implementation of multiple scenarios has been included for:

- Target temperature profiles as an input to EnergyPlus (HOM-003), which in turn impacts archetypes heat demand used in SAM, NAM and POM
- Cluster and inter cluster connection definitions (SAM-007/8), which will then be used in the NAM (network and transmission options and cost curves) and POM
- Non-domestic buildings demand (NAM-001, SAM-007, POM-001)Pathway Archetype cluster constraints (SAM-007, POM-001)

A distinction is made between databases - and corresponding sub-modules - which contain a full set of scenario-related data (i.e. tracking all scenarios simultaneously) and those, which are effectively agnostic (i.e. they are passed a sequential set of scenario data as a batch process, but have no knowledge of which scenario is active, as this is controlled remotely by the MCF):

- The MCF / primary database and the SAM contain information across the active set of scenarios in parallel[12] (this is necessary in the SAM to be able to compare multiple cluster definitions in ArcGIS)
- The HOM, NAM, POM modules and secondary databases are agnostic and contain only the scenario data passed by the MCF

Figure 3-2 Illustration of multi-scenario data management



12 An exception is made for data items for which this is not appropriate and which therefore use data associated with 'default scenario'.

4. Tool process flow and sub-module libraries

Each module (HOM, SAM, NAM, POM) contains one or more sub-modules (HOM-001, SAM-005, etc), which reflect discrete, coded libraries. This naming convention links back to the original Functional Specification and, as described above, is reflected in the database structure and naming convention.

In addition to this there are three further parts of the tool, which can effectively be considered as self-contained 'libraries':

- The MCF to control the process execution flow across the entire tool
- The Excel UI to facilitate the read/write of input/results from and to the primary DB
- The Excel UI to facilitate the generation of the Monte Carlo simulated inputs

The interaction between sub-modules is outlined in the diagram and a summary of each is provided in the table below. More detailed documentation for each library is provided in section 12.

Figure 4-1 Overview of functional sub-modules

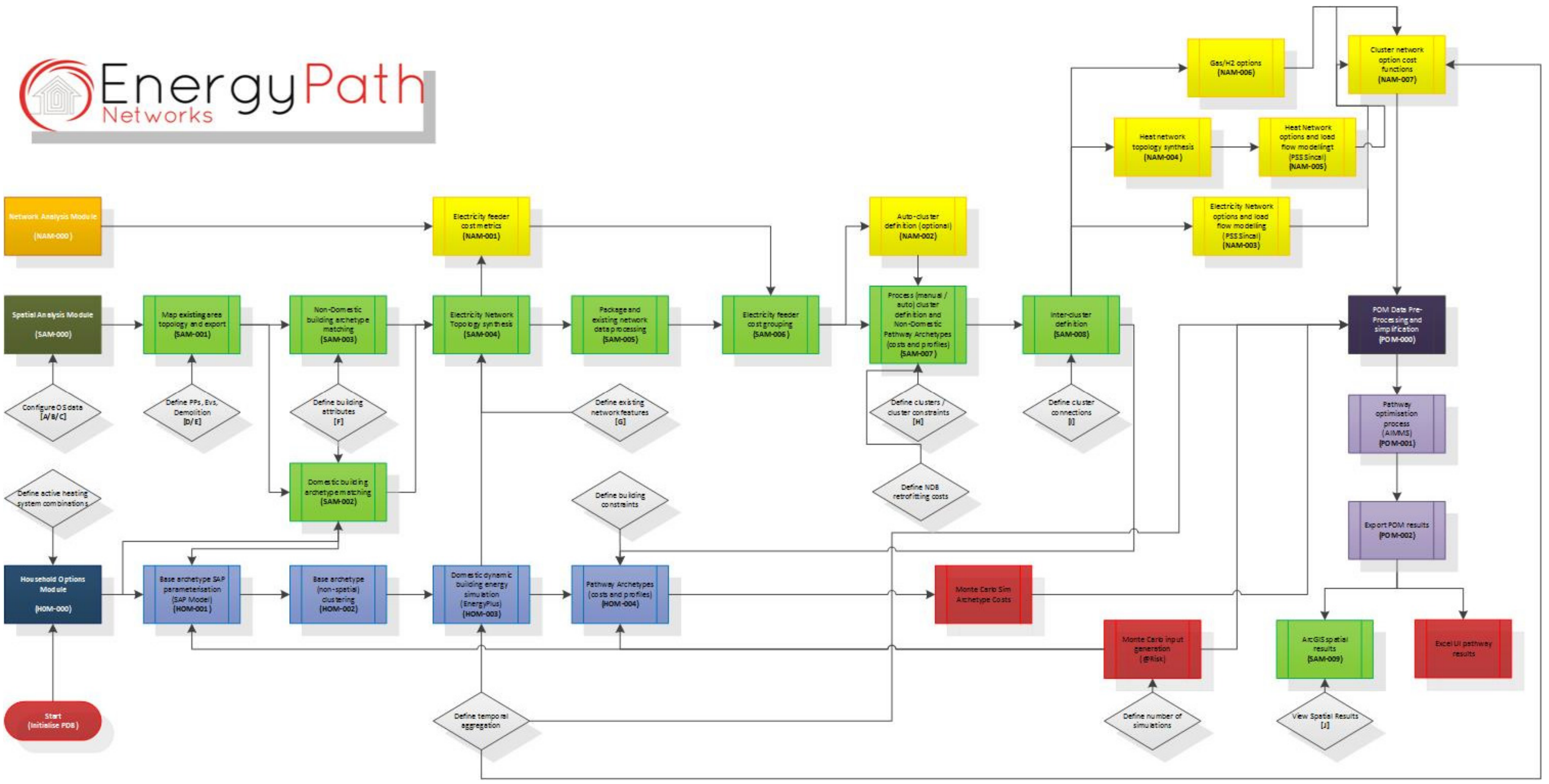


Table 4-1 Functional sub-module libraries

Module	ID	Name	Functional Description	Framework
MCF	Start	Initialise Primary Database	Creates the IDs associated with the master set of Base Archetypes (accounting for the user-defined active heating systems), initialises boundary conditions from ESME	MCF
HOM	HOM-000	Household Options Module (initialisation)	Transfers the relevant set of data from the PDB to the HOM secondary DB.	MCF
	HOM-001	Base archetype SAP parameterisation (SAP Model)	Calculates the extended set of SAP inputted variables for Base Archetypes representing buildings identified in SAM-002 (excluding heating) and passes these to the wrapped ETI SAP model to calculate the set of required inputs for HOM-002 (heat loss rate, thermal mass)	Python + ETI SAP Model
	HOM-002	Base archetype (non-spatial) clustering	Mathematically clusters (using a k-means based algorithm) the Base Archetypes using the subset of SAP inputted data and floor area to a smaller, more manageable number, which share similar energy performance characteristics. This creates a set of *Reduced Archetypes*. The number of clusters within floor area band groups (i.e. reduced archetype energy states) is defined by the user.	Python
	HOM-003	Domestic dynamic building energy simulation	This simulates the operational profile of each Reduced Archetype with different types of heating system and provides the final energy demand profile(s) (>1 where there is optionality related to storage or micro-CHP) associated with meeting space heat and hot water storage requirements. Uses multi-core processing for improved performance.	Python + EnergyPlus
	HOM-004	Pathway Archetypes (costs and profiles)	This module creates the expanded set of Pathway Archetypes which represent the (costed) possible transitions for the domestic stock from the base year to the final pathway year accounting for changes in heating systems and insulation packages (and their associated costs) as well as building and cluster level constraints where applicable. These reflect choices that can be made by the POM. The process for creating these draws on information from the Reduced Archetype definition to make the total number of pathways more manageable. The pathway archetype represents transition of final archetypes i.e. reduced archetypes that have similar costs hence representing group of base archetypes with similar energy and transition cost. The cost is calculated for each cluster based on the relevant stock of base archetypes represented by the pathway archetype. Uses multi-core processing for improved performance.	Python
SAM	SAM-000	Spatial Analysis Module (initialisation)	Transfers the relevant set of input data from the PDB to the SAM secondary DB.	MCF
	SAM-001	Map existing topology and export	This sub-module maps the key existing and future planned spatial elements in the LA including buildings, road topology, electricity and heat network component sizing, PV and EV uptake. It allows the user to define at a unit level the location of planned or	ArcGIS

			potential new build energy system features (e.g. the location of a new build housing estate or a polygon package for network-connected supply technologies)	
	SAM-002	Domestic building archetype matching	This sub-module defines the building-level attributes for all domestic dwellings at address level, and matches each individual building with one of the Base Archetypes defined in HOM-000. The user can also define known building attributes at address level or en-masse using spatial ArcGIS layers, which are then factored into the building matching algorithm. This matching uses external data sources e.g. GeoInformation, Xoserve data to define key attributes at an address level, while more aggregated datasets at GOR level are then used to statistically assign remaining attributes to each address.	Python
	SAM-003	Non-Domestic building archetype matching	This sub-module defines the activity class and floor area for non-domestic buildings at address level. The activity class is based on the OS basefunction, Scat code or other external address level data source. The user can also define the relevant basefunction or ScatCode directly in the OS buildings layer, where these are known.	Python
	SAM-004	Electricity Network topology synthesis	This sub-module synthesises an electricity network topology for the LA – for use in the load flow modelling and costing by NAM.. It utilises data on building location, road network topology and DNO data on the locations of existing substations. The synthesis is based on connecting each address to its nearest substation while following the road network. The user also has the ability to define network connectivity data, where available, including: full network topology (connection of each address/LV substation to its upstream LV/HV substation via LV/HV feeder), HV network only (connection of each LV substation to its upstream HV substation via HV feeder) and partial LV network (LV addresses that are served by LV substations located within their buildings), It also provides topology information relevant to NAM-007 such as the total length of roads on gas grid.	ArcGIS + Python
	SAM-005	Package and existing network processing	This function exports user defined data to the PDB related to the location of the defined polygon packages and known existing network features of the synthesised network topology (i.e. size of heat pipes and LV/HV feeder links)	ArcGIS + Python
	SAM-006	Cluster input layer creation	This sub-module provides the user with the input layers on the electricity network topology showing the network connectivity at LV feeder, LV substation, HV feeder and HV substation levels. This, along with the layer showing auto generated cluster definition by NAM, provides all of the information layers in ArcGIS to inform the user defined clustering process in SAM-007.	ArcGIS + Python
	SAM-007	Process (manual / auto) cluster definition and Non-domestic pathway costing	This sub-module allows the user to define clusters to spatially aggregate the local area. A number of different approaches are available to define the clusters (manual, semi-automated, automated) along with different information layers reflecting electricity network topology information. The user can define spatially contiguous clusters for: LV feeder(s) and downstream, LV substation(s) and downstream, HV feeder(s) and downstream and HV substation(s) and downstream. The user defined cluster definitions	ArcGIS + Python

			are then checked for meeting rules around inclusion of spatially contiguous elements of electricity network and the results are displayed as an ArcGIS layer. This layer gives feedback validating the clusters that meet the network rules and giving simple instructions to modify the cluster where these rules are broken. Following creation of the clusters, post-processing maps all energy system features to each cluster for use in subsequent NAM and POM sub-modules. This module also defines potential pathways for heating system transition in the non-domestic buildings and the annual cost for these pathways. The user is required to define the starting option and the potential options available for each activity class as well as the cost banding. The non-domestic buildings are then grouped based on cluster, similar option transitions, cost banding and demand banding.	
	SAM-008	Inter-cluster connection definition	Once the clusters have been defined in SAM-007 the existing and potential inter-cluster connections for heat, gas and hydrogen are defined by the user. Electricity network meshing connections can also be defined by the user between any two nodes at the LV/HV level. The resulting data is processed for use in the NAM and POM sub-modules	ArcGIS + Python
	SAM-009	ArcGIS spatial results	This function takes the cluster-level POM results (which have been pushed to the primary DB after POM-002) and automatically creates a series of layers within ArcGIS to view the spatial results.	ArcGIS + Python
NAM	NAM-000	Network Analysis Module (initialisation)	Transfers the relevant set of input data from the PDB to the NAM secondary DB.	MCF
	NAM-001	Electricity feeder cost metrics (optional)	This sub-module prepares the underlying nodal representation of the electricity network and associated data (e.g. length of links, location of demand points, etc) using data from SAM-004. It then calculates a normalised reinforcement cost metric for each individual LV feeder via calculation of allowable load through SINCAL. It also provides a database of viable reinforcement options that are parameterised into more aggregate options in NAM-007.	Python + VBS + PSS Sincal
	NAM-002	Auto-cluster definition (optional)	This optional sub-module undertakes a series of tests involving different 'cluster' shapes (of different sizes) and calculates a series of metrics for heat and electricity networks, which are then compared against defined thresholds. The end result is a set of suggested cluster definitions which respects the underlying electricity network topology, but which allows for a "good" representation of both heat and electricity cost functions within the same cluster, when these are calculated from the other NAM sub-modules. These suggested cluster shapes can be used by the user as part of SAM-007.	Python + VBS + PSS Sincal
	NAM-003	Electricity Network (HV) options and load flow modelling	This sub-module prepares the underlying nodal representation of the electricity network and associated data (e.g. length of links, location of demand points, etc) using data from SAM-004. It then generates a series of pre-defined network reinforcement options to be tested by the operational analysis tool (PSS SINCAL) for the HV network. It then assesses the allowable capacity limits, associated with each reinforcement configuration, subject to overarching constraints (e.g. on voltage) and the base costs	Python + VBS + PSS Sincal

			associated with each option. It also provides a database of viable reinforcement options that are parameterised into more aggregate options in NAM-007.	
	NAM-004	Heat network topology synthesis	This uses ArcGIS functionality to generate synthesised network topology information for the Heat Network (as per fully automated synthesis for electricity in SAM-004), but with an independent distribution heat network for each cluster as defined via SAM-007. As part of this process it also calculates the estimated heat load weighted centre for the cluster.	ArcGIS + Python
	NAM-005	Heat Network options and load flow modelling	As per NAM-003 this prepares the underlying nodal representation of the heat network and associated data for use in PSS SINCAL using data from NAM-004. It then generates a series of pre-defined network reinforcement options to be tested by the operational analysis tool (PSS SINCAL) *at a cluster (and inter-cluster) level*. It then assesses the allowable capacity limits, associated with each reinforcement configuration, subject to overarching constraints (e.g. maximum pressures) and the base costs associated with each option. It also provides a database of viable reinforcement options that are parameterised into more aggregate options in NAM-007.	Python + VBS + PSS Sincal
	NAM-006	Gas/H2 options	This creates a simplified set of Gas/H2 operation, extension, conversion or decommissioning options at cluster and inter-cluster level.	Python
	NAM-007	Cluster network option cost functions	This sub-module consolidates the more detailed reinforcement and new build options from NAM-003/5/6. For electricity it first consolidates these at the cluster level and then creates a parameterised set of simpler options for use in the POM (including capacity, cost, losses and other data). For heat networks the base cost functions are already at cluster level, but the simplification process is then applied.	Python
POM	POM-000	POM Data Pre-Processing and simplification	Views and stored procedures within the primary database (within the prepom schemas) contain the logic to automatically generate the full set of required POM inputs from other primary DB tables (which have themselves been generated as outputs from the HOM, SAM and NAM sub-modules). The pre-processing also includes aggregation logic for timeperiods and timeslices and handling of demand diversity (through exogenous scaling factors).	MCF
	POM-001	Pathway optimisation process	This solves a large number of Mixed-Integer Programming (MIP) problems generated by POM-001 to obtain a least-cost solution (within a user defined tolerance threshold). A branch & cut algorithm is used which creates a tree of LP sub-problems for each combination of the integer variables. The algorithm can prune sub-branches when it realizes a feasible solution won't be found there or the currently found solution is better than what can be found exploring the pruned sub-branch.	AIMMS + CPLEX solver
	POM-002	Export POM results	This function pushes the POM results back into the PDB so that they can be viewed within the Excel UI, or via the ArcGIS visualisation process in SAM-009	MCF

OTHER	MCF-001	Master Control Framework	This is responsible for managing the overall process execution flow for the entire tool including, transfer of data between primary and secondary databases, and execution of each sub-module, data validation checking and progress logging.	Python
	EUI-001	Excel UI pathway results	This undertakes two main functions. – It allows the user to read/write/update input and results data to/from the primary database (no direct user interaction is required with the secondary databases), outside of input data defined through the ArcGIS UI via the SAM sub-modules. – It also allows the user to initiate the process execution within the MCF / Monte Carlo input generation.	Excel + VBA
	MCG-001	Monte Carlo input generation	Whilst this is technically contained within the Excel UI it is designed as an isolated process, which reads from a number of primary DB input tables and generates a series of Monte Carlo simulated inputs, which are written back to the primary DB for use elsewhere in the tool. All Monte Carlo simulated inputs are used directly within the POM-001 process, with two exceptions. – Simulated insulation U-value parameters which are required as an input to the SAP parameterisation in HOM-001 – NDB retrofitting cost as calculated in SAM-007	Excel + VBA + @RIsk

4.1. Basic end-to-end process flow

The overall process flow across the tool is shown above in Figure 4-1 (each of the sub-modules are then described in Table 4-1). The execution of the tool is managed through the **[Process Flow]** sub-menu screen as described in section 8.4.1, which mirrors the above diagram.

The diamonds in the process flow diagram are key points of user input (outside of all other data which it is assumed has been entered appropriately by the user in the primary database prior to execution). If the user updates these particular points it is then necessary re-run all 'downstream' sub-modules that will be affected by these changes as outlined in the process flow diagram.

The key points of user input cover:

- *Define active heating system combinations* – which is used to limit the possible (valid) combinations of primary heating / secondary heating / storage technology /heating controls (within HOM-000), active within the overall tool to manage complexity
  - This is updated via the *hom.si_ActiveHeatingSystemCombinations* table accessed under UI sub-menu ***[HOM] -> [Pathway Archetype Configuration]***
- *Define potential non-domestic options at activity class level and their valid transitions* – which is used to limit the possible (valid) heating system and insulation package transitions (within SAM-007), active within the overall tool to manage complexity

  This is updated via the *mcf.pi_NDB_ActivityClassRetrofittingOptionsMap* and *mcf.pi_NDB_ValidRetrofittingOptionTransitions* tables
- *Define temporal aggregation* – which is used to simplify the complexity of the POM optimisation (via the pre-processing in POM-000) with respect to time/buildperiods, characteristic days and diurnal timeslices. It is also used to limit the number of buildperiod vintages, which are assessed within EnergyPlus (HOM-005) and to streamline the number of final network cost functions produced by NAM-007
  - This is updated via the following tables accessed under UI sub-menu ***[MCF] -> [Temporal configuration]***
  - mcf.si_FinalTimePeriods / mcf.si_FinalBuildYears
  - mcf .si_FinalDiurnalTimeSlices / mcf .pi_DiurnalTimeSlicesMap /
  - mcf .si_FinalCharacteristicDays / mcf .pi_CharacteristicDaysMap
- Define the number of sets of Monte Carlo simulated inputs used within the POM optimisation. The majority of simulated inputs are used directly at the POM pre-processing stage (POM-000). However, simulated insulation performance is an input to the SAP parameterisation of the Base Archetypes (HOM-001)
  - This is updated via the UI sub-menu ***[Simulate] -> [Simulation Manager]***
- ArcGIS steps [A to I], which are described further in section 9

Multi-scenario data management

As described in section 3.1 multi-scenario data management is focused on a number of discrete parts of the tool such as target temperature profiles for use in EnergyPlus. The user interacts with the multi-scenario data management via 3 areas:

- **Input data configuration**: the *mcf.si_Scenarios* table (accessible through the Excel UI as per all other input tables) is used to define the number of scenarios and whether they are considered to be 'active' (i.e. the data should be processed) when running the tool. Related input tables (e.g. *hom.pi_TargetComfortTemperature, mcf.pi_NDB_EnergyBenchmarks and mcf.sg_clusters*) have an additional column for "scenario" such that data can be entered against the multiple defined scenario names
- **Process execution**: the *[Process Flow]* screen described in 8.4.1 allows the user to run the tool for both the "default scenario" or automatically batch process a number of steps, cycling through the data for all scenarios which are defined as 'active'
- **Results**: the results tables returned from the POM to the Primary database have a scenario index (for both the default and other active scenarios which have been run) which can be used to compare results across the scenarios in the standard Excel UI (see section 8.5.1). To be clear, the POM processes a single scenario at a time (like the HOM & NAM) but appends the relevant scenario name to the results tables being returned.

  In addition, the spatial results layers which are generated for ArcGIS automatically in SAM-009 produce results for all active scenarios.

5. API Specification

5.1. Overview

This section provides an overview of the API design specification for the tool, the detailed description of each individual API (e.g. purpose and arguments) are *catalogued as part of the code documentation in section 12.*

The API within the MCF (api.py) has been defined in a flexible manner, given the modularity of the underlying tool. A central API has been defined as:

epdt.pro.RunProcess('Database Name', 'schema', 'ProcessName')

This is designed to execute a call based on argument information defined in the primary database *mcf.pi_ProcessDefinition* table. The information within this table provides information specific to the execution of each (sub)-module API. This way the MCF has the ability to execute (in the appropriate defined sequence) every sub-module API that has been 'exposed' in its code to the MCF layer.

As part of this *epdt.pro.RunProcess()* can be used to indirectly pass arguments specific to each sub-module API, via information that the user has defined in the *mcf.pi_ProcessDefinition* table - i.e. the MCF can access the full functionality of exposed APIs. Where this is blank each sub-module uses its own default argument(s), in most cases this is simply 'execute' the entire sub-module process.

The choice of sub-module API exposed directly to the MCF layer is a trade-off in complexity of defining the appropriate process execution flow and the degree of fine control. Where a process is fast it is generally sufficient to define the API to execute the entire process only. Therefore we have tended to focus exposure at the MCF layer on key processes that a) repeat a cycle of logic with b) significant overall execution time.

For example, the operation of EnergyPlus may require the simulation of between 10^2 to 10^3 different building simulations (or "Reduced Archetypes"), which is a time consuming process. During a single batch run some of these may 'fail' due to e.g. input data errors, whilst the majority may complete 'successfully'. Once the input data for the failed simulations is fixed it would preferable to re-execute only these via the MCF rather than having to repeat the full set.

5.2. MCF API example

The example below assumes an illustrative subset of the *mcf.pi_ProcessDefinition* table as below.

Process ID	Task Seq	Module	Process Name	Step	Task	Execute	Function Name	Function Arguments
1	1	HOM	Call RunEPAll	Call RunEPAll	Function call	1	epdt.hom.api.computeEPlus	
2	1	HOM	Call RunEPSubset	Call RunEPSubset	Function call	1	epdt.hom.api.computeEPlus	by_ID, RA002

Within the MCF api.py, executing the API as either of

```
epdt.pro.RunProcess(PDB, 'mcf', 'Call RunEPAll')

epdt.pro.RunProcess(PDB, 'mcf', 'Call RunEPSubset')
```

Indirectly executes the EnergyPlus sub-module API *epdt.hom.main.computeEPlus()* using the information from row 1 or 2, respectively, from the *mcf.pi_ProcessDefinition* table.

Passing no argument to *epdt.hom.main.computeEPlus*(), as per *'Call RunEPAll'* automatically runs all reduced Archetypes in EnergyPlus whereas the *'Call RunEPSubset'* equivalent effectively executes the API as:

```
epdt.hom.main.computeHOM(by_ID, RA002)
```

I.e. instructing EnergyPlus to run one Reduced Archetype only.

6. Installation and configuration

6.1. Underlying software installation

The following sections outline the installation of the software. For 3rd-Party software packages we only describe steps, which are additional or different from the *default* installation process. It is recommended that all software is installed onto the same drive e.g. C:

The core tool files and databases are provided within an EDPT.ZIP file (or equivalently pulled from the Assembla GitHub repository), the instructions assume that this is unzipped to a folder labelled C:\EDPT.

6.1.1. Overarching pre-requisites

The following are overarching pre-requisites for the EPN tool

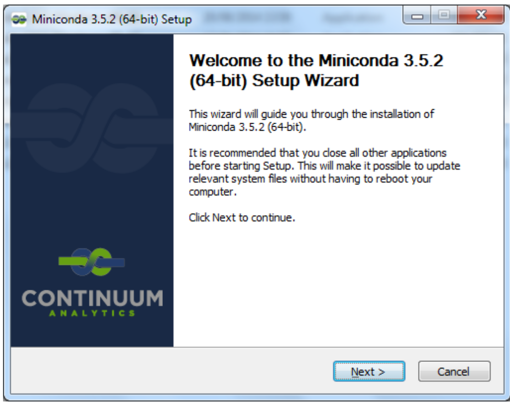
- Software
    - A 64bit version of Windows – EPN has only been tested to-date with Windows 7 Enterprise and Professional 64bit editions
    - MS SQL Server Express 2008 R2 or MS SQL Server Commercial as outlined in section 6.1.3, as this has been selected to be cross-compatible with all relevant EPN third party components
    - ArcGIS for Desktop requires Microsoft .NET Framework Version 3.5 SP1[13] to be installed first
    - Notepad++ v7.3.1 is not required, but is useful for viewing text based files (e.g. *.log, *.ini, *.py etc)
- Hardware
    - Database storage is the main factor in relation to the size of the EPN tool, it is recommended that a minimum of 50 GB of space is set aside for the tool itself (in addition to space required by the underlying software)
    - Memory – whilst 64bit operating systems allow use of the disk page file to extend the available system memory this is significantly slower than relying on system memory. In the overall tool process flow (see section 4.1) the AIMMS optimisation model (POM-002) and a number of EPN sub-modules that have been configured to run in parallel across multiple CPU cores (e.g. HOM-005 and HOM-006) are the key drivers of high memory use. 32GB+ of physical memory is strongly recommended. Windows 7 Pro has a constraint of 192GB of accessible memory, Windows 10 Pro allows 2TB and Windows Server 2012 allows 4TB.
    - A fast multi-core processor is strongly recommended given the intensive and parallelised processes identified above (AIMMS can already use multiple cores

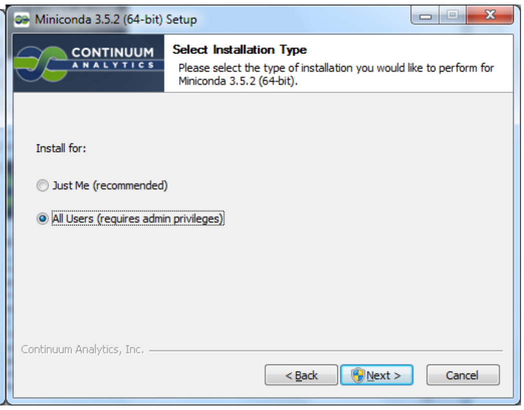
[13] http://www.microsoft.com/en-gb/download/details.aspx?id=22

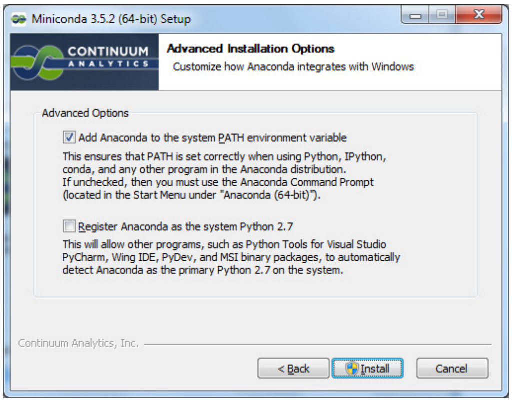
for a single optimisation process). The development machine for EPN was based on an Intel Xeon 3.1 GHz processor with 32 cores

6.1.2. Python

Python 2.7[14] (64bit) has been used to code key functional requirements across the HOM, SAM, NAM and MCF as well as 'wrap' key 3rd party software. To facilitate installation and management of python we use Miniconda 64bit for Python 2.7 from http://conda.pydata.org/miniconda.html







Once installed the user must first (via the DOS console – select *Start -> Main Console Module*) create the EDPT environment and install the relevant python libraries used by different parts of the tool

1. Create a new environment called "epdt", with python and pip packages:

   conda create -n epdt python=2.7.11=4 pip=7.1.0=py27_1 setuptools=18.1=py27_0 wheel=0.24.0=py27_0 vs2008_runtime=9.00.30729.1=1

2. Activate the newly created environment:

[14] https://www.python.org/

activate epdt

3. Install the packages for data analysis (pandas, scikit-learn, and their dependencies):

conda install mkl=11.3.3=1 mkl-rt=11.1=p0 numpy=1.9.3=py27_3 pandas=0.16.2=np19py27_0 scipy=0.16.0=np19py27_0 scikit-learn=0.16.1=np19py27_0 nose=1.3.7=py27_0 pytz=2016.4=py27_0 six=1.10.0=py27_0 python-dateutil=2.5.3=py27_0

4. Install matplotlib and its dependencies:

conda install matplotlib=1.4.3=np19py27_1 pyqt=4.10.4=py27_1 pyparsing=2.0.3=py27_0

5. Install the packages to connect to the SQL DBs:

conda install pyodbc=3.0.7=py27_0 sqlalchemy=0.9.8=py27_0

6. Install ipython and its dependencies:

conda install ipython=3.2.1=py27_0 pyreadline=2.0=py27_0

7. Install pywin32:

conda install pywin32=219=py27_0

8. Navigate to folder location of EPDT code:

C:\epdt\MODEL\CODE
(Or similar)

9. Install EPDT python package to epdt environment

pip install -e .

10. Check EPDT code installed correctly:

pip show epdt

6.1.3. MS SQL Server

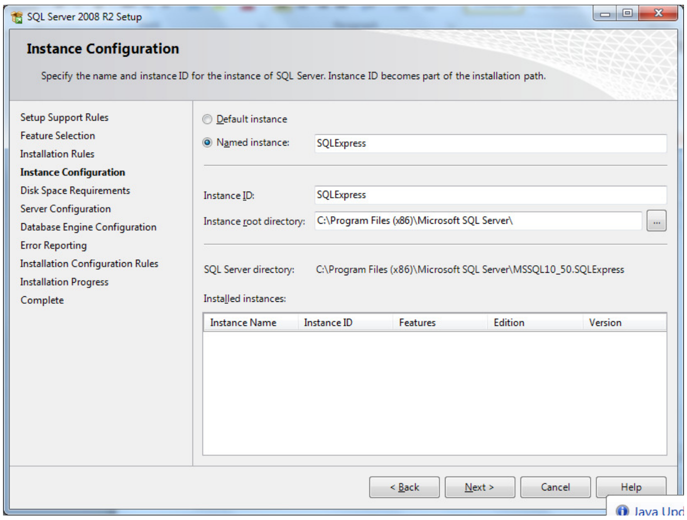
A key part of the Energy Path model is Microsoft SQL Server[15] Commercial 2008 R2 (64bit) and management studio[16]. This is used for all databases with the exception of ArcGIS which uses its own geodatabase format to store spatial information (relevant attribute data required from this is still stored within the MSSQL databases).

[15] http://www.microsoft.com/en-gb/server-cloud/products/sql-server/

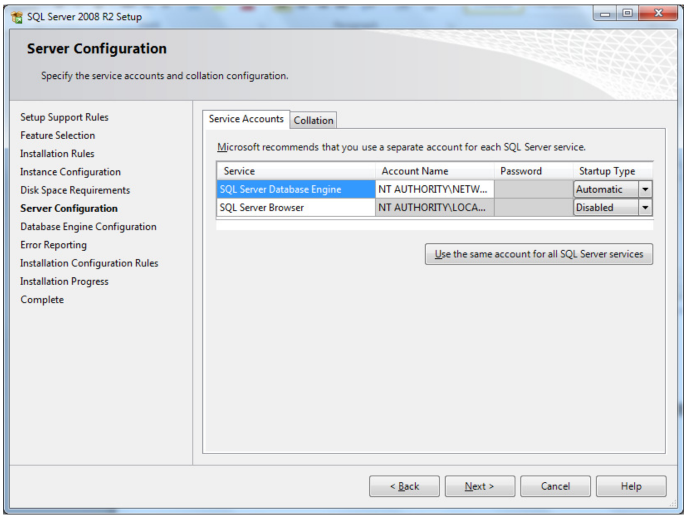
[16] MS SQL Server Express 2008 R2 can be used if the size of each individual database is maintained at <10GB and data processing is limited to a single core.

The following instructions are compiled for Microsoft SQL Server Express edition, but are virtually identical to Microsoft SQL Server Commercial edition. Detailed setup guidance can be found online[17]. After downloading:

1. Run the installation file and select "New Installation or add features to an existing installation"
2. Accept the license terms and click next
3. Select All components of the model to install and click next
4. Select 'Named instance:' of instance configuration and click next



5. Change the account name of SQL Server Database Engine to "NT Authority\System" or "NT Authority\Network" and click next



6. On the 'Database Engine Configuration' screen ensure 'Windows authentication mode' is selected.

---

[17] http://blogs.msdn.com/b/petersad/archive/2009/11/13/how-to-install-sql-server-2008-r2-express-edition-november-ctp.aspx

7. Complete the setup with the remaining default options. You may need to reboot your computer to complete the installation[2].

Once the default installation is complete the following settings must be changed through the management studio

- *Server properties -> security -> server authentication*
  - Apply *SQL server and windows authentication mode*
- Under security create a new login "*epdtadmin*" with password "*Epdt1234*"
  - This is used to allow the MCF to connect to all attached databases
- *Server properties -> Advanced*
  - Set the Maximum Degree of Parallelism to 8[18] (for improving performance)

The *EDPT.ZIP* contains a set of DB files *(.mdf* and *.ldf*) which should be attached to the server instance (see section 9.13.1 for further information on how to undertake this process):

- *EPDT-Primary_vX.X.X*
- *EPDT-HOM_vX.X.X*
- *EPDT-SAM_vX.X.X*
- *EPDT-NAM_vX.X.X*
- *EPDT-POM_vX.X.X*
- *SINCAL_ELEC_SQLSERV*
- *SINCAL_HEAT_SQLSERV*
- ETI-TE

One further temporary *SINCAL_DATABASE* DB is created and attached via the initial SINCAL setup, described in section 6.1.9.

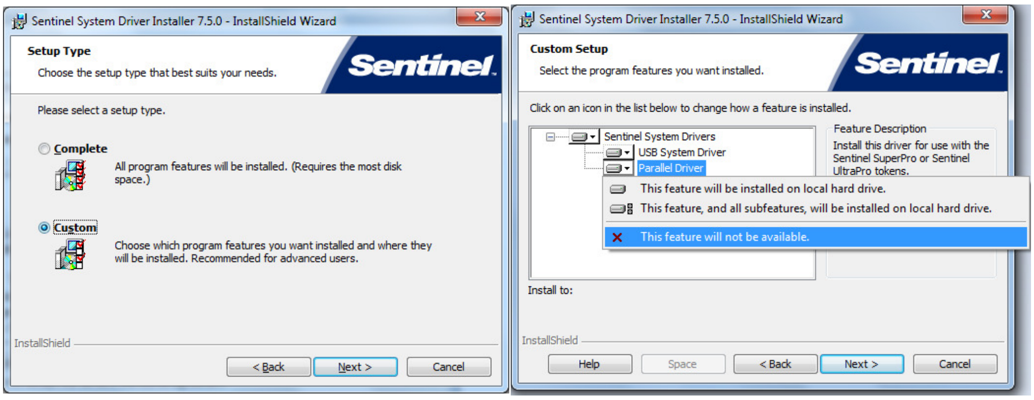
6.1.4. AIMMS

AIMMS[19] 4.23 (64bit) is used within the POM for generating the optimisation problem. This includes the CPLEX 12.6.3 solver. However, POM should also work with any other versions after AIMMS 4.0.

AIMMS has discontinued issuing USB dongle AIMMS license. However, if you are still using an USB dongle AIMMS license, the Sentinel System Driver needs to be installed first. During the installation process, users should select custom setup type and disable parallel driver as shown below.

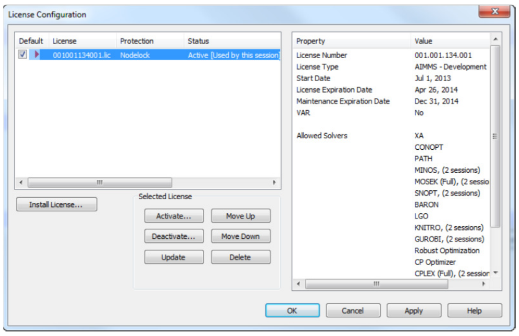
---

[18] This depends on the machine configuration. This setting applies for a machine with 32 cores and ~200Gb of RAM.
[19] http://www.aimms.com/

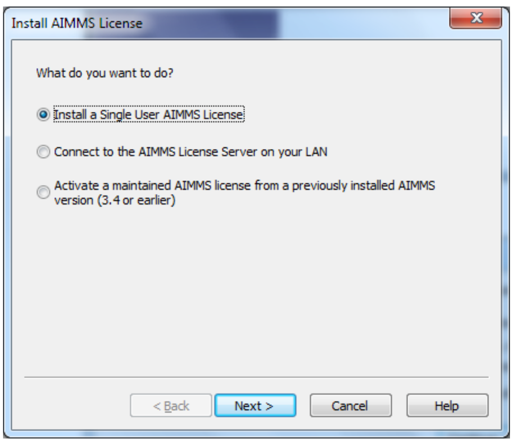


AIMMS offers several types of licenses. Below gives the details setting up single-user personal nodelock license. Full installation instructions for other types of licenses can be found here[20].

1. AIMMS 4.x is now installation-free. When you start up AIMMS 4.x for the first time, AIMMS will open the License Configuration dialog box below:

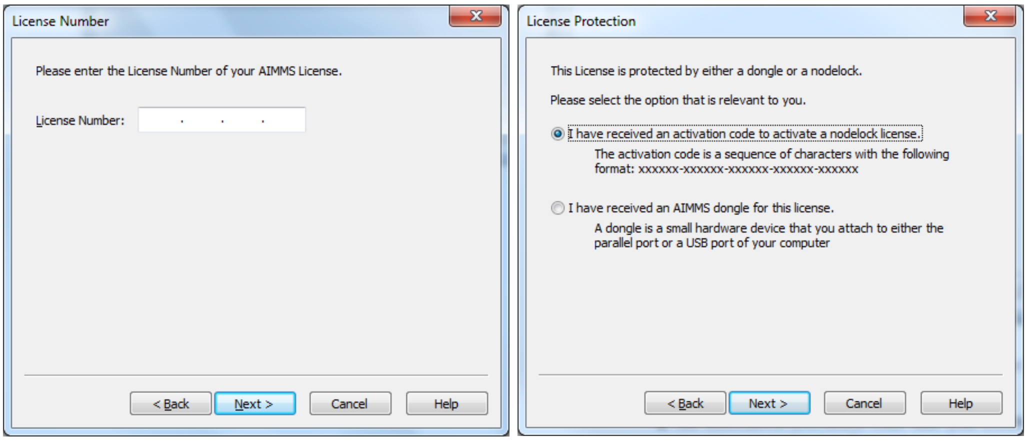


2. Click Install License if you have received a valid license and the associated activation code. Select “Install a Single User AIMMS License” and click Next.

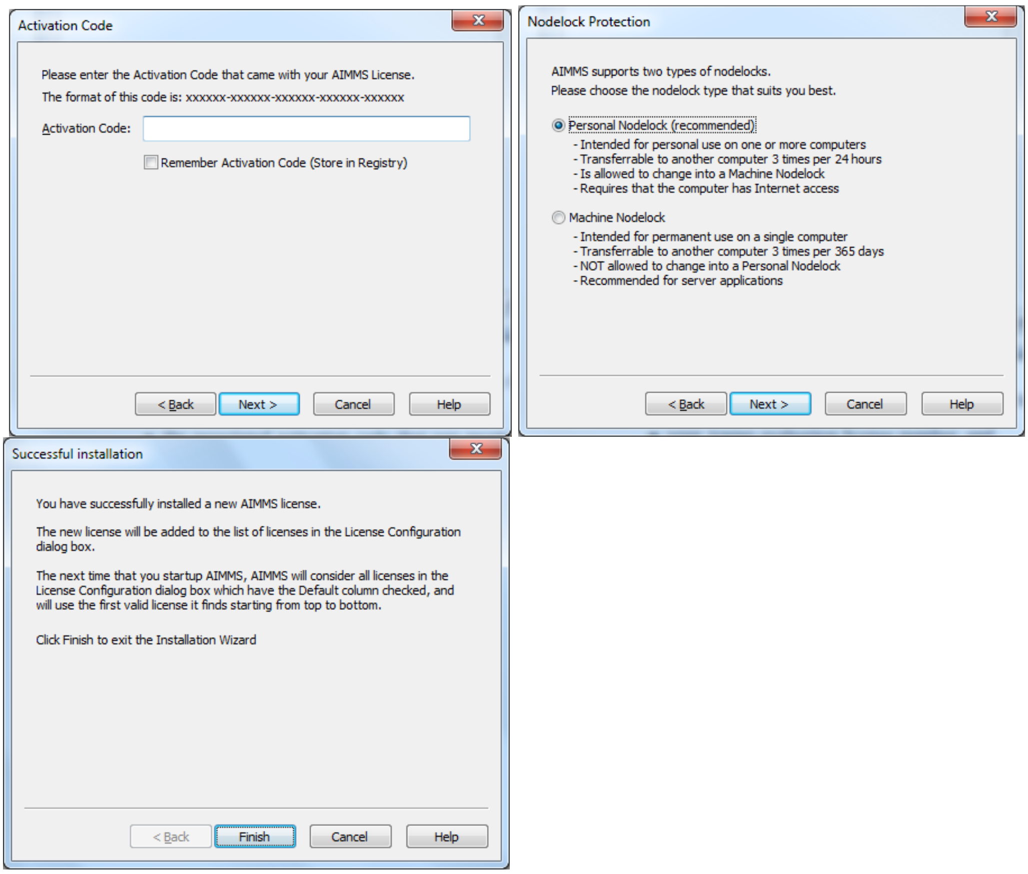


[20] http://download.aimms.com/aimms/download/data/4.17/1.374/AIMMS_release-Win64_4.17.pdf

3. Enter your license number and click Next and then select first option if you have received an activation code.

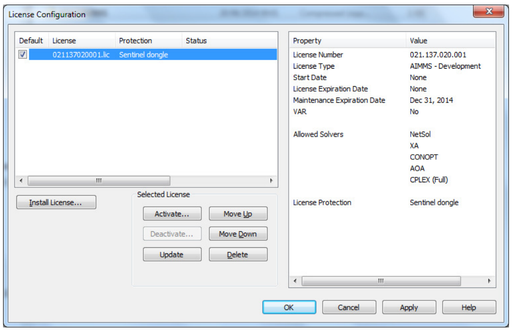


4. Enter the activation code that you received from AIMMS to activate a nodelock license. You have the choice to request a personal nodelock or a machine nodelock. Personal nodelock is our preference during development phase since it can be transferred to another computer 3 times per 24 hours whereas machine nodelock can only be transferred 3 times per 365 days. More information about their differences can be found in the installation instructions manual. If the license is valid the subsequent screen will indicate successful installation.

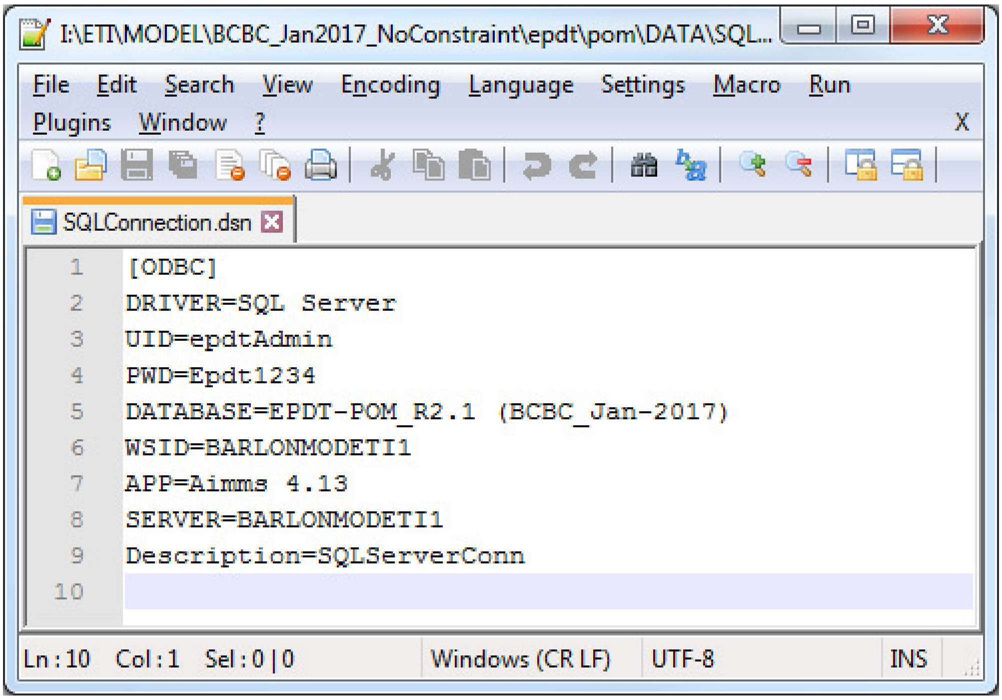


5. The next screen shows the details of licenses installed in your machine. Click Update to retrieve the latest properties and solvers. It should be noted that the solver configuration

parameters are contained within the EPN primary database[21] rather than being configured directly through the AIMMS UI.

- 

6. Once the default and license installation are complete an additional step is needed to map the POM AIMMS model to the appropriate POM secondary DB.

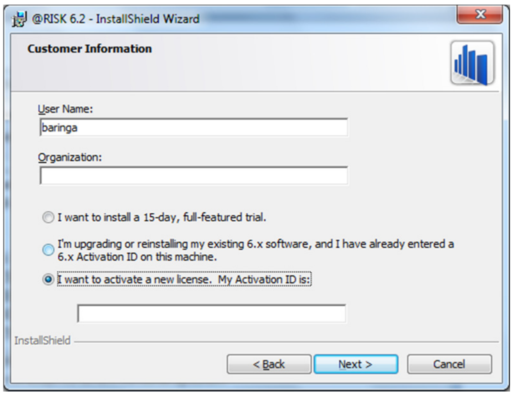
- Open the **.dsn** file in the *\POM\Data* subfolder with any text editor e.g. Notepad
  - 

  - Update with the appropriate server, WSID (i.e. workstation ID, same as server name) and database name

[21] pom.si_POMGlobalParameters

6.1.5. @Risk

@Risk[22] 6.2 is used via the Excel GUI for generation the Monte Carlo simulated inputs. Install @Risk via the default instructions select next with all default settings with the exception of the "Customer Information" screen where the purchased Activation ID must be entered.

- 

Following successful installation it is necessary to ensure that @Risk is configured properly in the main tool Excel UI (as @Risk is itself an Excel 'plug-in' rather than a standalone piece of software):

- Open the *UI*.xlsm file in C:\EDPT
- Ensure '*Data Connections*' and '*Macros*' are enabled and "*Trust access to the VBA project object model*" is checked
  - *Excel Options -> Trust Center -> Trust Center Settings -> Macro Settings*
- Ensure that the following VBA references are checked; *Alt-F11* (to open VBA) *-> Tools -> References*
  - Microsoft Excel 12.0 (or higher) Object Library
  - Microsoft ActiveX Data Objects 2.8 (or higher) Library
  - Palisade @RISK x.y for Excel Object Library where x.y is the version number installed locally
  - RiskXLA[23]
- Save and close the Excel UI.

6.1.6. EnergyPlus

EnergyPlus[24] v8.1 (64bit) is used within the HOM for dynamic building energy simulation. There are no additional steps beyond the default installation. Click next under all steps with the

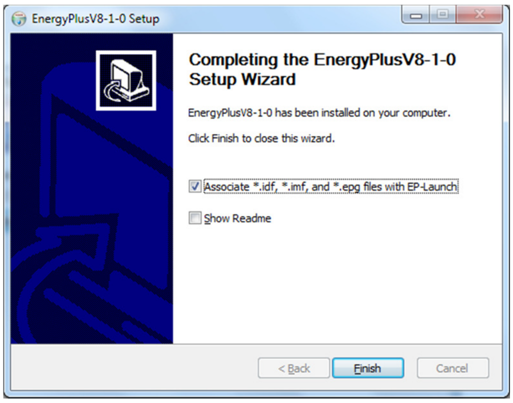
---

[22] http://www.palisade.com/risk/

[23] Note that when this reference is checked the @Risk excel add-in will automatically load when the UI file is opened. If you wish to prevent the @Risk excel add-in from automatically loading then uncheck this reference and re-save the UI

[24] http://apps1.eere.energy.gov/buildings/energyplus/

default selections (n.b. ensure that under the final step “Associate *.idf, *.imf, and *.epg files with EP-Launch is selected)

- 

Following successful installation the user needs to replace the Energy+.idd file in the installation directory with the one contained in the EPDT.zip file.

6.1.7. ETI Thermal Efficiency SAP model

A sub-folder (***eti-te-model***) is included in the main EDPT.ZIP and contains all the necessary files called by the SAP sub-module wrapper in the EPN tool. The SAP model uses the ETI-TE database attached as part of the instructions in section 6.1.3.

The filepath for the ETI-TE SAP model also needs to be configured (along with a number of other 3rd party components) as outlined in section 6.1.10.

6.1.8. ArcGIS

ArcGIS[25] 10.2.2 (64bit) is used within the SAM to support the spatial analysis requirements and spatial elements of the GUI. This includes the Network Analyst and Spatial Analyst extensions along with the Productivity Suite.

Follow the default instructions to install

- ArcGIS_Desktop_XXX.exe
- ArcGIS_BackgroundGP_for_Desktop_XXX.exe
- ProductivitySuite_vXXX.exe

After default installation is complete, you will need to

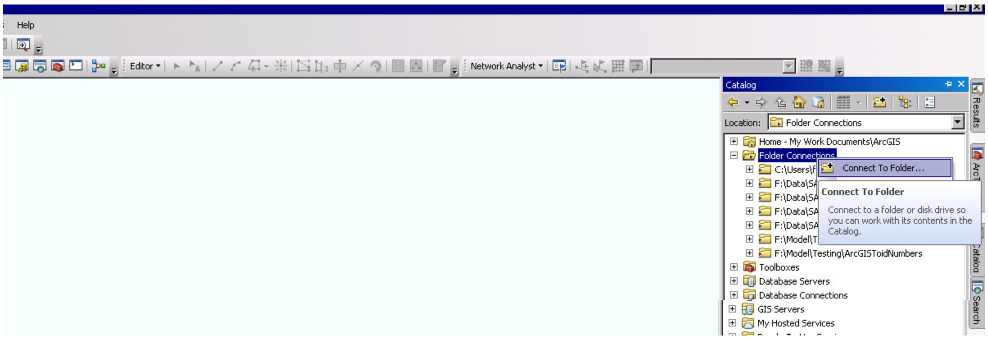
- Enter your authorization number for ArcGIS for Desktop via Start -> All Programs -> ArcGIS -> ArcGIS Administrator.
- Import license for ProductivitySuite via Start -> All Programs -> ProductivitySuit v3.4 -> Licence Manager.

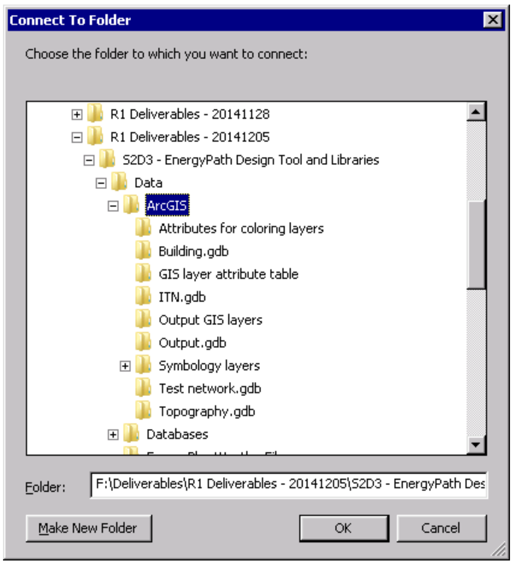
---

[25] http://www.esri.com/software/arcgis/arcgis-for-desktop

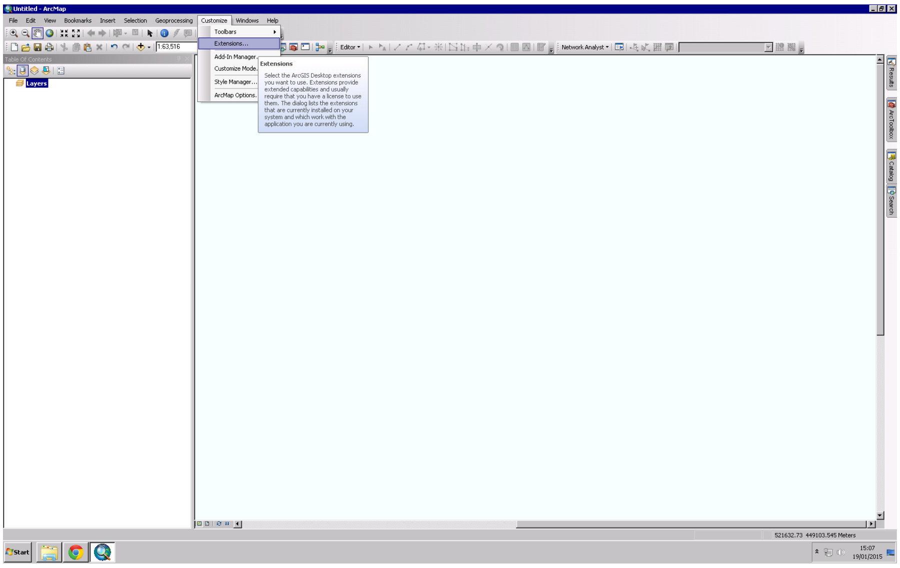
The EDPT.ZIP contains an ArcGIS Geodatabase folder. Upon opening ArcGIS for the first time it is also necessary to

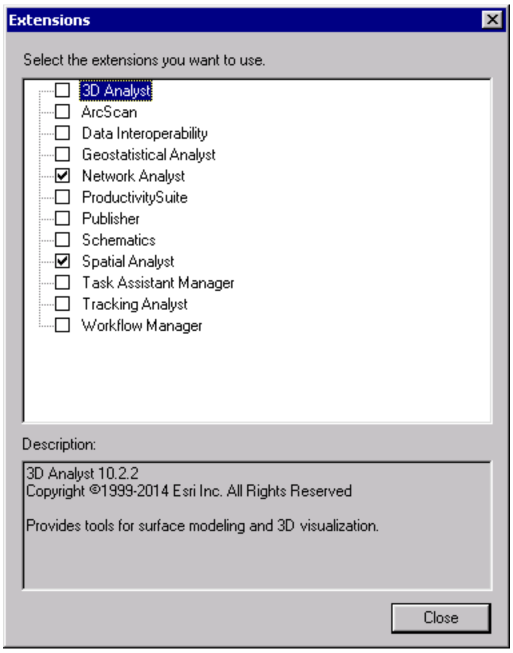
- Create a *Connection* to this folder via the *Catalog* window

  – 

  – 

- Add the Network Analyst and Spatial Analyst extensions to the basic ArcGIS setup from the toolbar by selecting Customize -> Extensions

– 

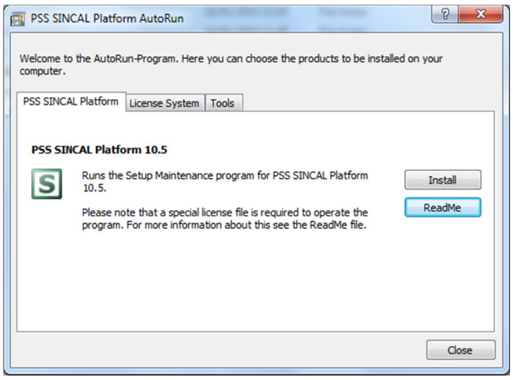
– 

6.1.9. PSS Sincal

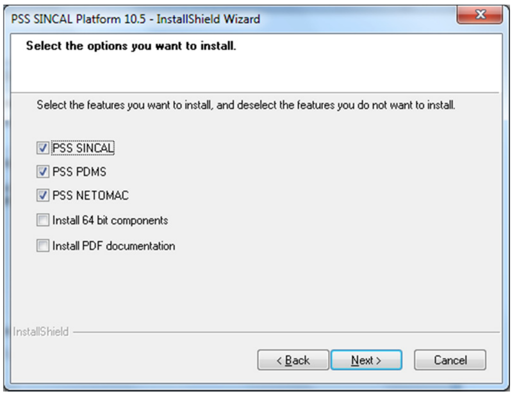
PSS Sincal[26] 10.5 (***32bit***) is used within the NAM for steady state analysis of electricity and heat distribution networks. This includes the Electro and Heating modules.

The user should execute the Sincal autorun.exe file and follow the default installation instructions under the first tab “PSS SINCAL Platform” -> install

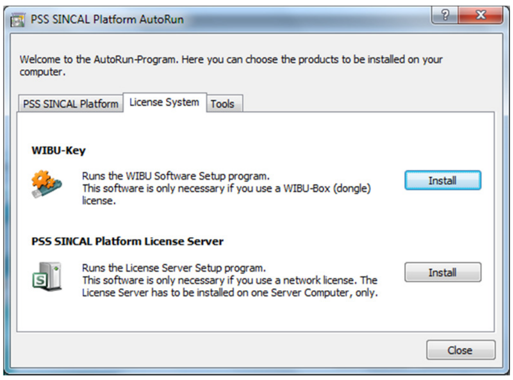
[26] http://w3.siemens.com/smartgrid/global/en/products-systems-solutions/software-solutions/planning-data-management-software/planning-simulation/pages/pss-sincal.aspx

- 

However, it is important to ensure that the “selected options” do *not* include “Install 64bit components”[27]

- 

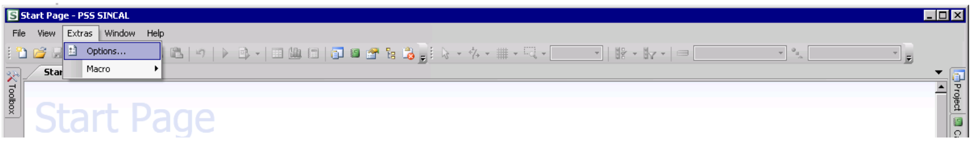
Depending on type of license, the WIBU-key dongle driver or PSS Sincal Platform License Server needs to be installed. This process is initiated by selecting the “License System” tab on the first autorun window and following the relevant instructions (which depend on the license type the user has purchased)

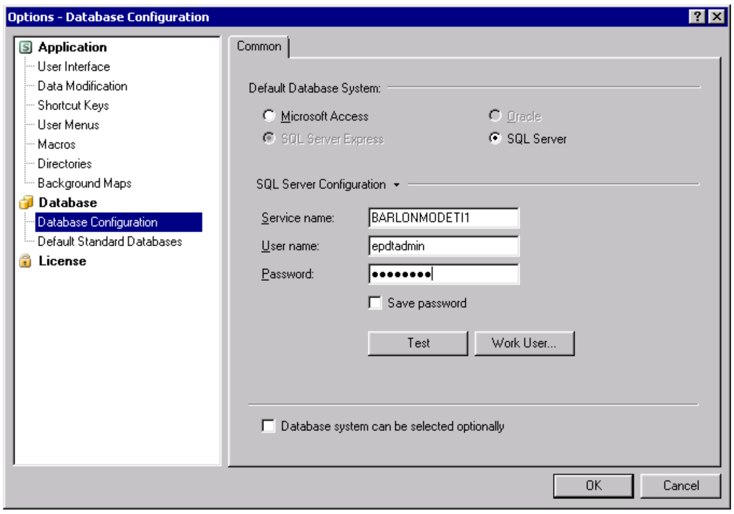
- 

After installing Sincal it is necessary to:

- Open the main Sincal UI to complete the configuration

[27] This is for compatibility reasons, due to the use of CScript to run the VBS-based COM objects for the automatic wrapping of Sincal in the NAM.

- From the toolbar select -> extras -> options ->Database configuration-> SQL server and connection details as per those used in 6.1.3
- This automatically creates the final SINCAL_DATABASES DB
- 

- 

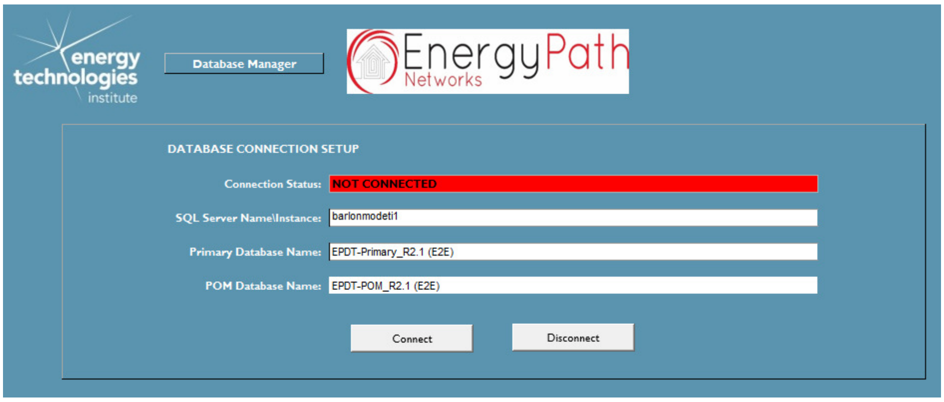
- Within the EDPT.ZIP there are two additional **.sin** files required by Sincal, which must be located in the same directory as their corresponding DBs (these should automatically be extracted to the relevant directories by default when the ZIP files is uncompressed)
  - *SINCAL_ELEC_SQLSERV DB* and *SINCAL_Elec.sin*
  - *SINCAL_HEAT_SQLSERV DB* and *SINCAL_Heat_SQLServ.sin*
- If the NAM is to be used in multicore mode for load flow analysis (NAM001, NAM003, NAM005) the databases *SINCAL_ELEC_SQLSERV DB and SINCAL_HEAT_SQLSERV DB* must be detached and copied to produce a database for each core, named in format *SINCAL_ELEC_SQLSERV_1, SINCAL_ELEC_SQLSERV_2*, etc.

6.1.10. Tool setup and configuration

Upon completing the installation of the underling software there are two further setup steps:

- Update the *Edpt.ini* file (included within the *EDPT.zip*) - where necessary - to reflect
  - The installed file paths for the 3rd party software
  - The names of all EDPT DBs
  - The path to the ArcGIS data folder
- In the main Excel UI on the *Database Manager* screen complete the *DATABASE CONNECTION SETUP* details with the relevant parameters from section 6.1.3
  - Click *Connect* to confirm successful connection and the save the Excel UI file

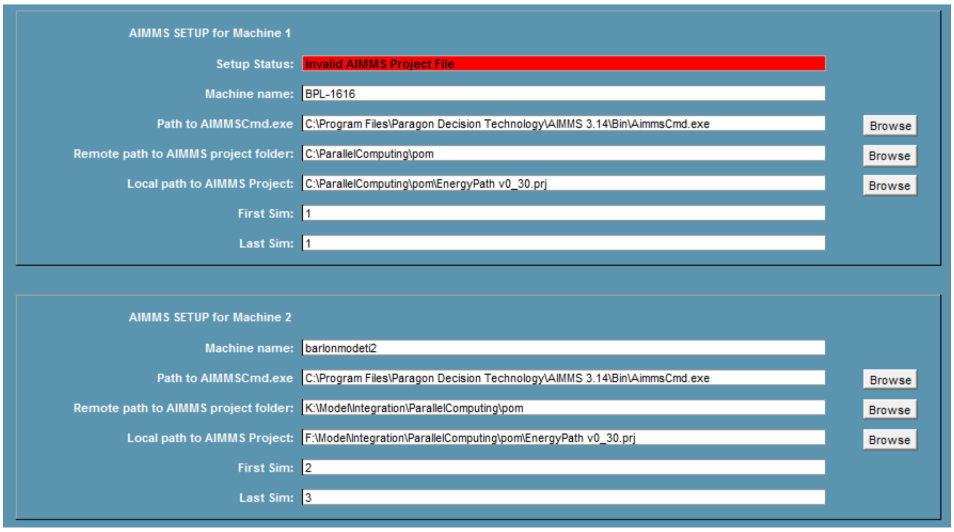
– It should be noted that the Excel UI ***only*** needs to be connected to the ***Primary*** and POM ***Database*** and not the secondary HOM, SAM, NAM or supplementary SINCAL database.



6.2. Parallel computing setup

The base setup for the non-master machine (2) is identical to the master machine (1), but only requires a copy of the POM model files and associated software. To complete the setup two further steps must be undertaken

- Update Windows Remote Management configuration on both machine (1) and (2)
  - Open DOS command prompt
  - Type: *winrm quickconfig*
  - Select “y” twice
  - AIMMS will need to use more than the default limitation of 150MB memory and this can be increased to e.g. 80%+ of the machines’ available memory in MB
  - Type: *winrm set winrm/config/winrs @{MaxMemoryPerShellMB="XXXXX"}*
- In the main Excel UI – stored on the master machine (1) go to the [Model] -> [Database Manager] screen and complete the AIMMS SETUP boxes for Machine 1 and Machine 2



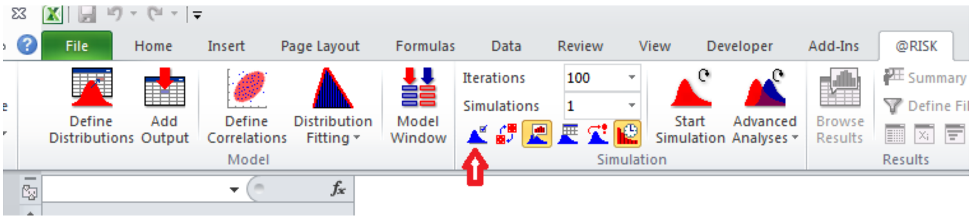
–

These configuration boxes also control the first and last *(consecutive)* simulations which are run on each machine. For example if you wanted to split 100 simulations you would Assign First/Last Sim as 1/50 to machine (1) and 51/100 to machine (2)

6.3. Known software issues

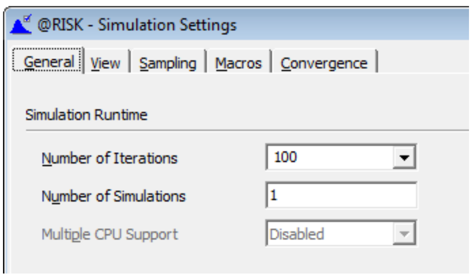
The following are known software issues:

- Multi-processor support in @Risk should be disabled (Nb not all versions of @Risk have this functionality and hence it may be disabled by default). To do this the user should
  - Open the Excel UI and run @Risk from the start menu to load the plugin
  - Select



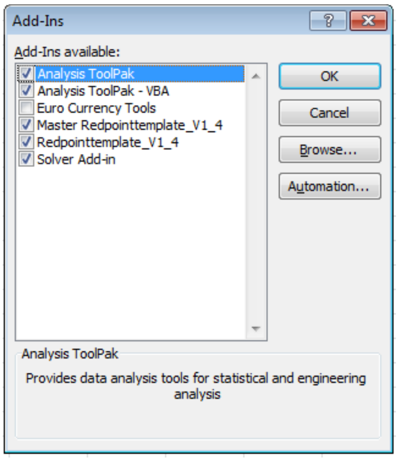
–

  - Ensure “Multiple CPU Support” is set to *Disabled*



–

  - Close and Save the Excel UI
- The Euro-Tool Excel Add-in should be disabled. To do this the user should
  - Open the Excel UI and go to File -> Options -> Add-Ins -> Go



- Uncheck the "Euro Currency Tools" box (if ticked) and select "OK"
- Close and Save the Excel UI

7. Overview of User Interface (UI)

7.1. Main Excel UI

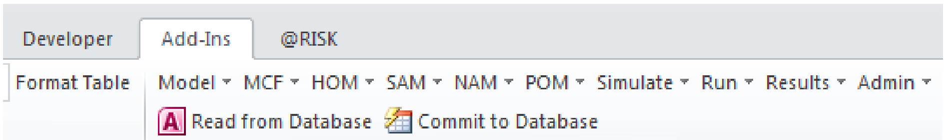
The Excel UI is only designed to facilitate more convenient operation of the tool by an expert user, as it has limited development focus to this point. Its primary roles are to:

- Facilitate the process execution of the tool (including @Risk) by calling the Master Control Framework
  - This includes basic logging information on progress and errors
- Enable the user to add, update, delete information in the primary database without having to access this directly
  - Simple and configurable functionality is included to allow the user to read and write subsets of data to individual tables rather than having to work with the entire table contents at once
- Allow the user to interrogate POM results via a number of predefined results tables and charts, which are updated automatically with data from the primary database

The Excel UI works in conjunction with the ArcGIS UI, as there are a number of user steps (see section 7.2) that are undertaken through this. These ultimately lead to additional or updated information sent to other sub-modules within the tool.

7.1.1. Navigation

The UI navigated via a menu bar within “Add-Ins” tab, containing drop-down options and further sub-menus.

- 

The [Model], [Simulate], [Run], [Results] and [Admin] drop downs and their sub-menu options contain bespoke screens. The other drop downs provide a navigation framework, which re-uses the same ‘***flexible data entry screen***’, but applying different configuration parameters to it (e.g. the choice of table to view) when the sub-menus are selected. These screens are described further in section 8.

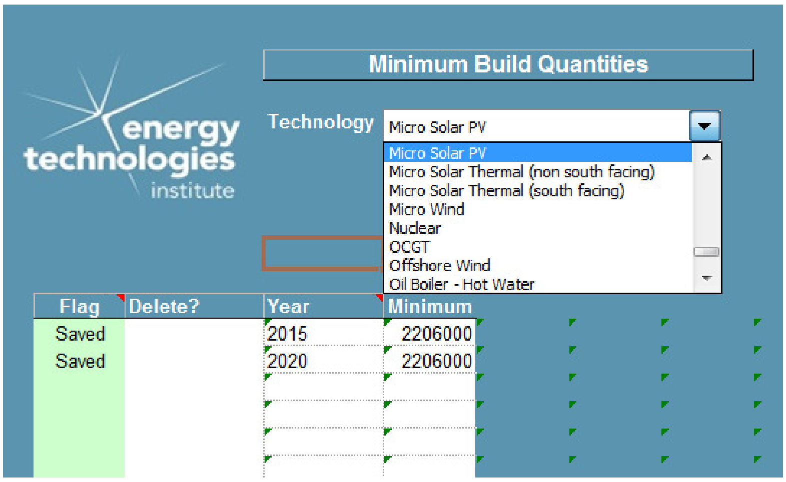
It is generally better to work top to bottom and left to right across the menu items - particularly when adding new elements such as new products - as validation and filtering checks are undertaken by the UI and DB based on these entries.

Additional toolbars are activated upon selection of a new screen depending on the screen type

- **Inputs Toolbar** – activated where data can be added/modified/deleted on an input screen
  - 

7.1.2. GUI formatting conventions

The following cell formatting conventions are used throughout the UI.



–

1. Input cells – data in white cells can be edited by the user
2. Formula cells – grey cells contain formulas
3. Table headings – blue cells with white text show table headings
4. Custom formatting – some cells use conditional formatting to highlight specific information, such as the presence of duplicate entries, which must be corrected before the data is allowed to be written to the Database
5. Drop down (combo) boxes
   - The user can pick from a pre-defined list of entries
   - The data displayed on the worksheet will be filtered according to the current drop down box selection

7.1.3. Reading and writing to the database

Upon selecting a new worksheet via the menu bar as described in section 7.1.1, that worksheet will be refreshed automatically with data from the currently active database.

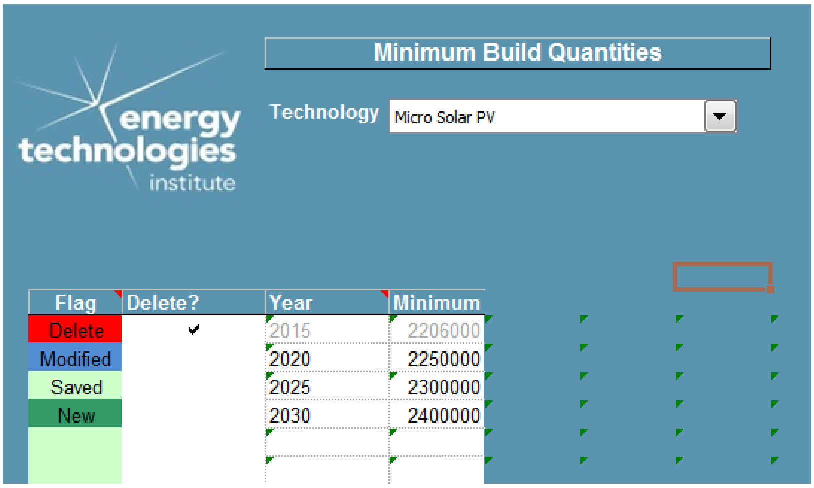
- Any worksheet containing data retrieved from the database can be refreshed by clicking the Read from Database or Refresh Data button from the toolbar just below the main menu.
  - Shortcut: Ctrl + Shift + R
- Equally, modified data can be written to the database by clicking the Commit to Database button

- Upon clicking this button, the user will be prompted to confirm the database write operation via a pop-up window. The pop-up message highlights that that there is no undo option – i.e. any modifications to existing data or the deletion of existing data is permanent.
- If there are errors on the sheet an alternative pop-up window will notify the user that the database write cannot occur until the current errors (as indicated by conditional formatting) have been resolved.
- Shortcut: Ctrl + Shift + C

7.1.4. Adding, modifying and deleting input data

Each input table generally has the following characteristics:

1. A data input range where new data can be added or existing data modified.
   - To modify a saved record, simply edit the existing data in the data input range.
   - A new record can be created by adding data to a blank row at the bottom of the data input table
2. A "Flag" column which highlights whether a record has been added (New), changed (Modified), is for deletion (Delete) or is unchanged from the existing record in the database (Saved).
3. A "Delete?" column which enables the user to select to delete an existing record
   - Double clicking a cell within the Delete column will add/remove a tick mark. When a tick mark is present that record will be formatted with a grey font and the corresponding cell in the Flag column will show "Delete"



–

After entering new data, modifying existing data or marking existing data for deletion the user must click on Commit to Database for these changes to take effect.

7.2. ArcGIS UI

The EPN tool involves a number of tasks and inputs related to the Spatial Analysis Module (SAM), for which the user needs to interact with ArcGIS:

Core setup of underlying data

A. Load **Ordnance Survey** (OS) data into ArcGIS
B. Add **new build** information to OS data (individually or en-masse)
C. Create a **network dataset** using road information (ITN) from OS

Modification of 'unit-level' energy system data

D. Identify **demolition** of existing building stock, **existing PV** uptake and uptake of **electric vehicles** (over time)
E. Define polygon packages for **non-building, non-network energy system features**
F. Define other known **building attributes** (individually or en-masse and including grid-level connections for non-domestic buildings)
G. Define existing electricity / heat network features

Definition and modification of 'cluster-level' energy system data

H. Create **clusters** (from HVS level downstream and more disaggregated), assign cluster **attributes and constraint classes** (represent building archetype, network and technology/storage packages constraints)
   - This includes multiple scenario cluster definitions
I. Define **inter cluster connections** (heat/gas), and potential **meshing options** (electricity)
   - This includes multiple scenario cluster definitions

Viewing Spatial Results

J. View **spatial results** from POM

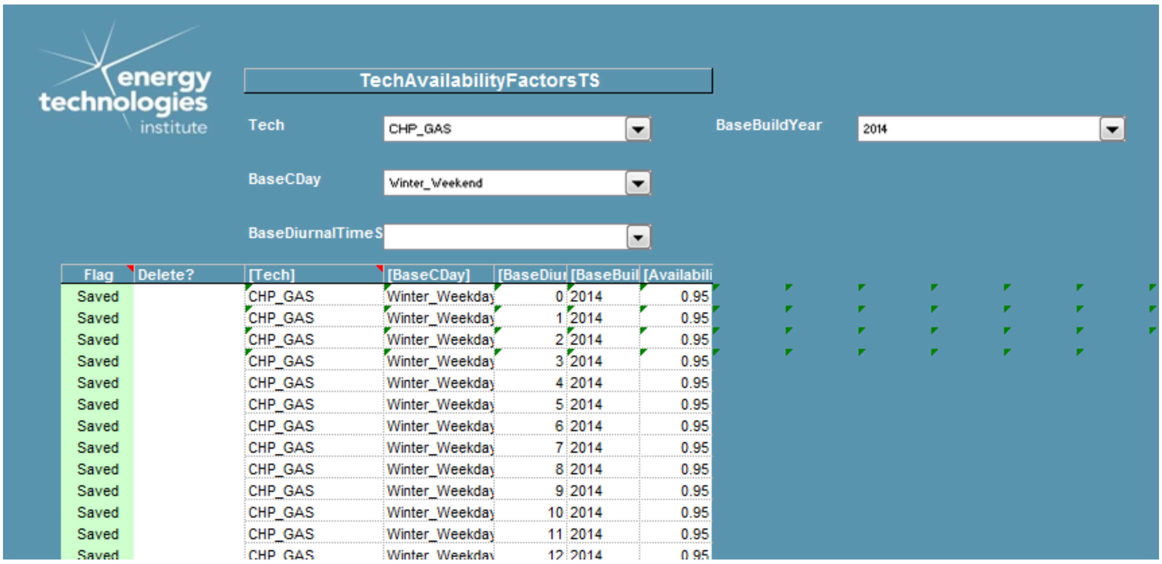
Steps A. to C. will effectively only need to be undertaken once for a given area, whilst the remaining steps could be subject to revisions to test different sensitivities or scenarios. Where these steps fit into the overall tool process flow is shown in section 4. They are in sequential order for the full pass through the tool and are grouped into steps which would be undertaken at the same time in the process flow.

Each individual ArcGIS process step is described in more detail under the sample use cases in section 9, however, it is assumed that the user already has a basic familiarity with the ArcGIS UI, e.g. in terms of standard approaches for editing layers.

8. Overview of specific Excel UI screens

8.1. Flexible data entry screen [MCF to POM menu items]

The main data entry screen is set up to be able to interrogate any si_ (set) or pi_ (parameter) tables - i.e. those that require direct user inputs. It is not intended to access other primary DB tables used for other purposes (e.g. pc_ tables which contain intermediate calculated data being passed from one module database to another)[28].



For pi_ tables the user can, via the *[Parameter Data Config]* menu screen define a number of possible filters, based on the sets relevant to that table, which can be used within the main data entry screen to read and write a subset of the pi_ table only.

For example, the pi_TechAvailabilityFactorsTS table is comprised of >10^5 rows, but it is defined by a number of pieces of si_ set information related to:

- Tech (i.e. technology name]
- BaseCDay (i.e. Characteristic Day)
- BaseBuildYear (i.e. to reflect different technology vintages)
- BaseDiurnalTimeslices (i.e. half hours of the day)

In this example the user has configured filters for the first 3 of these and is applying them with si_ values set for CHP_GAS/Winter_Weekend/2014 respectively, to pull back and adjust a subset of the data in pi_TechAvailabilityFactorsTS. As no value has been set for BaseDiurnalTimeslices filter all half hours are returned for entries which match the other filtered criteria.

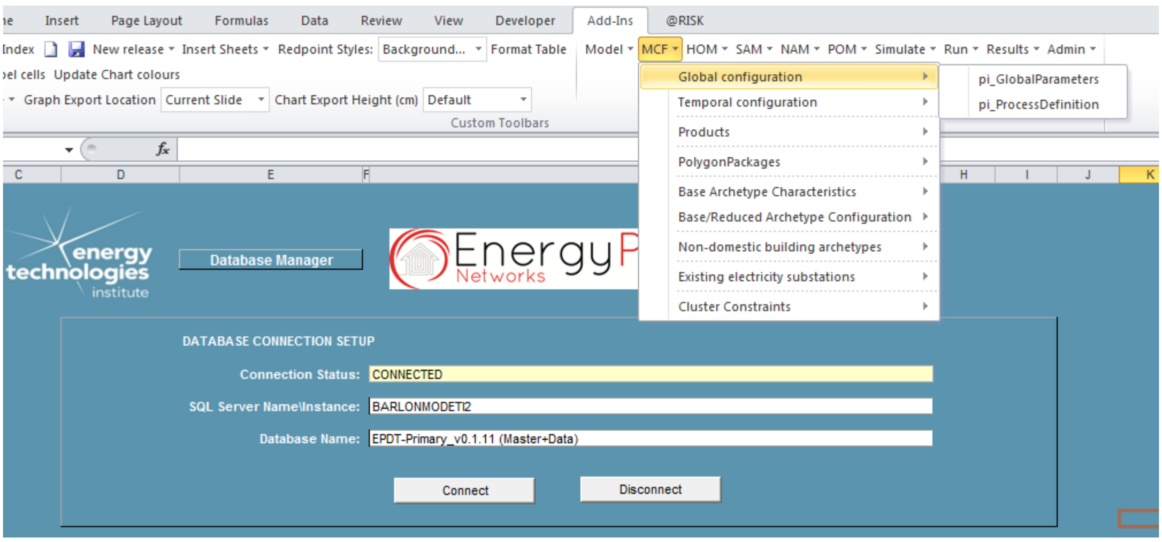
[28] It is also configured to view (but not edit) a number of (vi_) information views, which are used to summarise data from a number of tables in a more convenient format.

Once the user has updated any of the values, the "Commit to Database" button only updates the subset of data selected here, i.e. all other pre-existing data in the table remains unchanged.

MCF to POM menu bar

The sub-menu ordering is designed to reflect the most appropriate way to enter new data to given consistency across the tool (i.e. you cannot define the costs for building new Technology Z, until you have first created new Technology Z) .

When adding new elements to the model (as opposed to modifying values associated with existing elements) the user should work from top-to-bottom, left-to-right from the MCF to the POM menus.



*It is important to note that all sub-menu options moving from the MCF button through to the end of the POM sub-menus buttons are re-using the **same** underlying flexible data entry screen described above. However, depending on the sub-menu option selected the data entry screen options are reconfigured accordingly (e.g. the drop-down filters are updated to reflect those of the specific data table selected by the sub-menu).*

8.2. [Model]

8.2.1. [Database Manager]

This is the same screen as that shown in section 6.1.10 and is used to connect the Excel UI to the relevant SQL server instance and primary database.

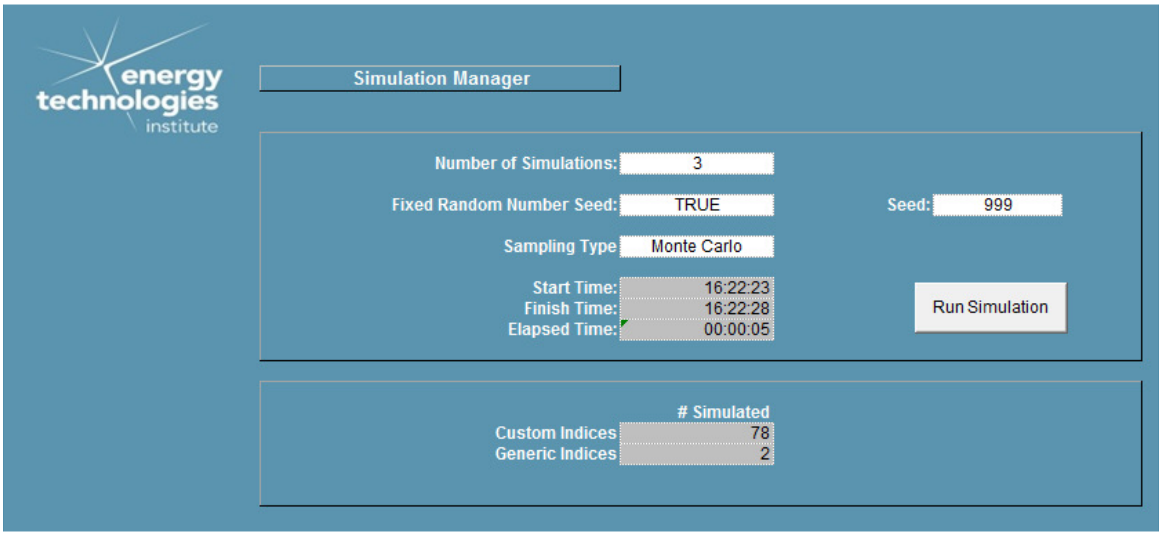
8.3. [Simulate]

8.3.1. [Monte_Carlo – Set/Parameters and associated sub-menus]

These re-use the flexible data entry screen described in section 4.1 and allow the user to enter/delete/modify the input data relevant to the Monte Carlo parameters.

8.3.2. [Simulation Manager]

This screen allows the user to configure and execute the setup parameters associated with generating the Monte Carlo inputs. The “Run Simulation” button simply generates the simulated inputs, which are used by other parts of the tool (as controlled by the process execution flow under the ***[Compute]*** screen).

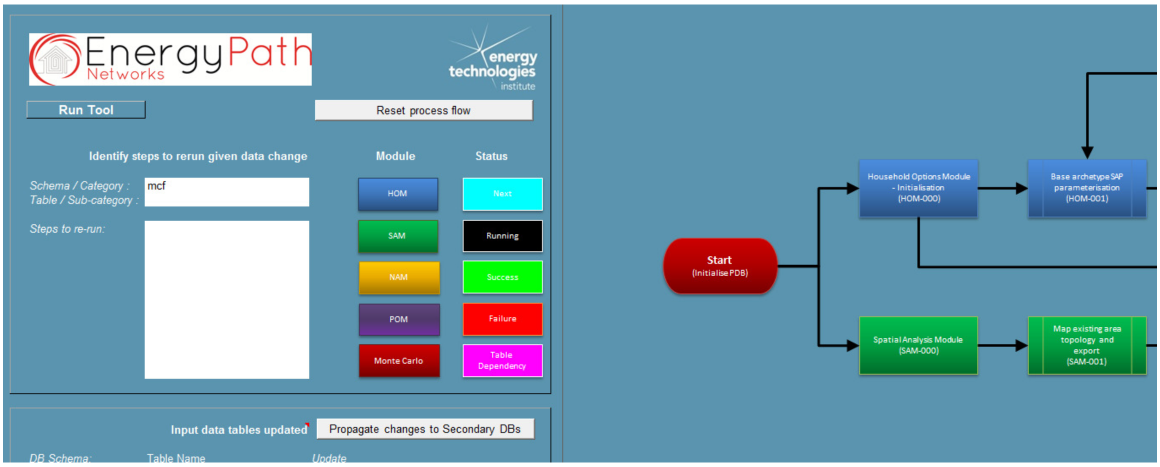


8.4. [Run]

8.4.1. [Process Flow]

This screen contains an interactive process flow which allows the user to:

1. Remotely execute different parts of the tool by clicking on the process flow boxes (including batch processes related to multi-scenario data configuration) and understand the ordering of sub-module process steps and associated dependencies
2. Understand which process steps need to be re-run if they change given input data (and simplify the process

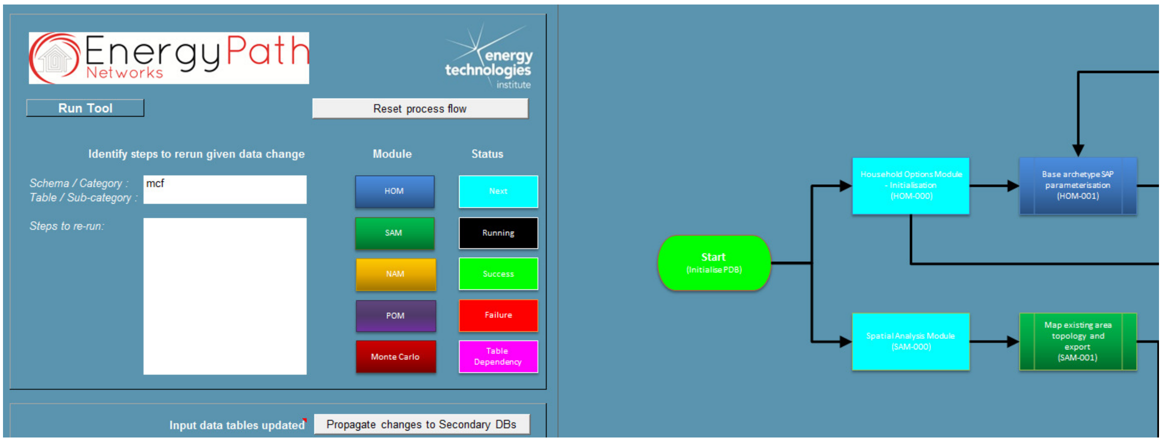


Each of these areas in described in more detail below

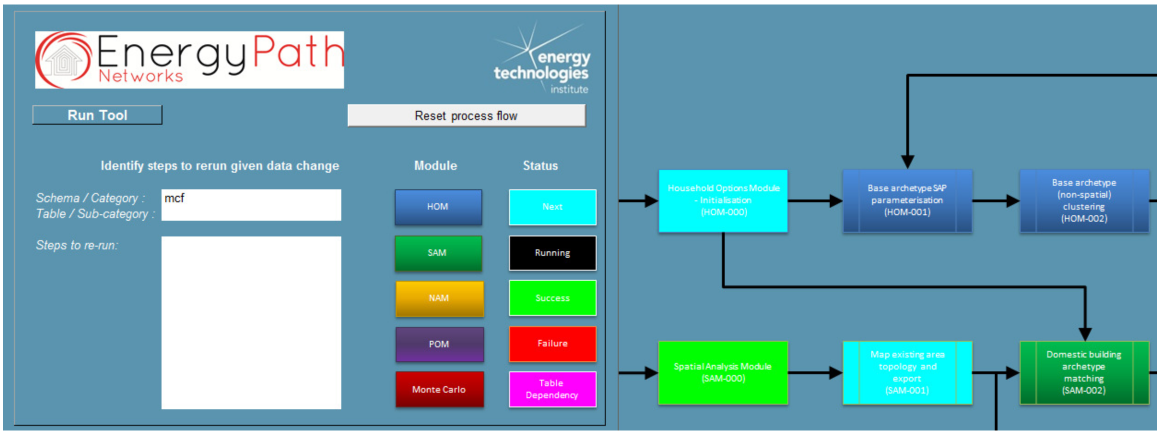
Running the tool via the interactive process flow diagram

The interactive process flow diagram in the UI screen mirrors that in section Figure 4-1. Clicking each box in the process flow diagram remotely executes the relevant sub-module APIs in the EPN tool. The initial colour coding of the boxes highlights which part of the tool each sub-module relates to; as the user works their way through the process flow, selecting each box, the colour coding updates to reflect their status.

For example, after selecting the “Start (initalise PDB)” box the colour will change to **black** to indicate that the process is running. If this process successfully completes it will change to **green** and also highlight the subsequent steps which can then be run in **blue**.

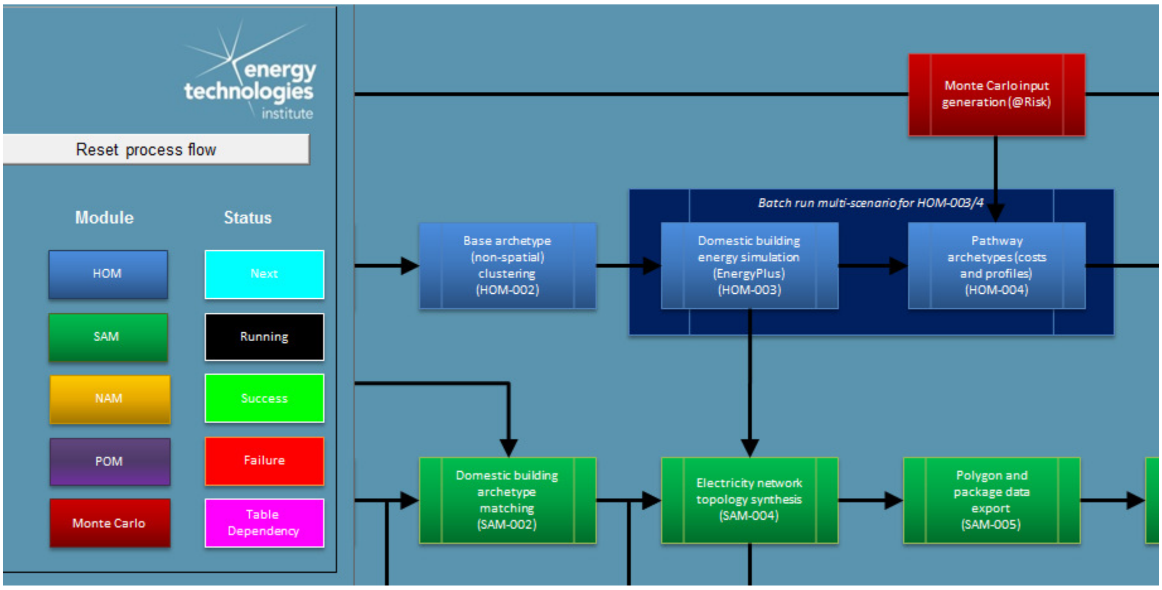


Multiple **blue** next steps indicate that these can be undertaken in parallel, up to the point where there are multiple dependencies. In the example below it is now assumed that the user has successfully run SAM-000. However, before SAM-002 can be run both SAM-001 and HOM-000 must successfully be completed. If the user tries to run SAM-002 before all relevant preceding steps have successfully completed a warning message will be shown.



Where a step fails to complete successfully the user will need to investigate the causes of this (e.g. looking at the EPN tool log files for information) and correct the issue (e.g. data inconsistencies) before re-running the step.

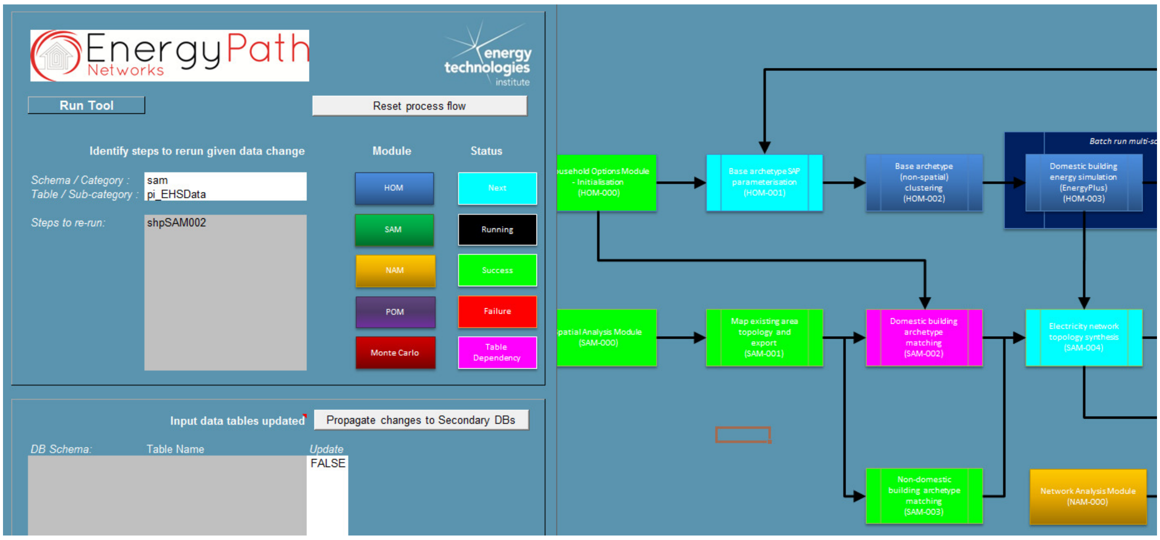
Some process steps are bounded by a wider process box to indicate that these process steps can also be set to run automatically in a batch across data for multiple scenarios that the user has configured. For example, by selecting the outer dark blue box shown below ("*Batch run multi-scenario for HOM-003/4),* rather than the individual process flow boxes, will automatically run both HOM-003 and HOM-004 for all scenarios, as opposed to the "default scenario" data which is used throughout the basic process flow.



Re-running process steps after a data change

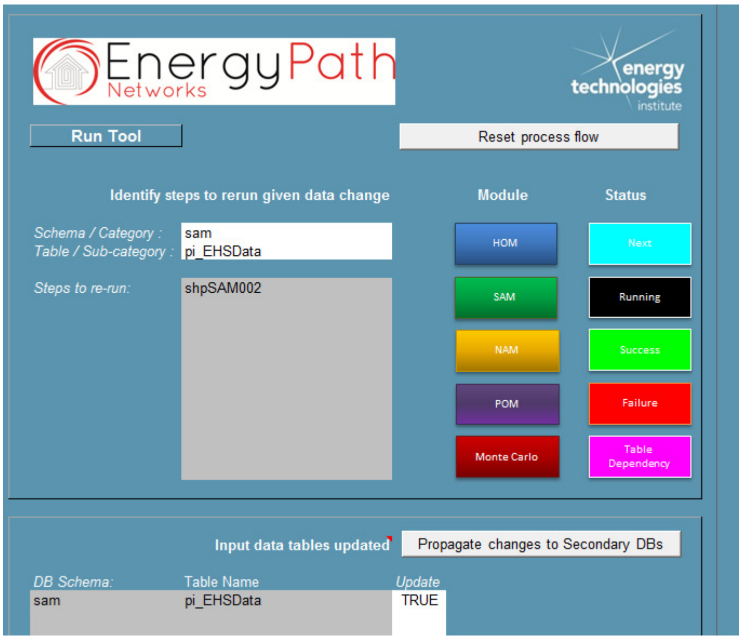
The left hand side of the screen allows the user to select a particular data input and understand where they need to go back to in the process flow; to re-run the required steps to ensure the new data is included correctly in tool.

In the example below, the user has successfully run all steps up to those in **blue** (HOM-001 / SAM-004). They now want to update the EHS (English Housing Survey) input data table (*pi_EHSData*) and have selected this from the drop down box on the left-hand side table. The relevant box to be re-run (followed by the relevant downstream steps), due to the input data table dependencies, are then highlighted in **pink** (SAM-002).



The user can make changes to the data tables via other parts of the Excel UI, but it is important to note that these updates are only made to the Primary Database. Steps within the process flow (e.g. the -000 sub-modules) manage the process of pulling (and pushing) data from the Primary to the Secondary database (or vice versa), but these cover all relevant data and not just the data table that has been updated (e.g. *pi_EHSData*). In the example above, restarting at the SAM-000 process steps means that this, SAM-001 and SAM-003 all need to be re-run, as opposed to just SAM-002.

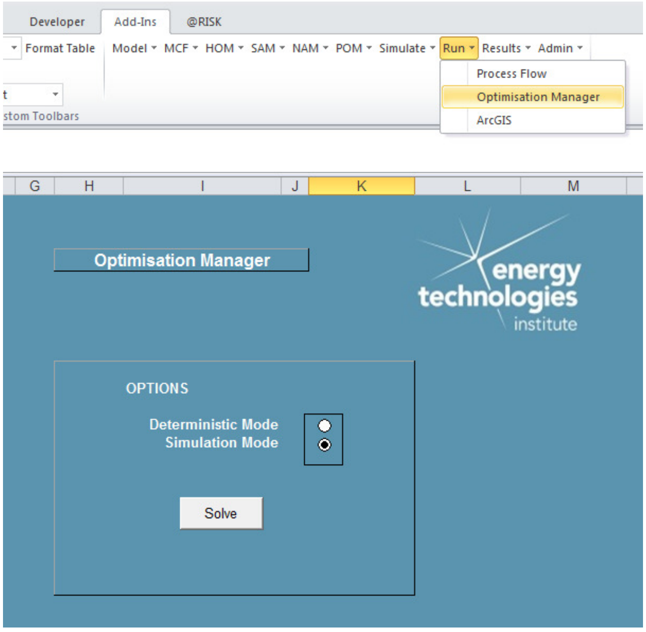
To shortcut the process of only moving the updated input data tables to the relevant Secondary Databases, the user can update the table in the bottom left of the *[Process Flow]* screen and select the "*Propagate changes to secondary DBs*" button.



After this step has finished, the user can then re-run the process step flagged in **pink** and carry on working through the EPN tool process flow as before.

8.4.2. Pathway Optimisation

This opens a separate screen from which the POM AIMMS model can be executed remotely (note that the direct POM AIMMS model file cannot be open at the same time).



Alternatively, once the model process flow has been followed up to this point (i.e. the previous ***[POM-000 Data Pre-Processing and Simplification]***) has completed successfully it is possible to open the POM AIMMS model file directly and run the model via the simple UI interface. An advantage is that the user can then use features within AIMMS such as the Math Programme Inspector (MPI), which can help in resolving any infeasibilities within the optimization problem.

8.4.3. [ArcGIS]

Selecting this menu item opens ArcGIS.

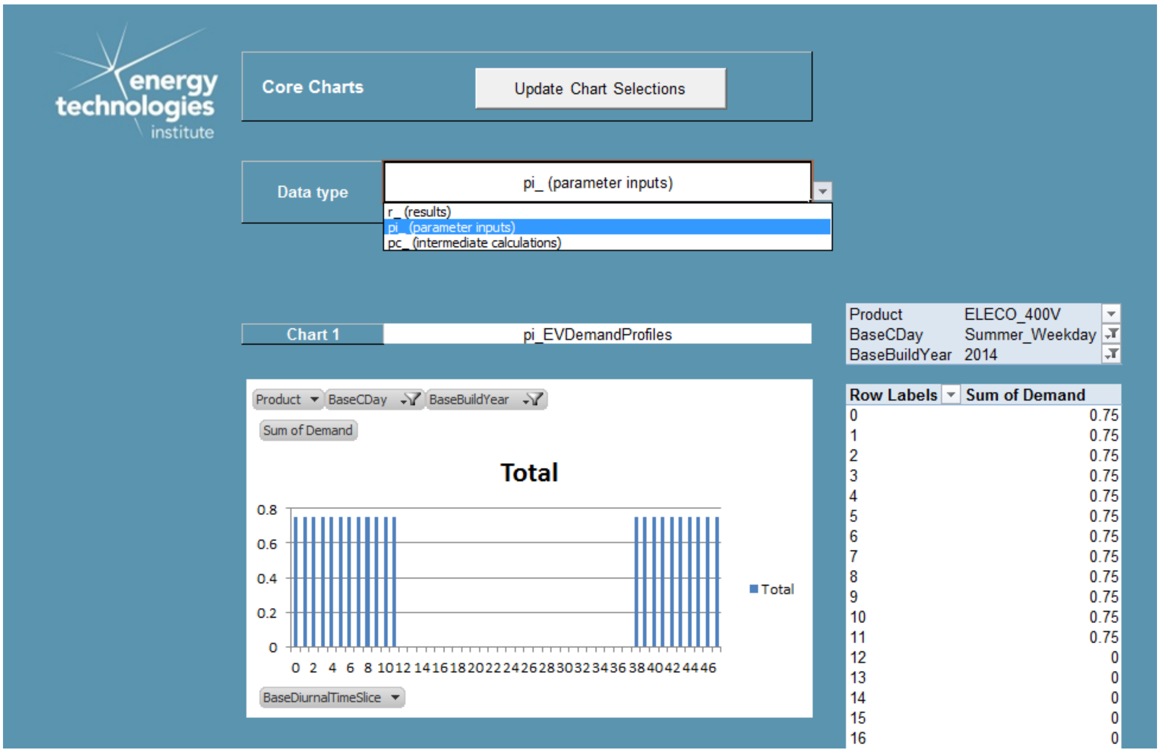
8.5. [Results]

8.5.1. [Core Charts]

This screen allows the user to flexibly configure a number of pre-defined pivot charts (and associated pivot tables) drawing data directly from the primary SQL database.

Via the drop –down next to each chart, the user can select the table of interest and then press -> “Update Chart Selections”. The user can then configure the pivot chart as required. These are saved when the UI is itself saved.

In addition this sheet can be used to visual parameter input data (pi_) or even intermediate calculation (pc_) tables by selecting from the “Data Type” drop down.



8.6. [Admin]

The screens accessed by the [Admin] sub-menu buttons allow the user access to a number of supplementary UI sheets and additional primary DB data. These do not need to be used as part of general operation of the tool and it is strongly recommended that these are only altered by an expert user.

8.6.1. [Hide all] / [Show all]

The buttons simply hide or unhide all other worksheets outside that currently selected by the menus.

8.6.2. [Set data]

This is used to retrieve all active elements in the primary DB si_ and sg_ tables, as these are used to create the filter drop down menus in the flexible data entry screen described in section 8.1.

8.6.3. [Simulation sheets] and associated sub-menus

These are temporary sheets used by the automated Monte Carlo generation process to structure the data into a format suitable for @Risk.

8.6.4. [wksCommandBars]

This sheet is used to configure the Excel menu bar structure in the UI.

8.6.5. [Parameter Data Config]

This sheet is used to configure the available filters for each of the pi_ data parameter input tables viewed via the Flexible Data Entry Screen described in section 8.1.

8.6.6. [Process Config]

This contains the process execution configuration used to control the tool via the ***[Run]*** sub-menu options.

8.6.7. [MCF to Monte_Carlo] and associated sub-menus

These re-use the flexible data entry screen described in section 4.1, to enter/delete/modify a number of additional tables in the primary DB. However, these tables contain key configuration parameters and set information and should only be changed by an expert user.

9. Appendix - Sample use cases

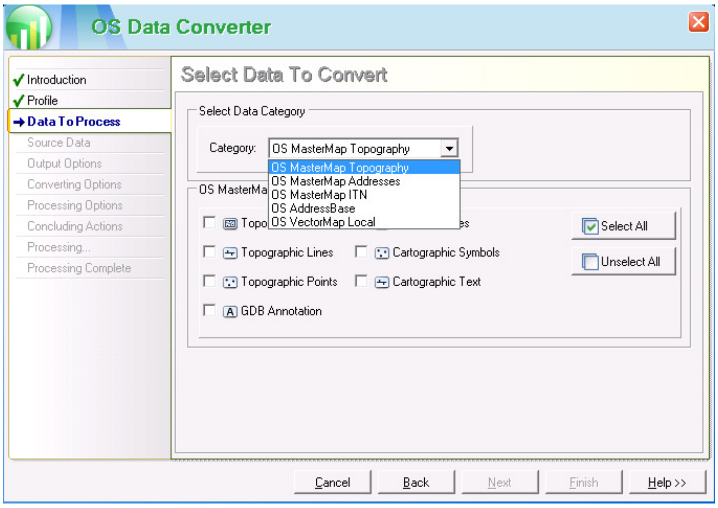
9.1. Relation to process flow

Key user inputs/decision points that relate to the use of the tool (outside of core data input to the database) are flagged in the process flow diagram in 4 via the standard diamond convention. Each of the use cases below can be referenced via the naming convention in the diamonds. For example, each of the key ArcGIS use cases is flagged by the [A], [B], etc, codes.

9.2. ArcGIS [A] Load OS data into ArcGIS

The user is required to load single OS GIS file containing data on buildings, roads and topography. This requires the user to first manually convert the raw OS files (.gz or GML) into the relevant topography, buildings and road file geodatabases (.gdb). This can be undertaken easily using the semi-automated productivity suite OS data converter tool which is designed specifically for this purpose. NB the OS data converter is a standalone tool accessed outside of ArcGIS (via the standard program menu bar in windows) rather than through an ArcGIS menu option.

At the end of the process the user is required to save the output geodatabases (one of the native ArcGIS formats) in a designated folder and with a defined name (i.e. within the core tool folder structure outlined as part of the installation in section 2) . It should be noted that for audit purposes this input data will need to be stored along with the primary SQL database.



To undertake the above process the following steps are required:

- Click on the start button and then open the 'OS Data Converter' software form the 'OS Productivity Suite' folder
- Select the relevant data 'Category' in the step 'Data to Process' and 'Select All' features
- Select the relevant OS files (*.gz or *.gml) in the step 'Source Data'
- Select 'File Geodatabase' from the drop down list as the 'Storage Type' in the step 'Output Options' and name the output file in the designated working ArcGIS folder.

- The data category and the file name to be used are:

Data Category	Output file name
OS MasterMap topography (click on select all)	Topography
OS MasterMap ITN (click on ITN Road Layer)	ITN
OS AddressBase (click on AddressBase Premium')	Building

9.3. ArcGIS [B] Add new build and existing stock information to OS data

The user is required to edit the OS GIS files with data on buildings, road links and road nodes for new build areas or existing buildings.

This requires the user to manually select individual layers (copies of original OS files with unused fields deleted) and edit them to create new points (buildings and road nodes) and lines (road links).

In addition, each of these new entries will require input of basic attributes[29], including:

- Archetype definition for new build defined for:
  - Individual address in Building layer
  - En-masse in StockAttribute layer
- Built year (building)
- Built year (road link - optional)
- Built year (road node - optional)

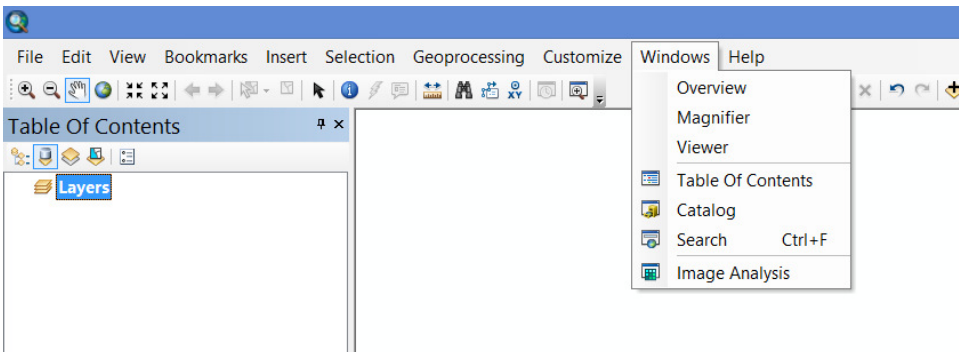
Python script has been developed to auto-populate the following fields relating to unique naming and spatial relationship between new build addresses, new build roads and new build road nodes:

1. UPRN for new build addresses
2. Adjacent road link for new build addresses
3. Road node ID for new road nodes
4. Road link ID for new road links[30]

9.3.1. Steps for editing the OS GIS layers

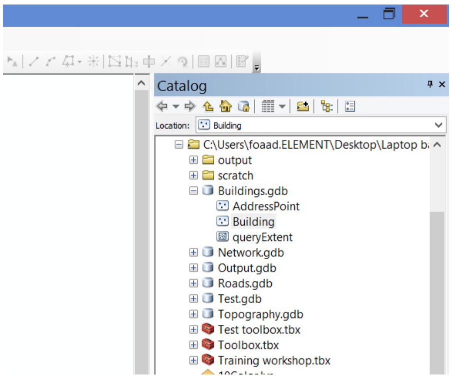
Selecting the appropriate layer

1. Click on the 'Windows' button at the top panel and then select 'Catalog'.

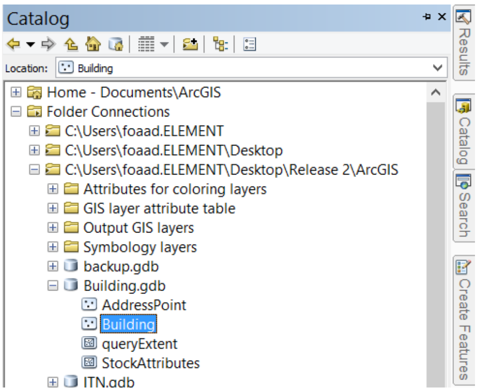


[30] Assigning end road nodes for new road links functionality is available but not used due to license restrictions in ArcGIS basic version

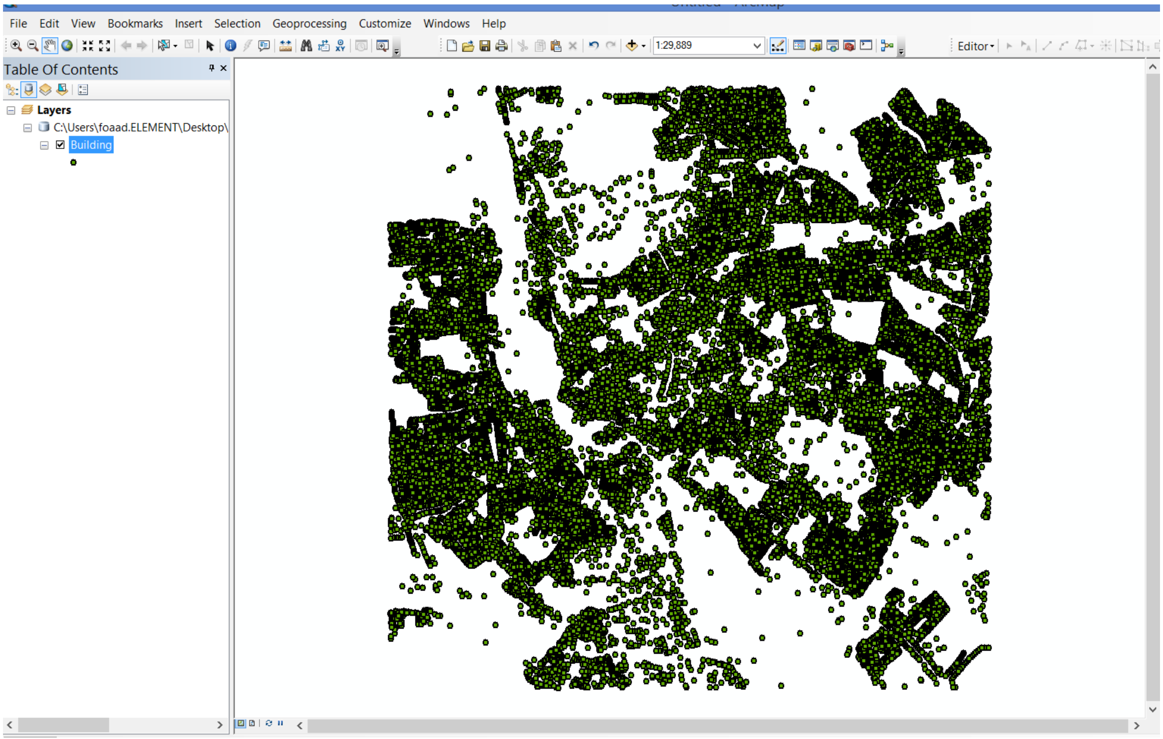
2. This will open up a folder view on the right hand side of the screen.



3. Expand the Building.gdb and select the Building and StockAttributes layer.

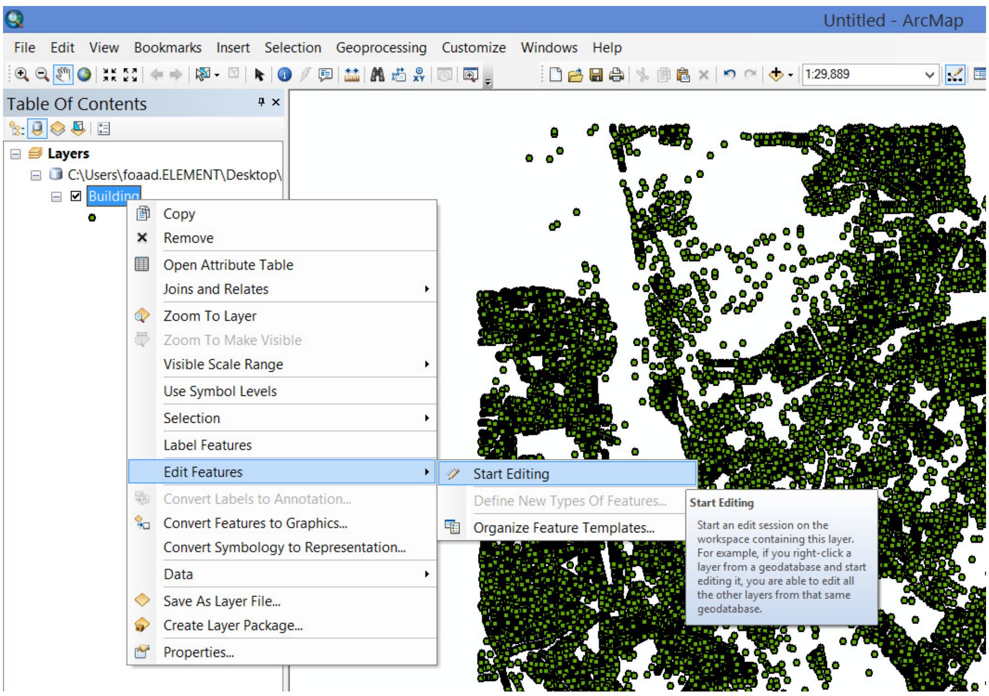


4. Drag this layer into the main view on the left to display the contents of the layer.



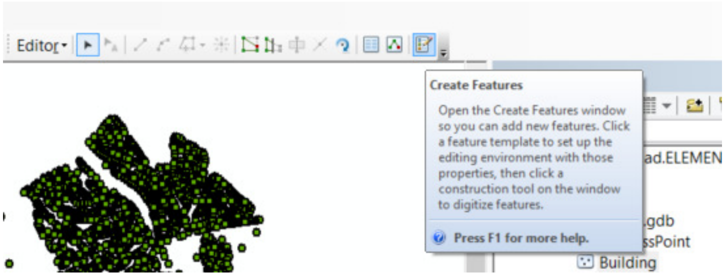
Editing the layer with new build features

1. Right click on the relevant layer and select ‘Edit features’ and ‘Start editing’

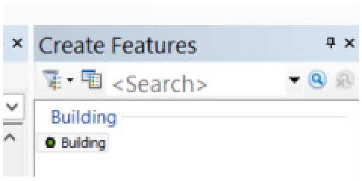


2. This will enable the edit toolbar on the top

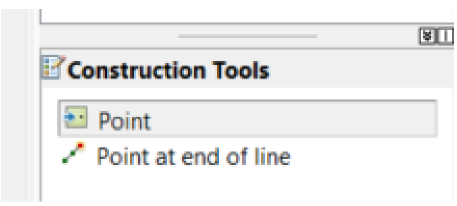
3. Click on the right most icon called 'Create feature'.

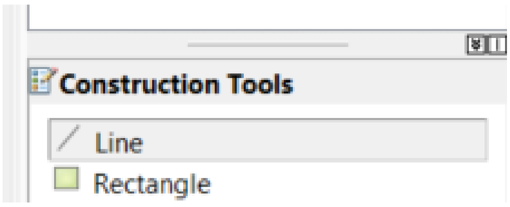


4. This will open the 'Create Feature' tab on the right of the screen, showing the editable layers. Click on the desired layer to edit.

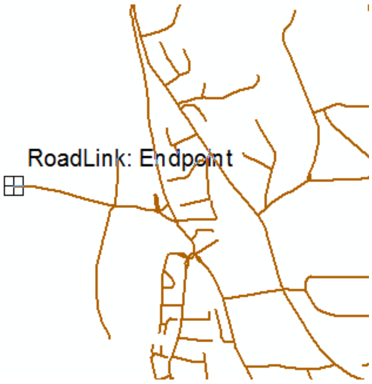


5. The available options for creating new features are shown. For the Buildings and Road Node layer, select 'Point' while for Road Link 'Line' should be selected.

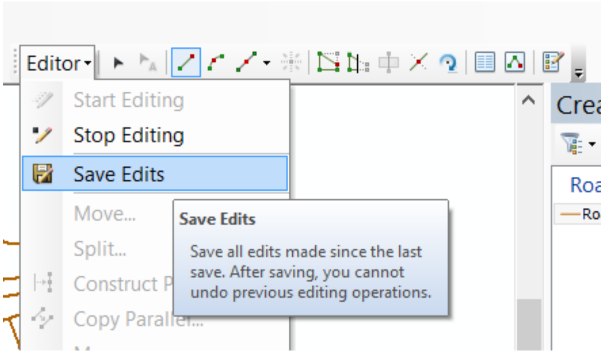




6. After selecting the appropriate construction tool, double click on the appropriate place in the map for creating new nodes (Buildings and RoadNode layers), lines (RoadLink layer) and regions (StockAttributes layer). For RoadLink, it is necessary that the new roads are connected to the existing roads (this can be achieved by hovering over the endpoint of existing road and the 'Line' construction tool locks on to the end point)

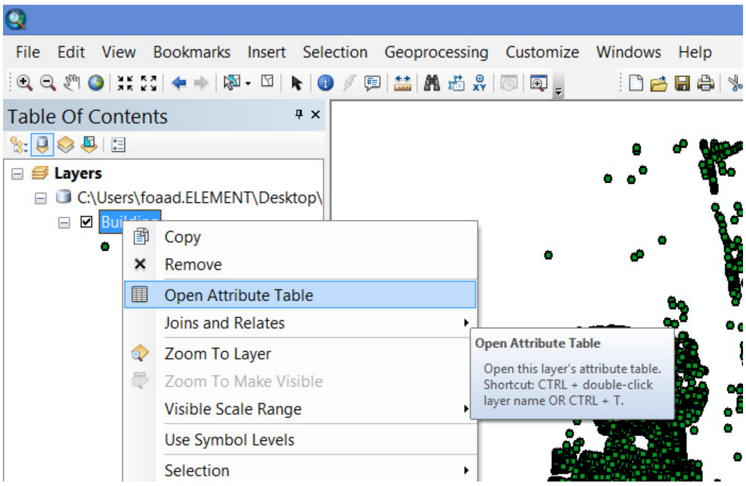


7. After drawing all the nodes / lines for the new build / road node / road links, click on the 'Editor' button and then 'Save edits' on the edit toolbar.



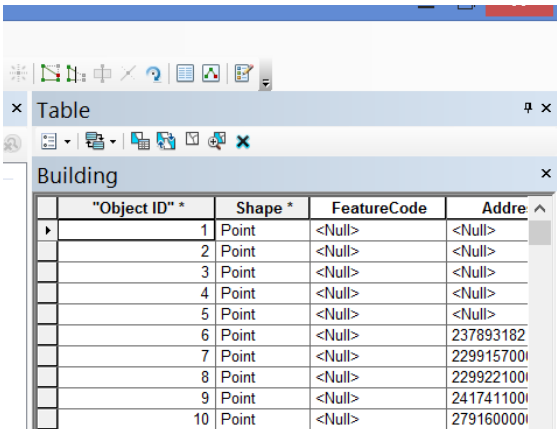
Editing the layer with new build feature data and existing domestic stock attributes at address level

1. Right click on the layer in the 'Table of contents' section on the right of the screen and then select 'Open Attribute Table'.



2. This displays the attributes of the layer contents on the right of the screen. These attributes can then be defined by clicking on the appropriate cells. The attributes can be

numeric (e.g. UPRN, floor space etc), string or drop down lists (e.g. the building attributes of age, type, fuel etc)



3. The following attributes need to be defined for the GIS layers:

RoadNode layer

a. Toid (optional)
b. Year of construction (optional)

RoadLink layer

a. Toid (optional)
b. LnkLength (optional)
c. Node1
d. Node2
e. Year of construction (optional)

Building layer[31]

Item	Description	Corresponding Primary DB table(s)
a. Uprn	Unique reference ID	n/a
b. RoadLink	Unique reference ID	n/a
c. Year of construction (if new build)	Year that new build is constructed	mcf.si_BaseTimePeriods
d. Age (mandatory for new build, optional for existing stock)	Age category assigned to building. For new build this reflects two new categories with	mcf.si_AgeBands

[31] A number of the building layer entries are populated with drop down lists from the Primary Database. These tables can also contain further information related to the categories in the drop down list, which can be interrogated directly in the DB or via the Excel UI. For example, "*area*" relates to the FloorAreaBand 1-8 assigned to the dwellings. The corresponding table *mcf.si_FloorAreaBands* contains further information on the min/max/mean values assigned to each band.

	improving thermal efficiency HLR1 and HLR2	For R1 the final imputed performance of new builds is captured indirectly in the mcf.si_BaseArchetypesExcHeating table. For R2 this will be updated due to the direct integration of the ETI SAP model
e. Type (mandatory for new build, optional for existing stock)	Physical building type	mcf.si_PropertyTypes
f. Area (mandatory for new build, optional for existing stock)	Floor area band assigned	mcf.si_FloorAreaBands
g. Storeys (mandatory for new build, optional for existing stock)	Number of storeys of building (flat = 1 regardless of number floors in overall property)	mcf.si_NumStoreys
h. Fuel (mandatory for new build, optional for existing stock)	High level fuel product group (e.g. electricity)	mcf.si_ProductGroups
i. Wall (mandatory for new build, optional for existing stock)	Wall type and current insulation, specific category for new build properties	mcf.si_WallTypes
j. Window (mandatory for new build, optional for existing stock)	Window type, specific category for new build properties	mcf.si_WindowTypes
k. Loft (mandatory for new build, optional for existing stock)	Loft type and current insulation, specific category for new build properties	mcf.si_LoftInsulation
l. Ventilation (mandatory for new build, optional for existing stock)	Ventilation type	mcf.si_Ventilation
m. Primary Heating (mandatory for new build, optional for existing stock)	Current primary heating system assigned to property	mcf.si_PrimaryHeatingSystems
n. Secondary Heating (mandatory for new build, optional for existing stock)	Current secondary heating system assigned to property (including "none")	mcf.si_SecondaryHeatingSystems
o. Heating controls (mandatory for new build, optional for existing stock)	Current heating control assigned to property (including "none")	mcf.si_HeatingControls
p. Storage (mandatory for new build, optional for existing stock)	Current storage system assigned to property (including "none")	mcf.si_BuildingStorageTechnologies
q. Tenure (mandatory for new build, optional for existing stock)	Tenure assigned to building	sam.si_Tenures

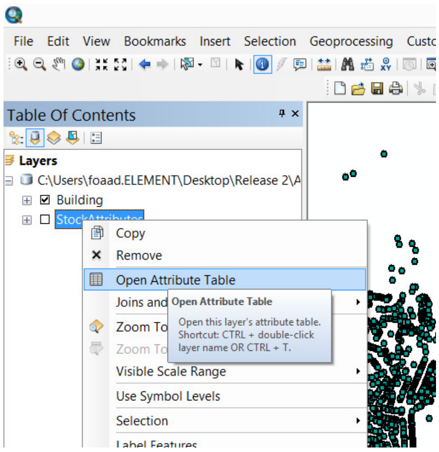
a. Uprn (optional)
b. Description (mandatory for new build)
c. RoadLink (optional)
d. Year of construction (mandatory for new build)
e. Age (mandatory for new build, optional for existing stock)
f. Type (mandatory for new build, optional for existing stock)
g. Area (mandatory for new build, optional for existing stock)
h. Storeys (mandatory for new build, optional for existing stock)
i. Fuel (mandatory for new build, optional for existing stock)
j. Wall (mandatory for new build, optional for existing stock)
k. Window (mandatory for new build, optional for existing stock)

l. Loft (mandatory for new build, optional for existing stock)
m. Ventilation (mandatory for new build, optional for existing stock)
n. Primary Heating (mandatory for new build, optional for existing stock)
o. Secondary Heating (mandatory for new build, optional for existing stock)
p. Heating controls (mandatory for new build, optional for existing stock)
q. Storage (mandatory for new build, optional for existing stock)
r. Tenure (mandatory for new build, optional for existing stock)
s. Floor space (mandatory for new build non-domestic properties, optional for existing stock)
t. Stock (optional)
u. Grid Level (optional)

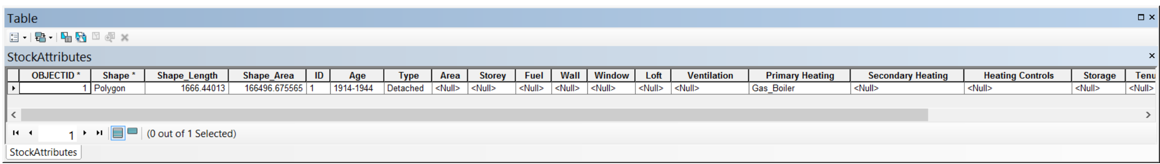
4. It is important to maintain consistency between the layers when the spatial relationship is user defined i.e. the 'Node1' and 'Node2' for RoadLink layer must relate spatially to the RoadNode layer while the 'RoadLink' in the Building layer must relate spatially to the RoadLink layer. If left unfilled, the spatial relationship of adjacent road link is defined by the python script.

Editing the StockAttributes layer for existing domestic stock en-masse

1. Right click on the layer in the 'Table of contents' section on the right of the screen and then select 'Open Attribute Table'.



2. This displays the attributes of the layer contents on the right of the screen. These attributes can then be defined by clicking on the appropriate cells. The attributes can be numeric (ID, stock or Percentage of stock) or drop down lists (e.g. the building attributes of age, type, fuel etc)



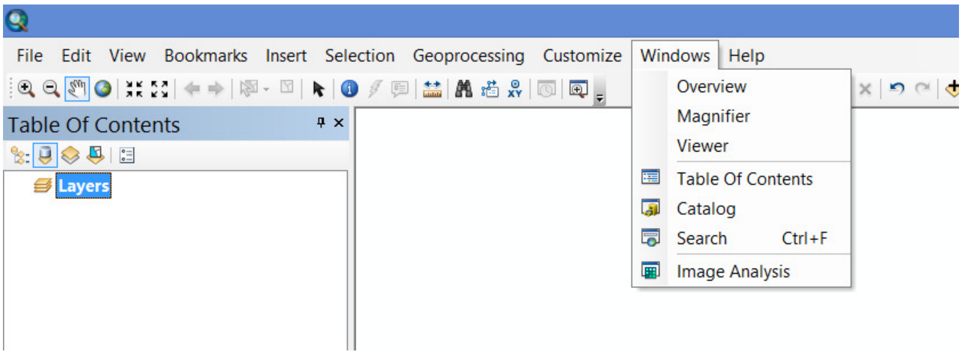
9.4. ArcGIS [C] Create a network dataset using OS road information (ITN)

ArcGIS requires the ITN layer to be converted into network dataset format in order to perform the network synthesis. This requires user to manually generate network dataset from the converted ITN geodatabase (created in the previous step **[B]**).

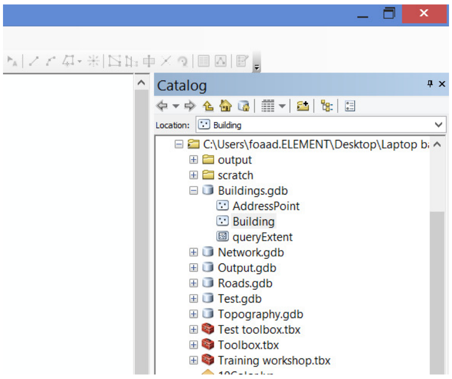
The user will be required to save the output geodatabase containing the network dataset in a designated folder and with a defined name[32].

9.4.1. Steps for creating the Network Dataset

1. Click on the 'Windows' button at the top panel and then select 'Catalog'.

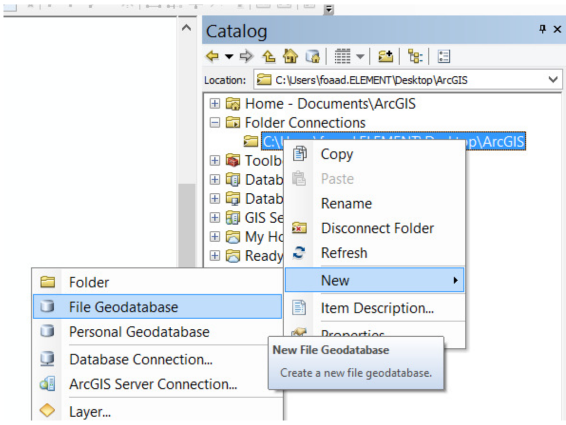


2. This will open up a folder view on the right hand side of the screen.

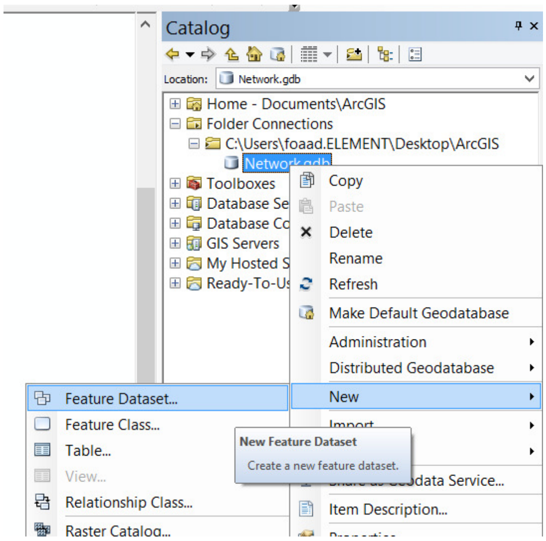


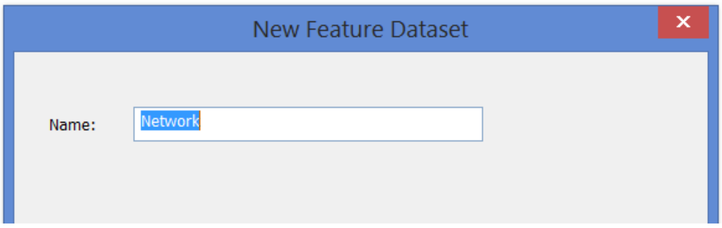
3. Right click on the working ArcGIS folder, select 'New file Geodatabase' and name it as "*Network.gdb*"

[32] Discussions with the ArcGIS developers have indicated that it is not, at present, possible to script these steps automatically through Python, however, the user is only likely to append the existing ITN layer infrequently.

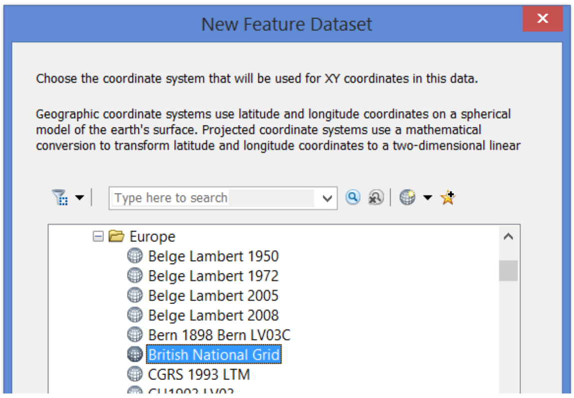


4. Right click on *Network.gdb*, select 'New' and 'Feature Dataset', name it as "*Network*".

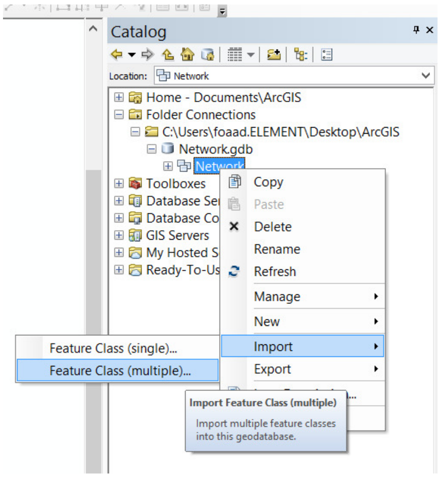




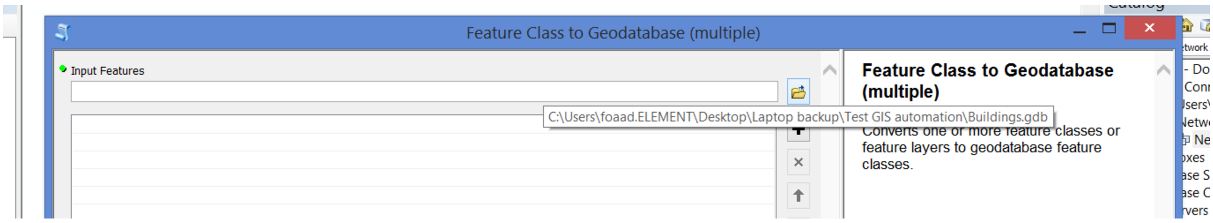
5. Click on 'Next' and select the coordinate system as 'Projected Coordinate Systems' –> 'National Grids' –> 'Europe' –> 'British National Grid'

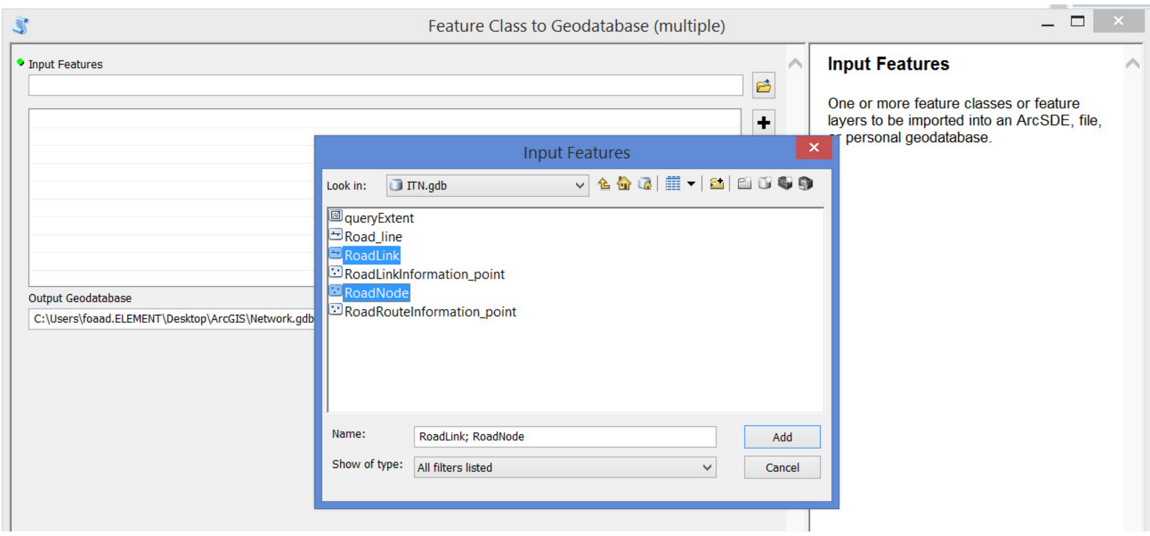


6. Click on next and keep the default settings for remaining steps.
7. Right click on *Network* feature dataset created in the previous step, select ‘Import’ and ‘Feature Class (multiple)’.

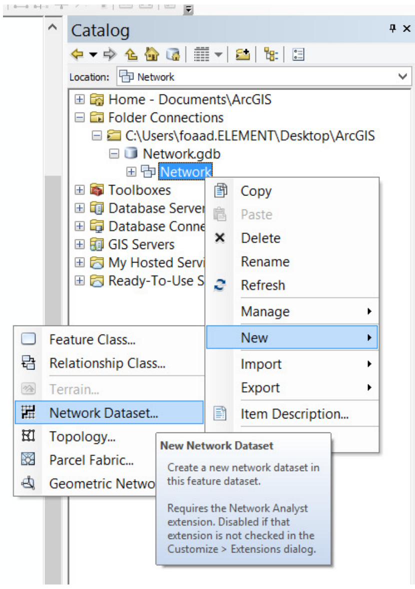


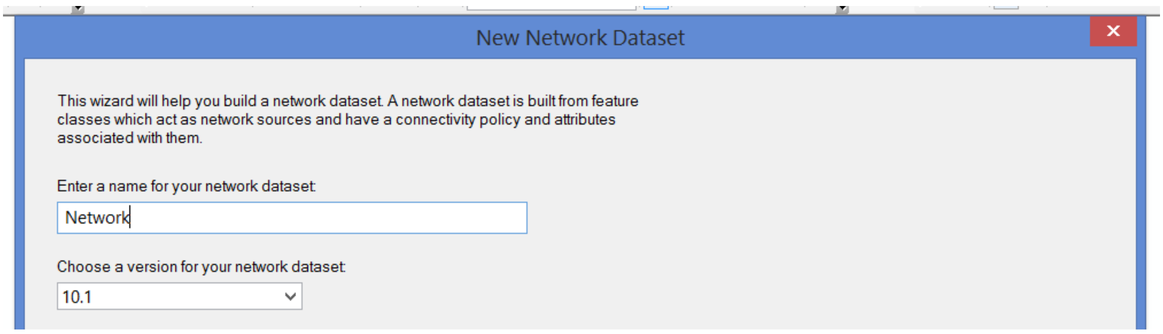
8. Click on the folder icon at the top and select the ‘RoadLink’ and ‘RoadNode’ layers from the ITN.gdb folders.



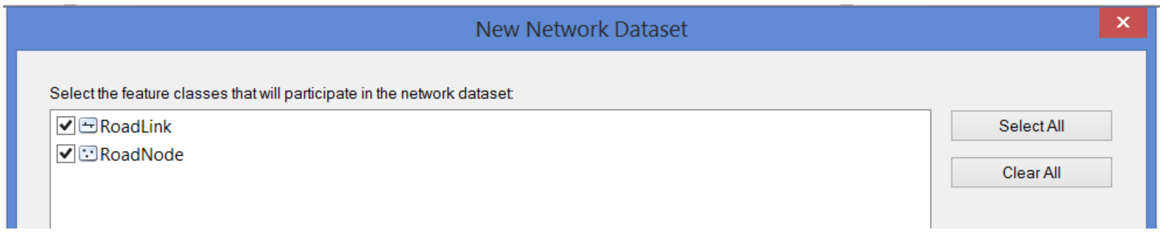


9. Right click on *Network* feature dataset created in the previous step, select 'New' and 'Network Dataset', name it as "*Network*".

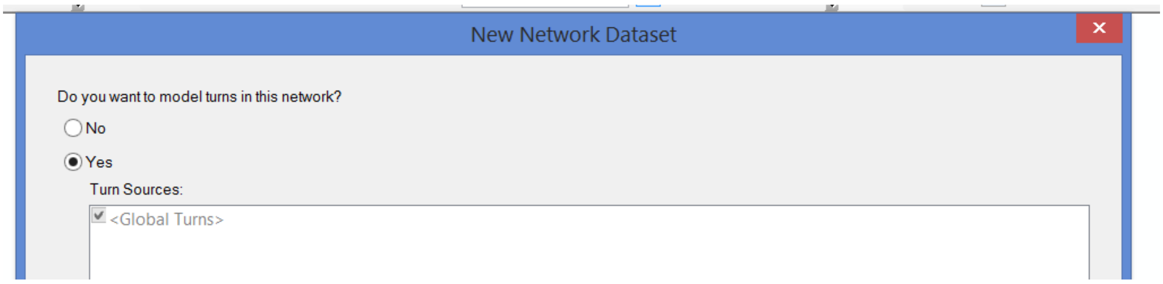


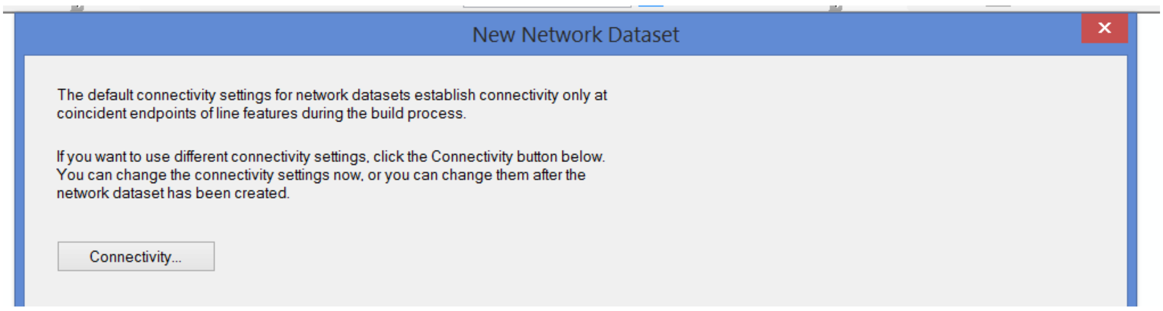


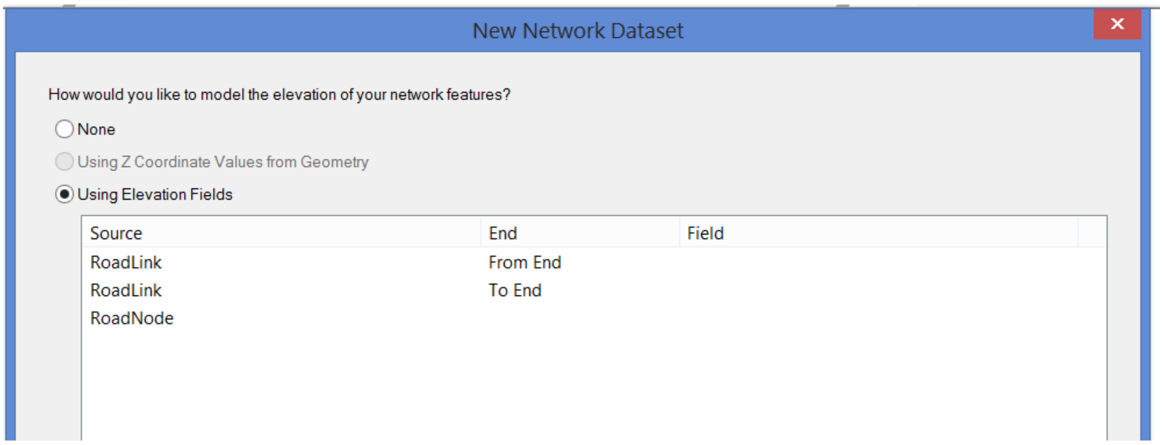
10. Click on 'Next' and then 'Select All' to include both layers in the network dataset.

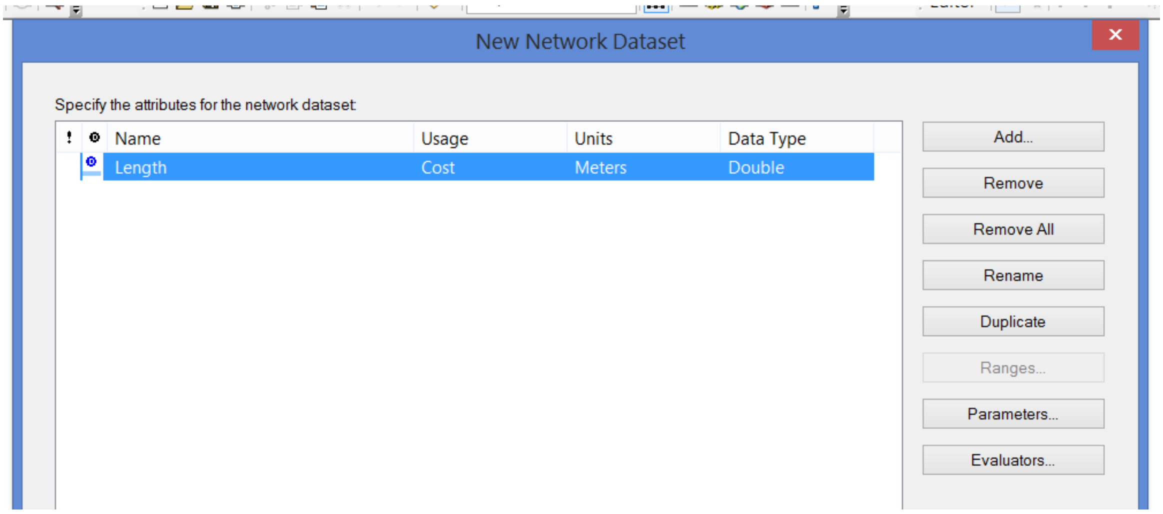


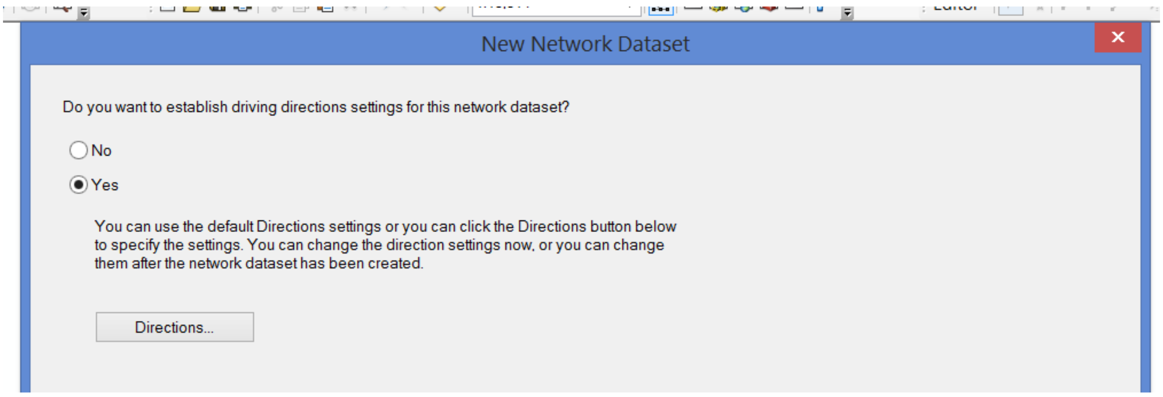
11. Click on 'Next' for the next 5 steps until the 'Directions' button appears.



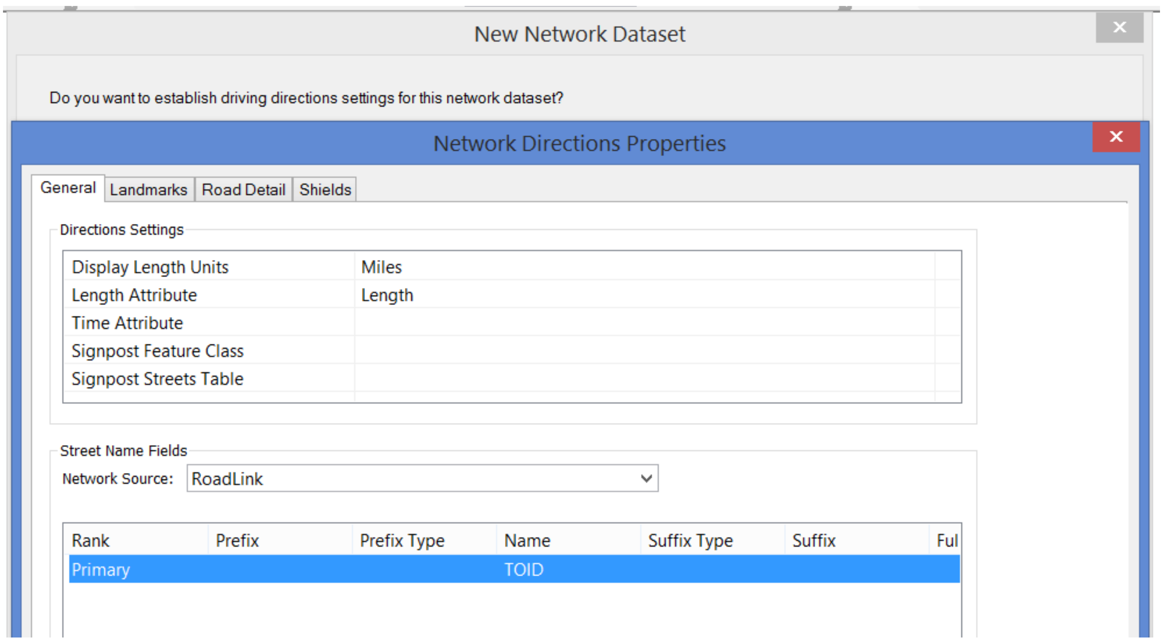




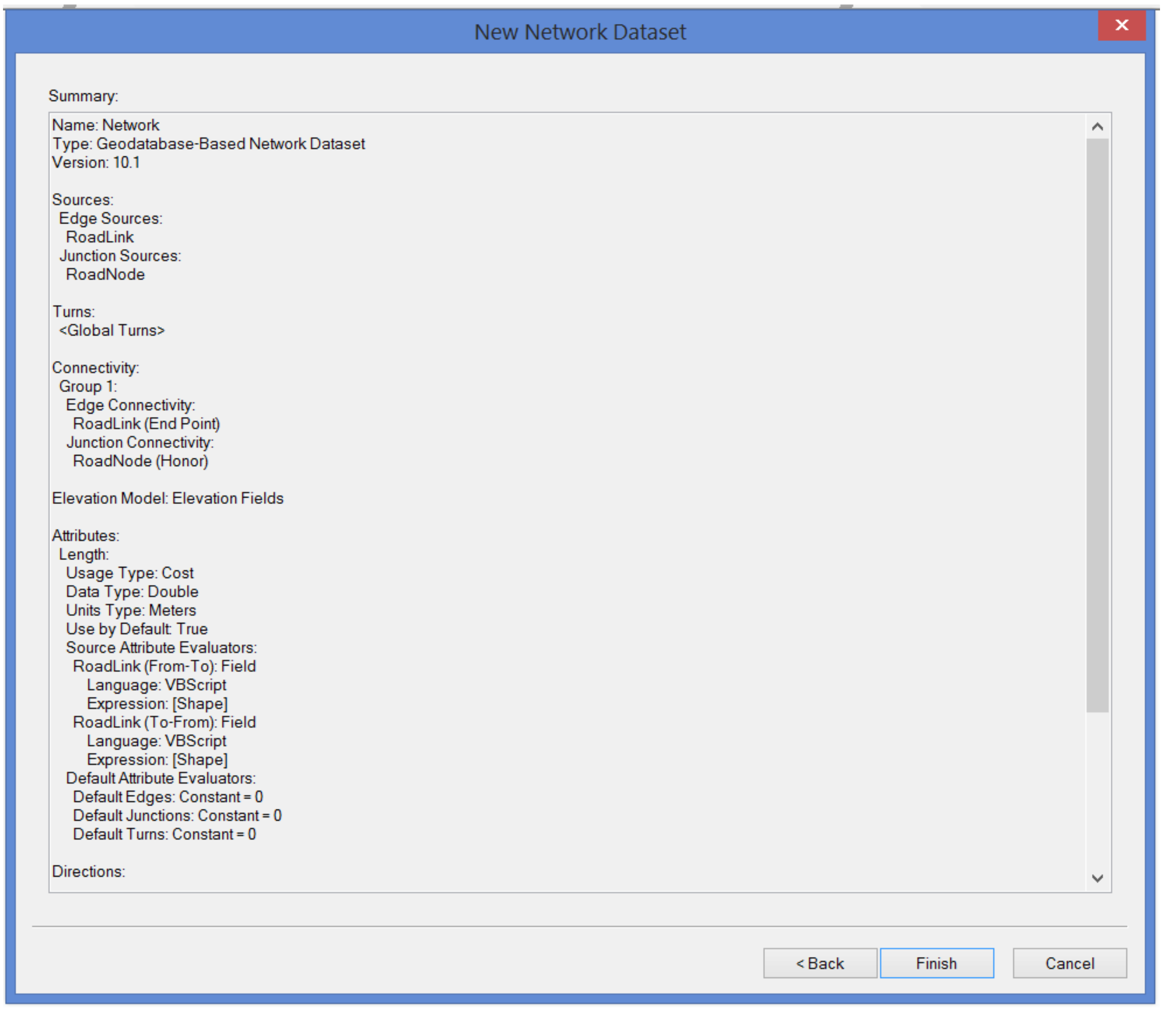


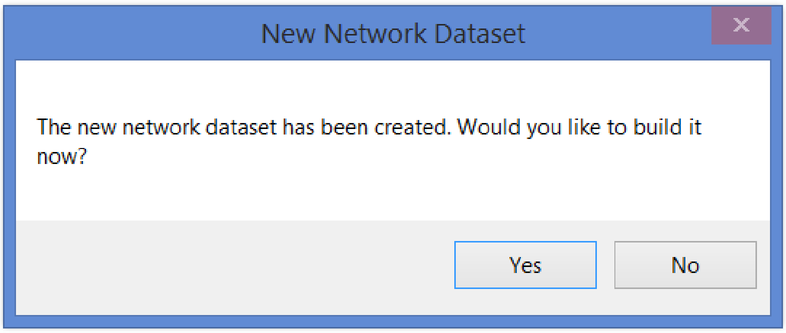


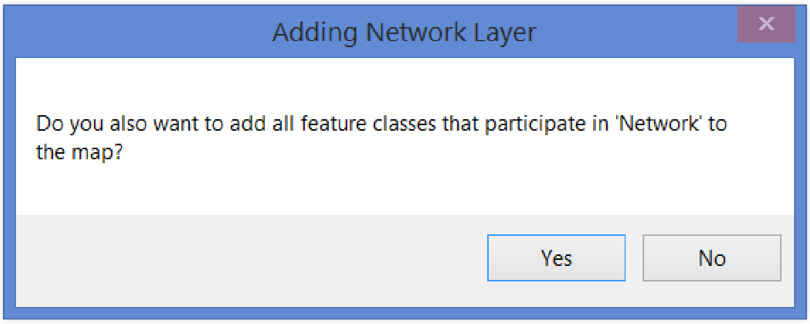
12. Click on the 'Directions' button and select 'TOID' from the drop down list under the header 'Name'.

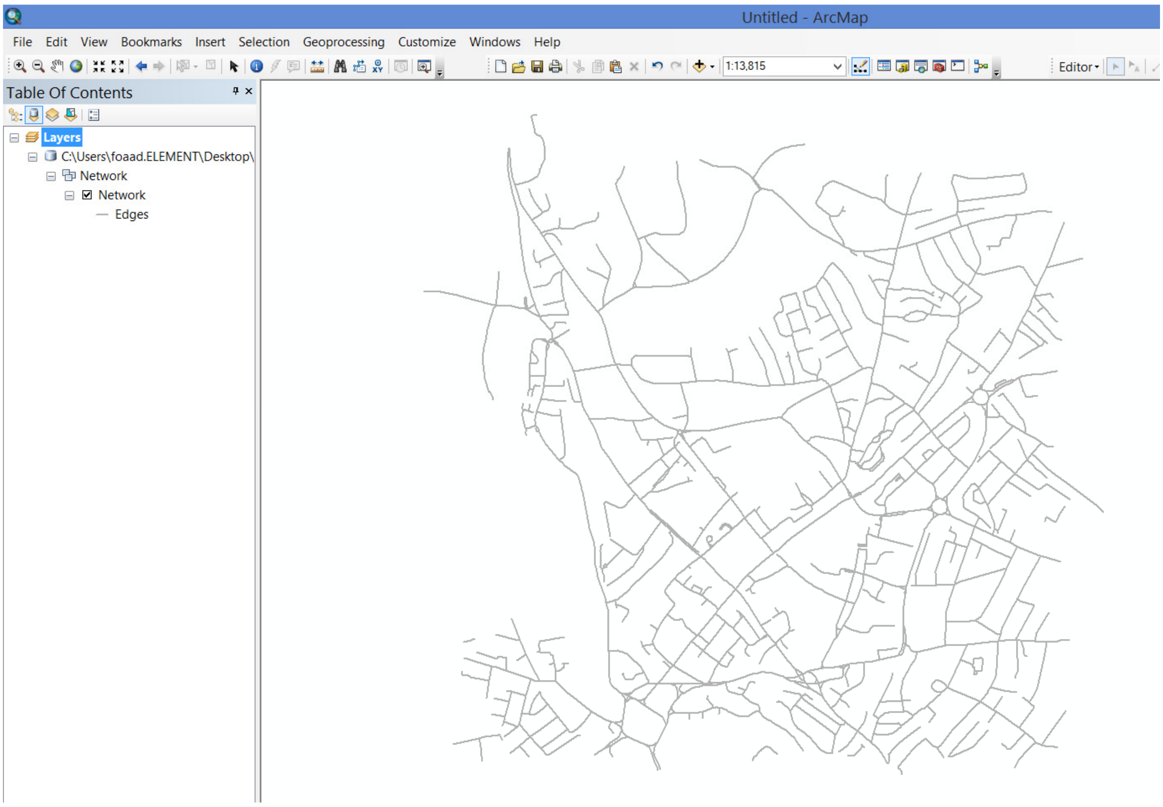


13. Click 'OK', then 'Next' and finally 'Finish' to complete the process of network dataset creation. Finally click on 'Yes to build the network dataset and 'Yes' to include the feature layers to the map.






Adding Network Layer
Do you also want to add all feature classes that participate in 'Network' to the map?
Yes
No

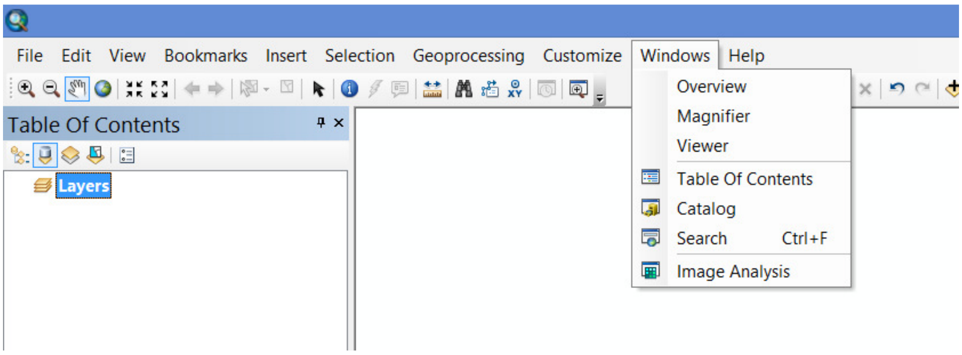

Untitled - ArcMap
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:13,815
Editor
Table Of Contents
Layers
C:\Users\foaad.ELEMENT\Desktop\
Network
Network
Edges

9.5. ArcGIS [D] Identify building demolition, PV and EVs

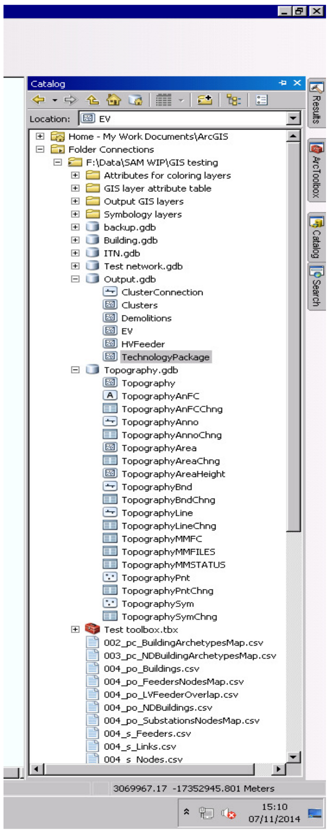
The SAM generates a spatial database of existing building stock based on the OS Addressbase layer. An automated python script creates a feature dataset that allows the user to draw polygons representing areas where demolition will take place, existing stock of installed PV and the stock projections of electric vehicles.

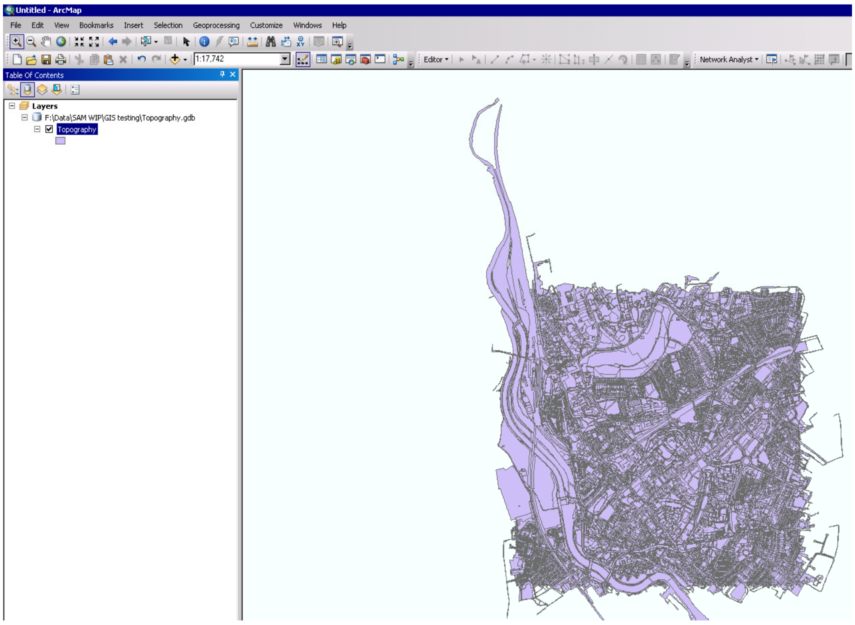
9.5.1. Steps for editing the demolition, PV and EV layers

1. Click on the 'Windows' button at the top panel and then select 'Catalog'.

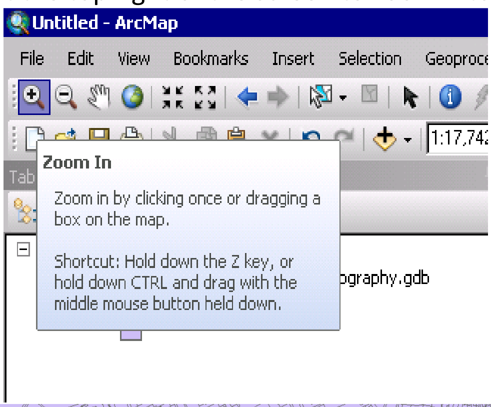


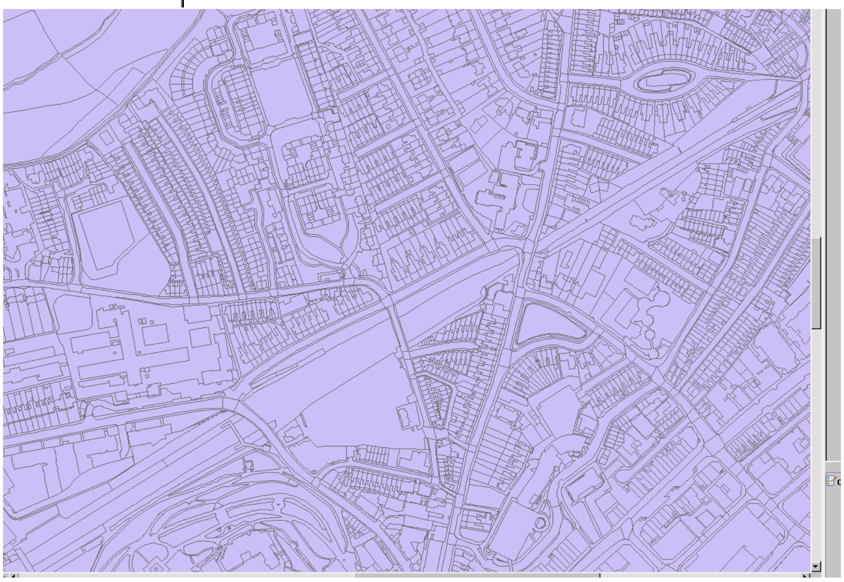
2. This will open up a folder view on the right hand side of the screen.
3. Expand the 'Topography.gdb' and drag the 'Topography' feature layer into the main view on the left.



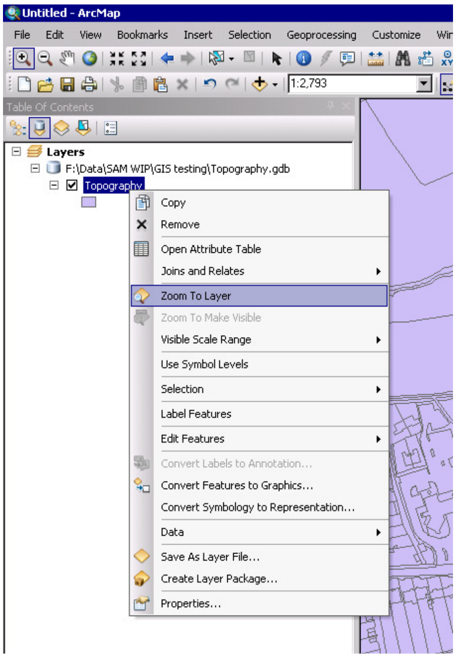


4. Click on the zoom icon at the top right of the screen to zoom into the LA map.

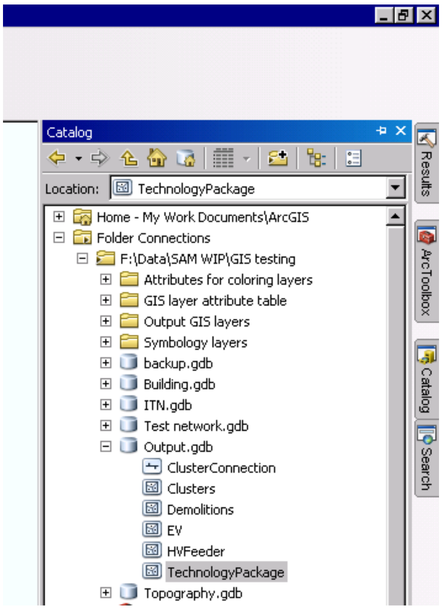




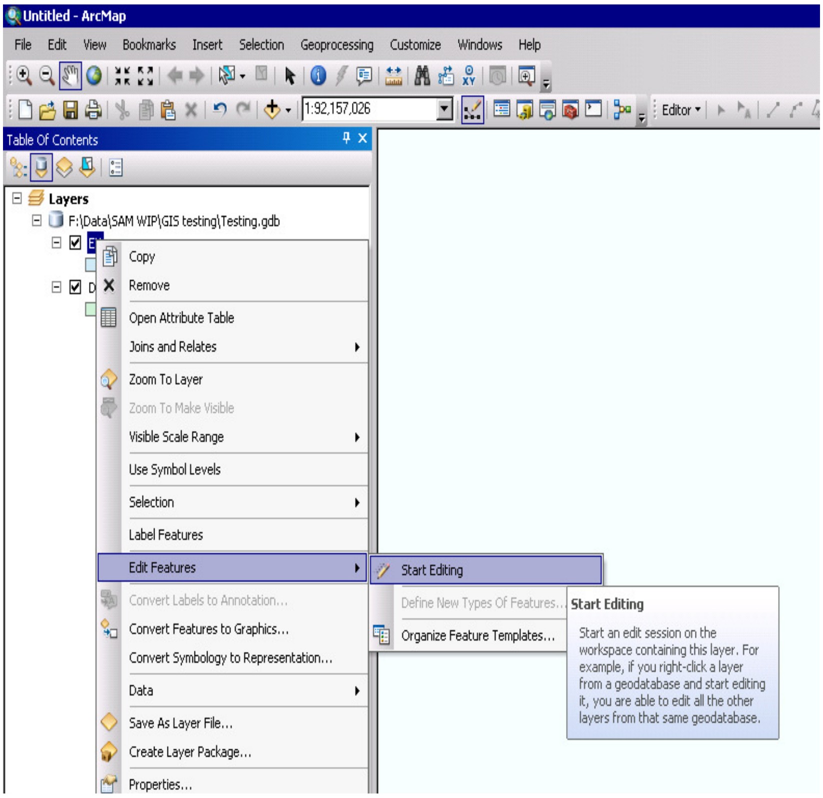
5. To go back to the original view, right click on the 'Topography' layer in the table of contents and select 'Zoom To Layer'.



6. Expand the 'Output.gdb' and drag the 'Demolitions' and 'EV' feature layers into the main view on the left.

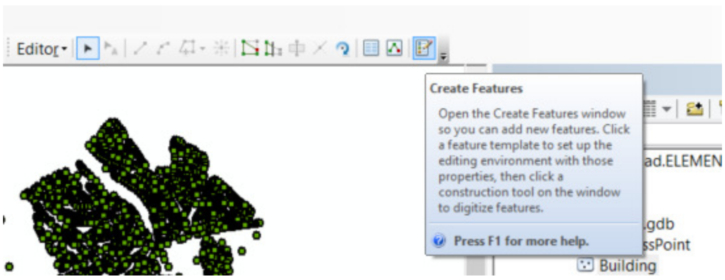


7. Right click on the 'Demolitions' layer and select 'Edit features' and 'Start editing'.

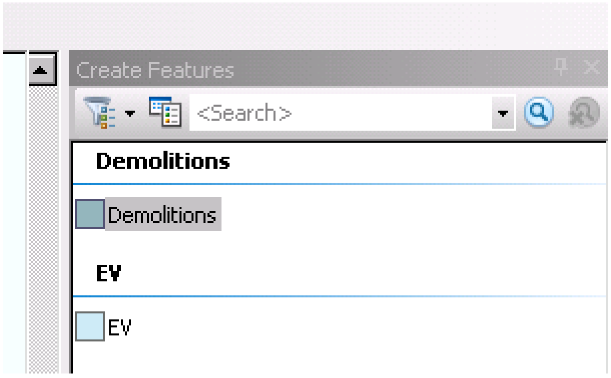


8. This will enable the edit toolbar on the top

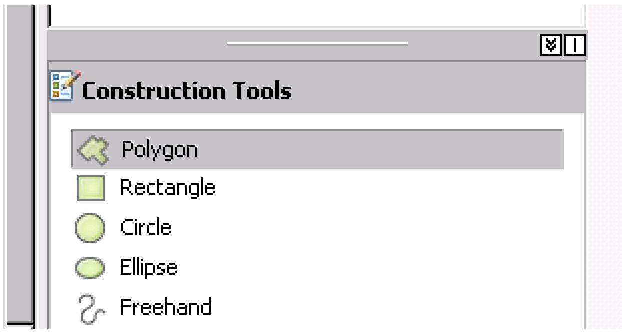
9. Click on the right most icon called ‘Create Features’.



10. This will open the ‘Create Feature’ tab on the right of the screen, showing the editable layers. Click on the ‘Demolitions’ layer to edit.



11. The available options for creating new features are shown. Select the ‘freehand’ option to draw the demolition areas.



12. Press the left click on the map at the desired location and drag to draw the boundary of the demolition region, then double click to complete the boundary.

13. Once all the demolition regions have been drawn, right click on the demolition layer in the ‘Table of contents’ section on the right of the screen and then select ‘Open Attribute Table’.

14. This displays the attributes of the layer contents on the right of the screen. Enter ‘ID’, ‘Year of demolition’ and either ‘Total Number of Building Stock Demolished’ or ‘Percentage of Building Stock Demolished’ for each area.

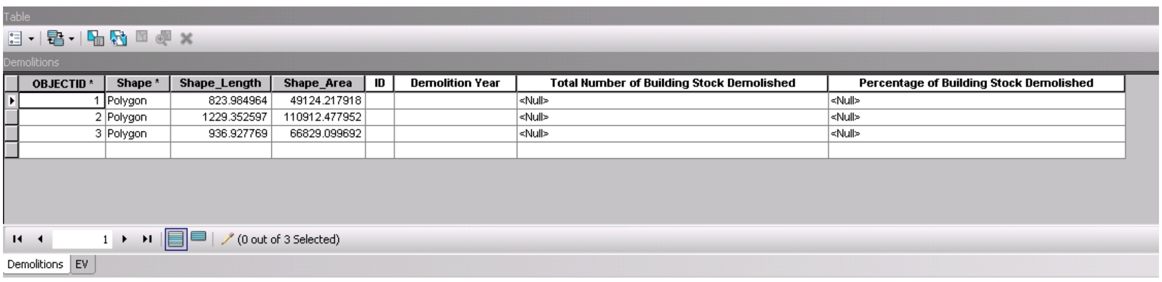


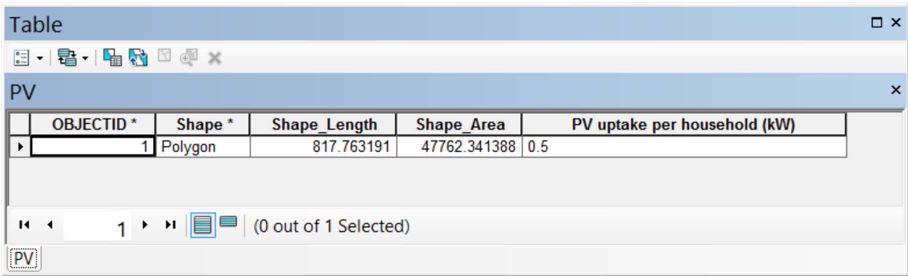
Table
Demolitions

OBJECTID *	Shape *	Shape_Length	Shape_Area	ID	Demolition Year	Total Number of Building Stock Demolished	Percentage of Building Stock Demolished
1	Polygon	823.984964	49124.217918			<Null>	<Null>
2	Polygon	1229.352597	110912.477952			<Null>	<Null>
3	Polygon	936.927769	66829.099692			<Null>	<Null>

(0 out of 3 Selected)
Demolitions | EV

15. For the ‘PV’ and ‘EV’ layer, the editing steps are exactly the same. For the PV layer, the user needs to define ‘PV uptake per household (m2)’ for each individual region drawn in the layer, while for EV layer the ‘ID’ and ‘Electric vehicles per household’ has to be defined for each individual region drawn in the layer. The user can define the EV uptake

for up to 5 time periods to define the future uptake as well as existing stock. The time periods and the EV uptake have to be in increasing order for each ID entry.

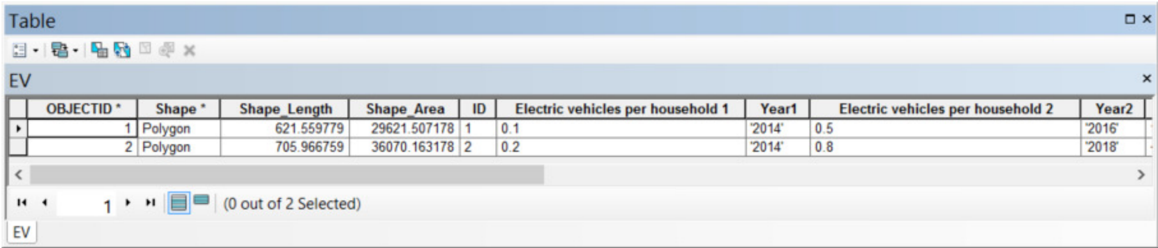


Table

PV

OBJECTID *	Shape *	Shape_Length	Shape_Area	PV uptake per household (kW)
1	Polygon	817.763191	47762.341388	0.5

(0 out of 1 Selected)



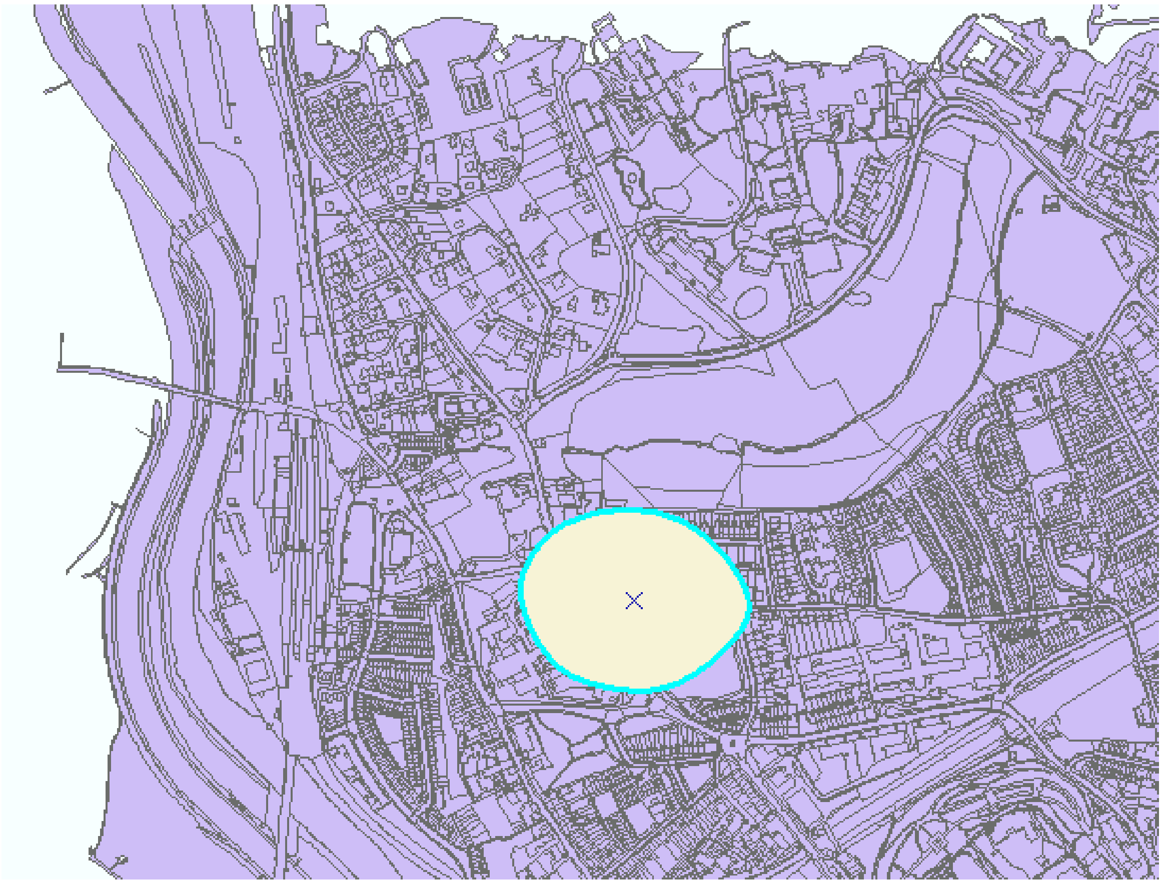
Table

EV

OBJECTID *	Shape *	Shape_Length	Shape_Area	ID	Electric vehicles per household 1	Year1	Electric vehicles per household 2	Year2
1	Polygon	621.559779	29621.507178	1	0.1	'2014'	0.5	'2016'
2	Polygon	705.966759	36070.163178	2	0.2	'2014'	0.8	'2018'

(0 out of 2 Selected)

16. To identify 'ObjectID' the relevant region drawn in the layer, select the row and its corresponding area is highlighted in the layer.



9.6. ArcGIS [E] Define technology packages

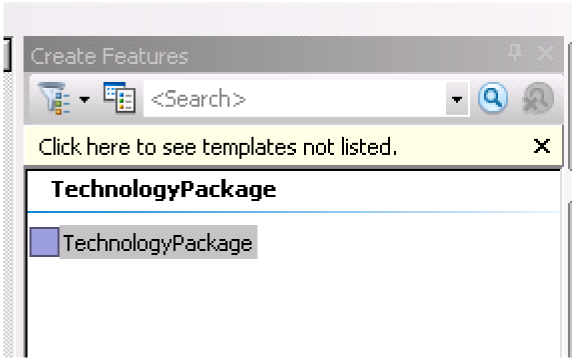
An automated python script creates a feature dataset that allows users to draw polygons and identify technology packages for each individual polygon. The contents of each package are defined separately through the main Excel UI and can contain one or more network-level supply or storage technology options (whose individual characteristics have also been defined separately).

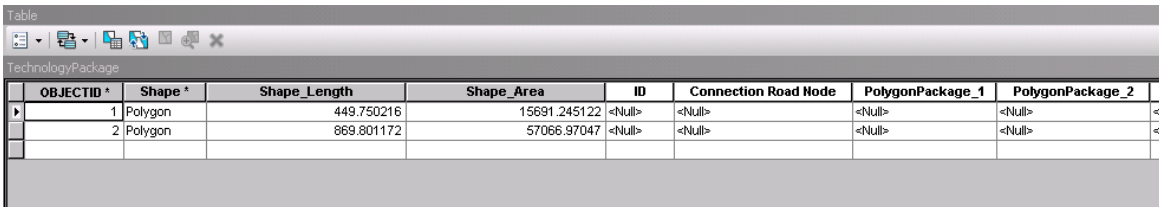
9.6.1. Steps for editing the technology package layer

1. Repeat steps 1-13 as identified in the previous section 9.5.1.

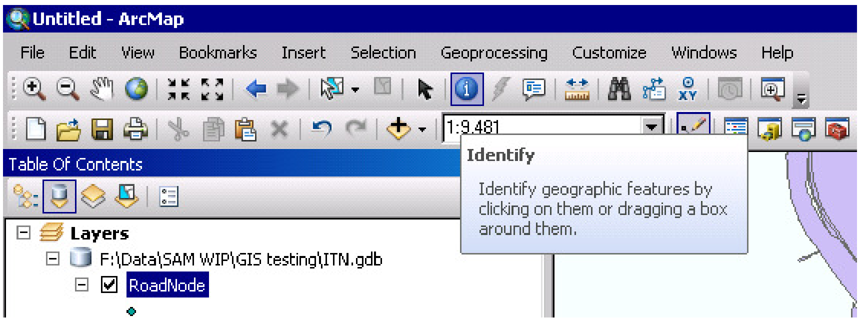
2. Expand the 'ITN.gdb' and drag the 'RoadNode' feature layer into the main view on the left.

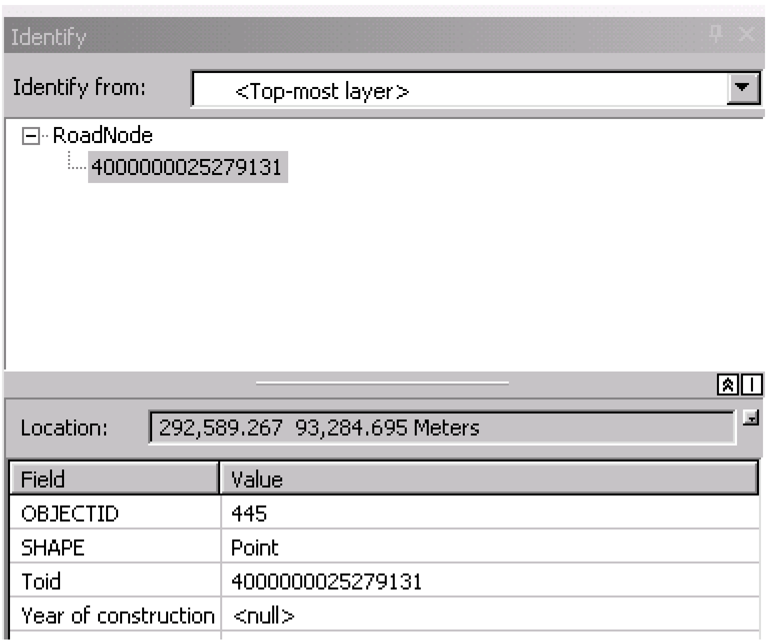
3. Enter the 'ID', 'Connection Road node' and 'Polygon packages' in the attribute table of the 'TechnologyPackage' layer. The user can define up to 20 packages within each drawn region.





4. To identify the id of the relevant road node, select the identify button at the top toolbar ribbon and click on the road node to see the details.



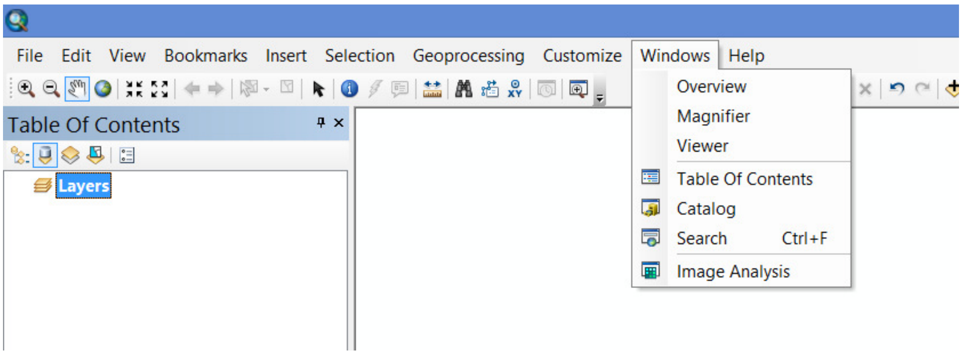


9.7. ArcGIS [G] Define existing electricity / heat network features

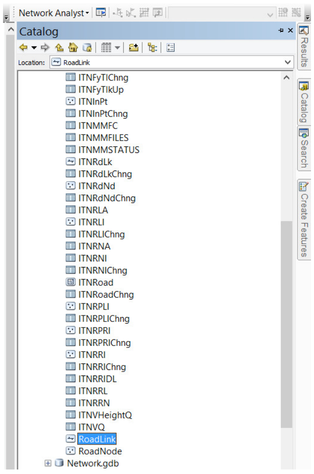
An automated python script creates a feature dataset that allows the user to draw polygons representing areas where there is existing district heating network, as well as to specify the size of the existing electricity network feeders.

9.7.1. Steps for editing the ExistingDHN, ExistingLVFeeder and ExistingHVFeederLinks layers

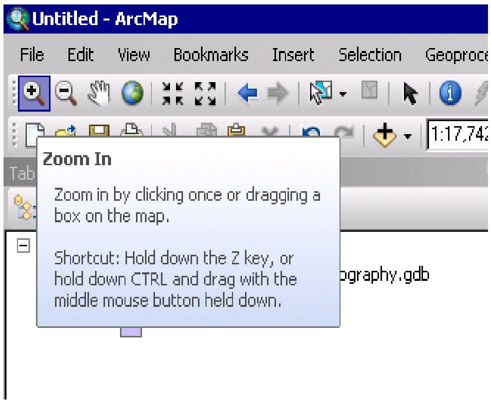
1. Click on the 'Windows' button at the top panel and then select 'Catalog'.



2. This will open up a folder view on the right hand side of the screen.
3. Expand the 'ITN.gdb' and drag the 'RoadLink feature layer into the main view on the left.

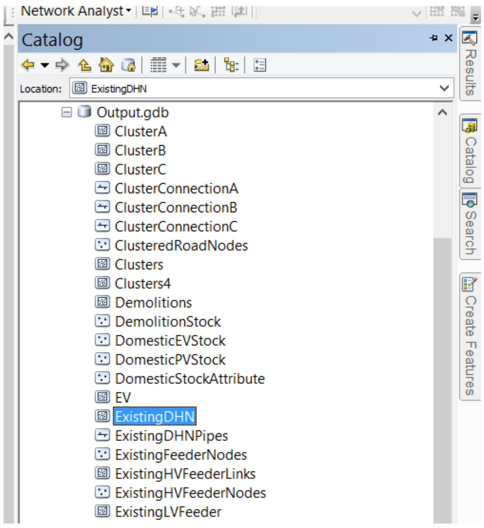


4. Click on the zoom icon at the top right of the screen to zoom into the LA road network map.



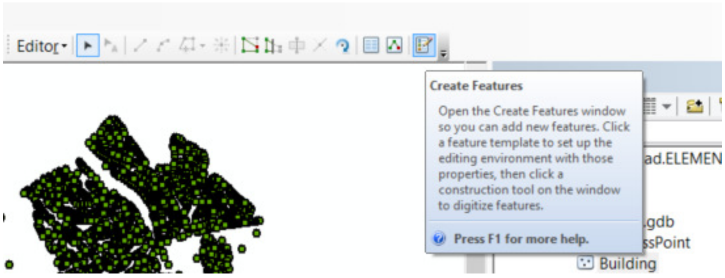
5. To go back to the original view, right click on the 'RoadLink layer in the table of contents and select 'Zoom To Layer'.

6. Expand the 'Output.gdb' and drag the 'ExistingDHN', 'ExistingLVFeeder' and 'ExistingHVFeederLinks' feature layers into the main view on the left.

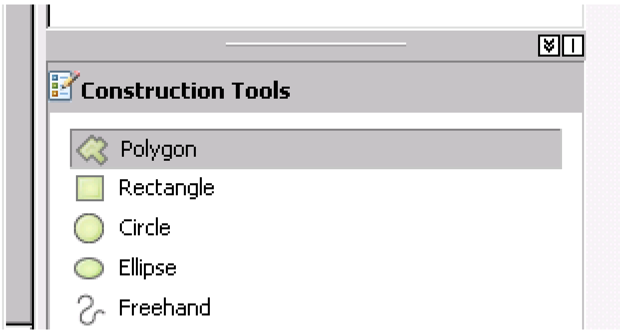


7. Right click on the 'ExistingDHN' layer and select 'Edit features' and 'Start editing'.
8. This will enable the edit toolbar on the top

9. Click on the right most icon called 'Create Features'.



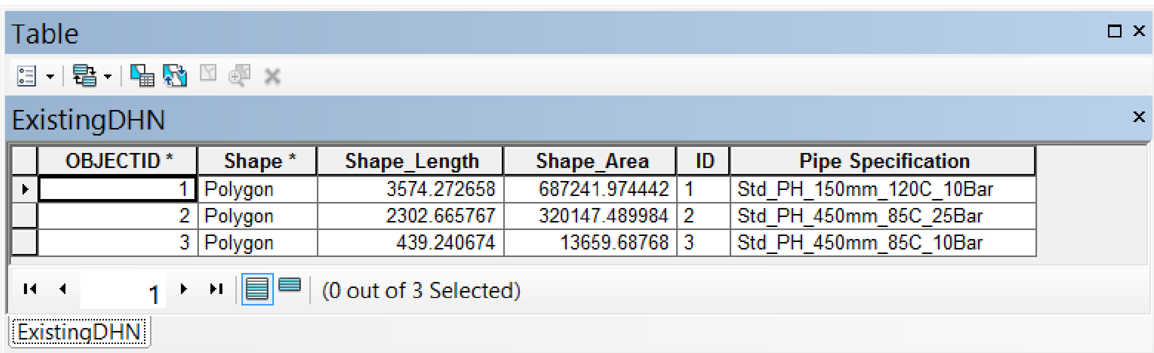
10. This will open the 'Create Feature' tab on the right of the screen, showing the editable layers. Click on the 'ExistingDHN' layer to edit.
11. The available options for creating new features are shown. Select the 'freehand' option to draw the demolition areas.



12. Press the left click on the map at the desired location and drag to draw the boundary of the demolition region, then double click to complete the boundary.

13. Once all the demolition regions have been drawn, right click on the demolition layer in the 'Table of contents' section on the right of the screen and then select 'Open Attribute Table'.

14. This displays the attributes of the layer contents on the right of the screen. Enter 'ID' and Pipe specification' for each area.



Table

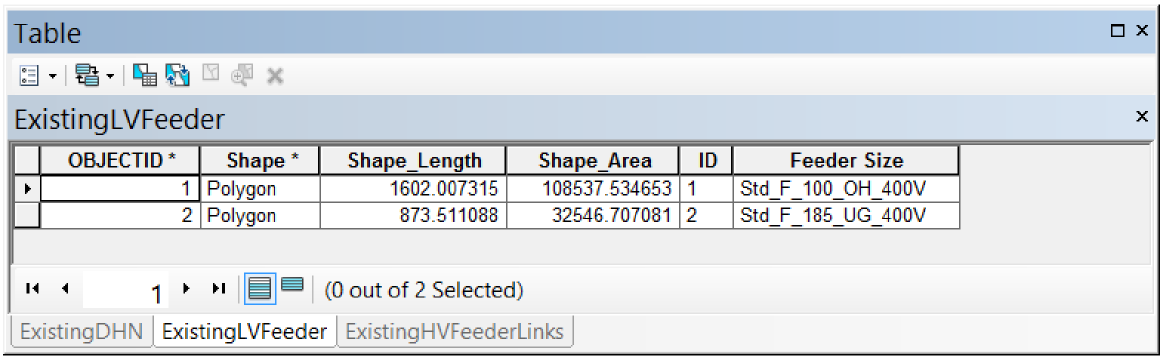
ExistingDHN

OBJECTID *	Shape *	Shape_Length	Shape_Area	ID	Pipe Specification
1	Polygon	3574.272658	687241.974442	1	Std_PH_150mm_120C_10Bar
2	Polygon	2302.665767	320147.489984	2	Std_PH_450mm_85C_25Bar
3	Polygon	439.240674	13659.68768	3	Std_PH_450mm_85C_10Bar

1 (0 out of 3 Selected)

ExistingDHN

15. Repeat the above procedure for 'ExistingLVFeeder' and 'ExistingHVFeederLinks' layers by drawing the regions around the LV feeders and HV feeder links and defining their size.



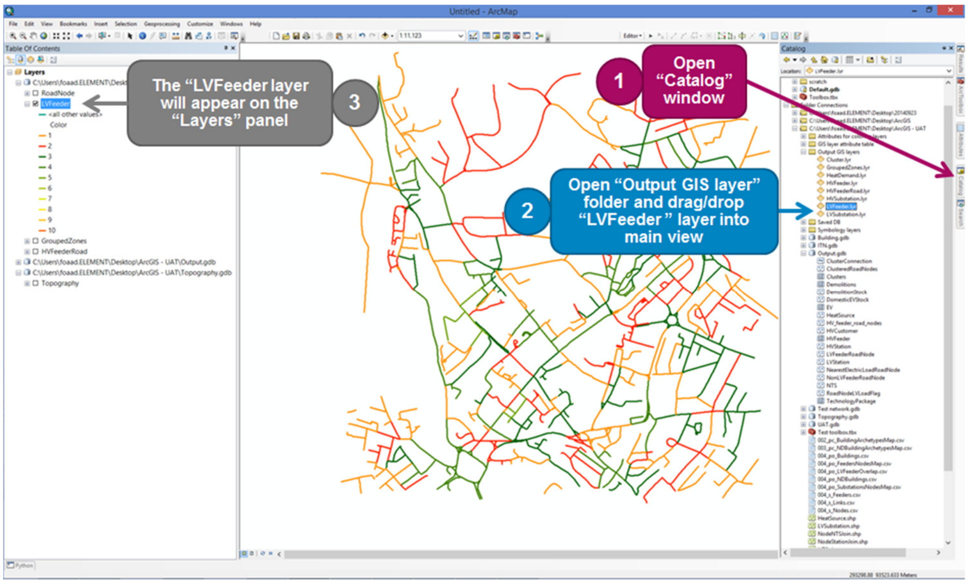
OBJECTID *	Shape *	Shape_Length	Shape_Area	ID	Feeder Size
1	Polygon	1602.007315	108537.534653	1	Std_F_100_OH_400V
2	Polygon	873.511088	32546.707081	2	Std_F_185_UG_400V

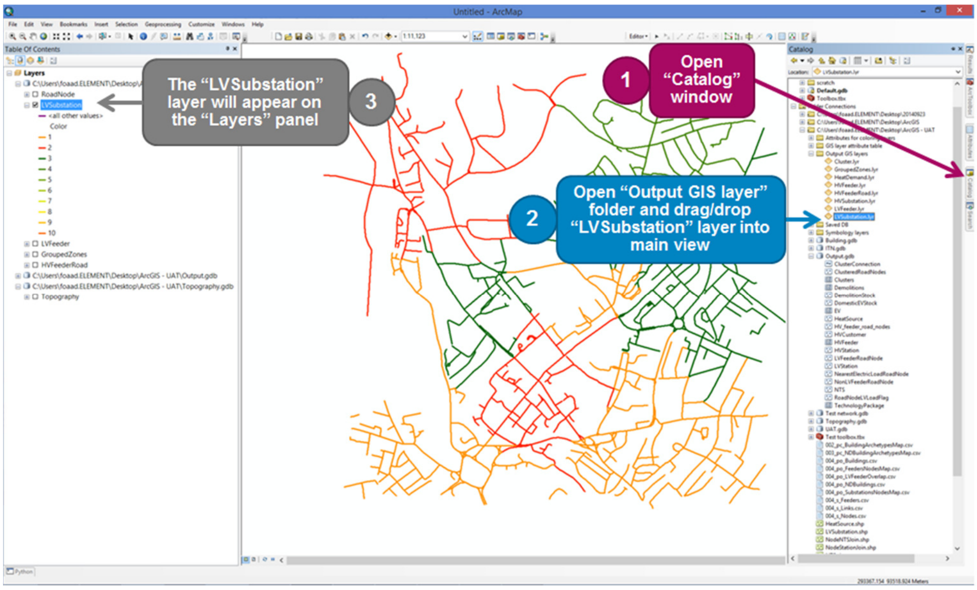
9.8. ArcGIS [H] Create clusters and assign constraints

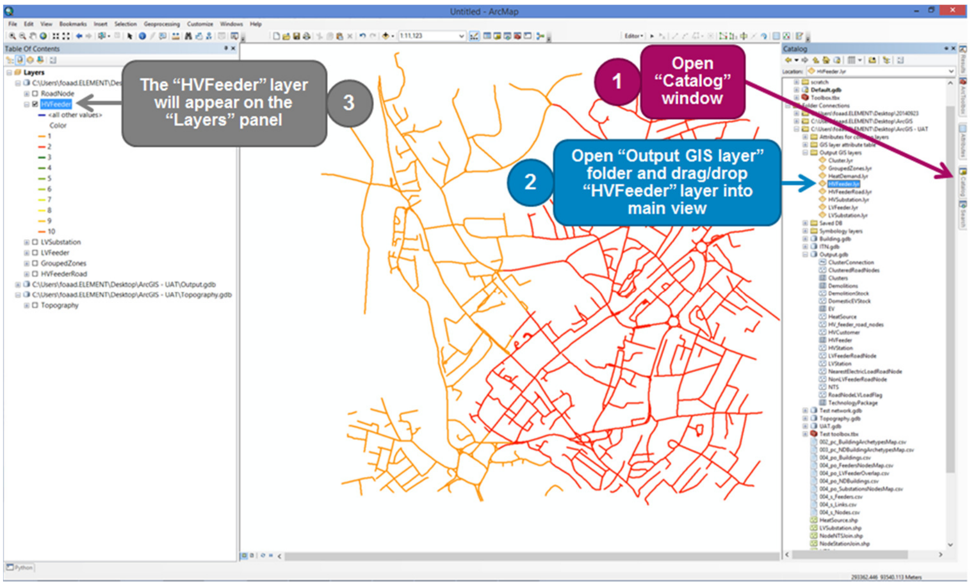
The user is required to define the clusters in the Local Authority area that provide a 'good representation' of both the choices and costs available for heat and electricity networks.

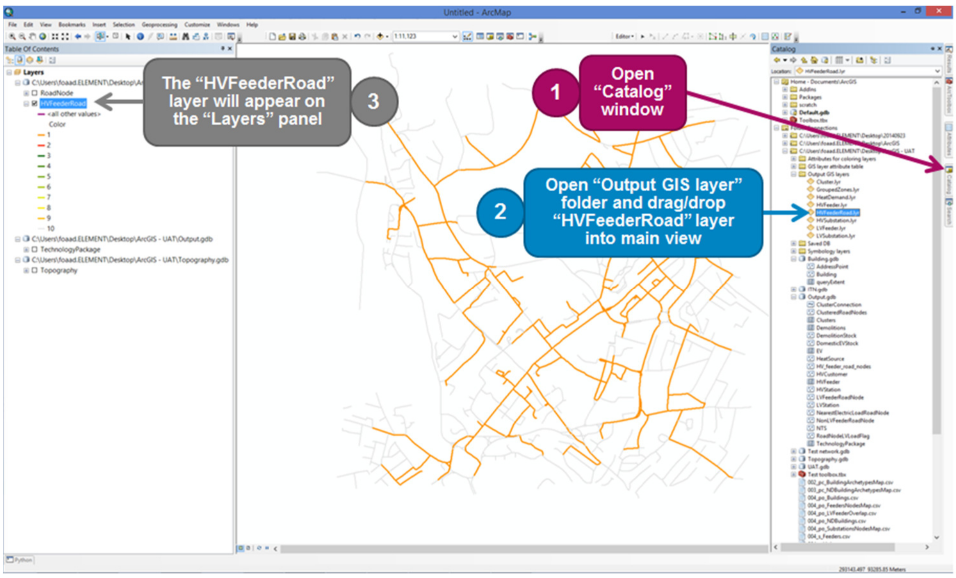
To aid this process a number of key information layers are automatically created in ArcGIS to inform the cluster definition process

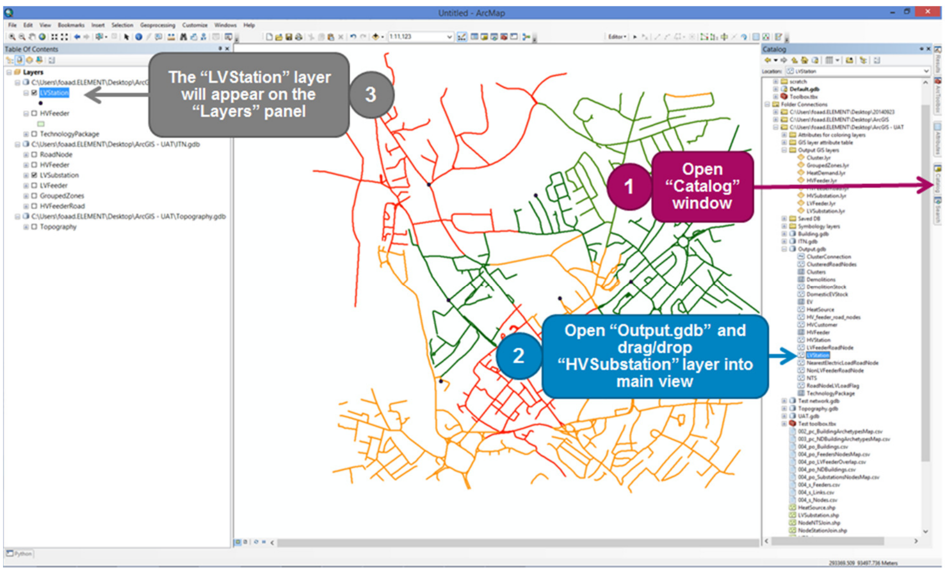
- Layers highlighting the topology of the existing electricity network to help ensure that defined clusters do not cross cut this inappropriately (LV feeders, LV substations, HV feeders, HV substations)

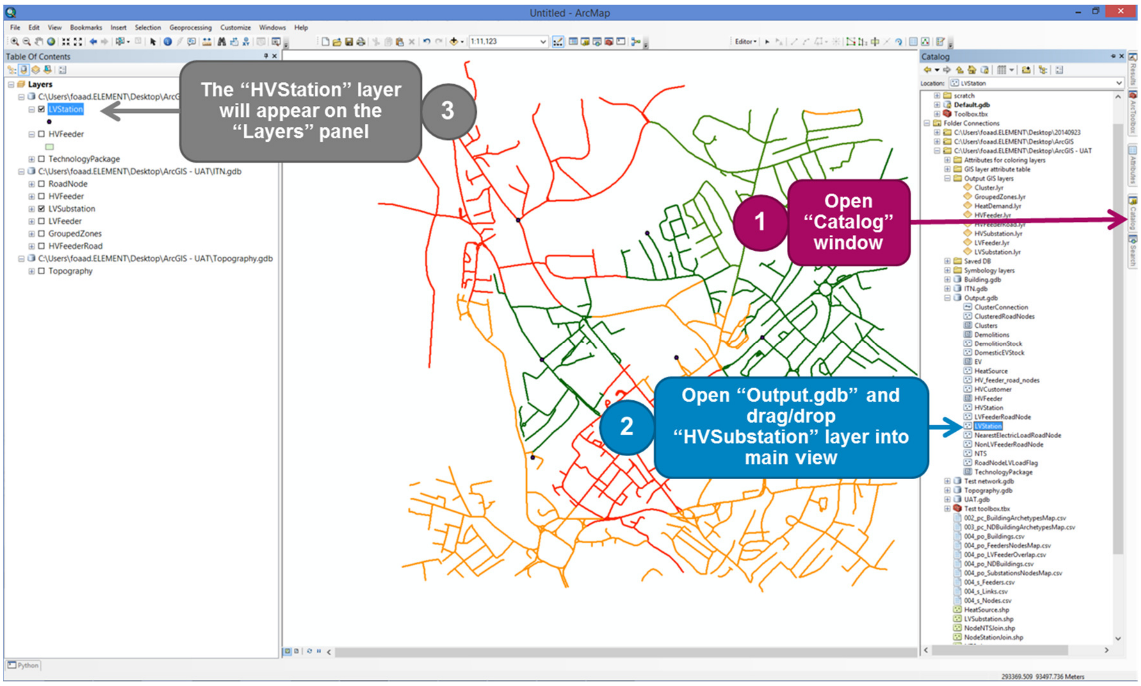

Untitled - ArcMap
1
Open "Catalog" window
2
Open "Output GIS layer" folder and drag/drop "LVFeeder" layer into main view
3
The "LVFeeder layer will appear on the "Layers" panel


Untitled - ArcMap
1
Open "Catalog" window
2
Open "Output GIS layer" folder and drag/drop "LVSubstation" layer into main view
3
The "LVSubstation" layer will appear on the "Layers" panel


Open "Catalog" window
Open "Output GIS layer" folder and drag/drop "HVFeeder" layer into main view
The "HVFeeder" layer will appear on the "Layers" panel


Open "Catalog" window
Open "Output GIS layer" folder and drag/drop "HVFeederRoad" layer into main view
The "HVFeederRoad" layer will appear on the "Layers" panel


Open "Catalog" window
Open "Output.gdb" and drag/drop "HVSubstation" layer into main view
The "LVStation" layer will appear on the "Layers" panel



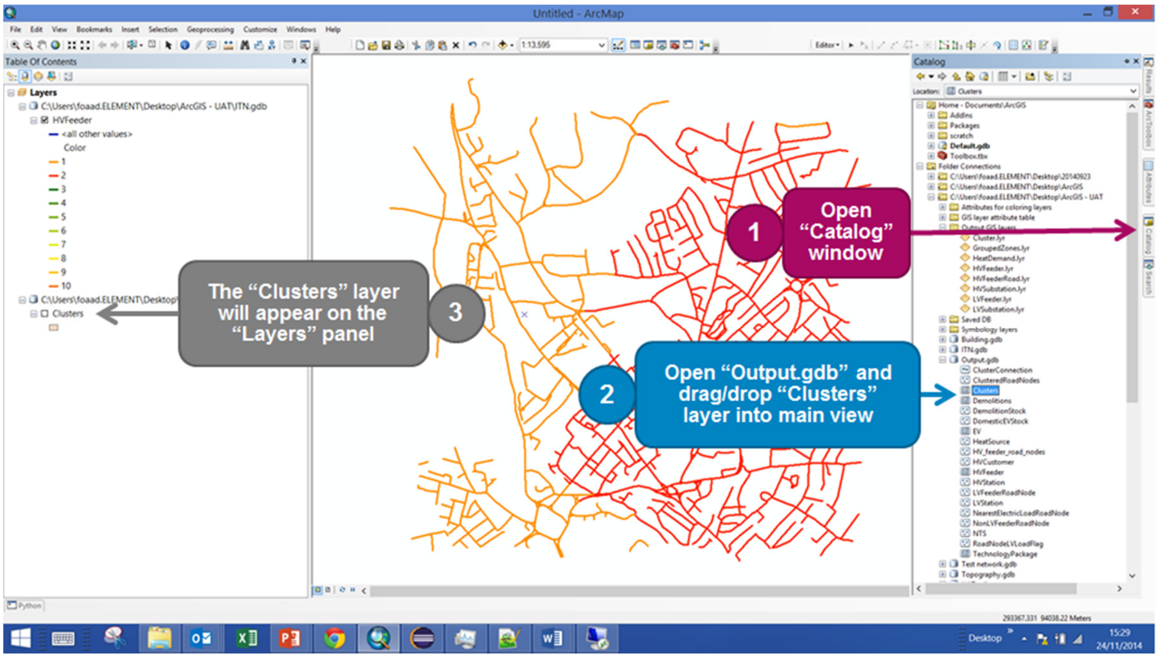
To define the clusters the user has two main options using the ArcGIS interface[33]:

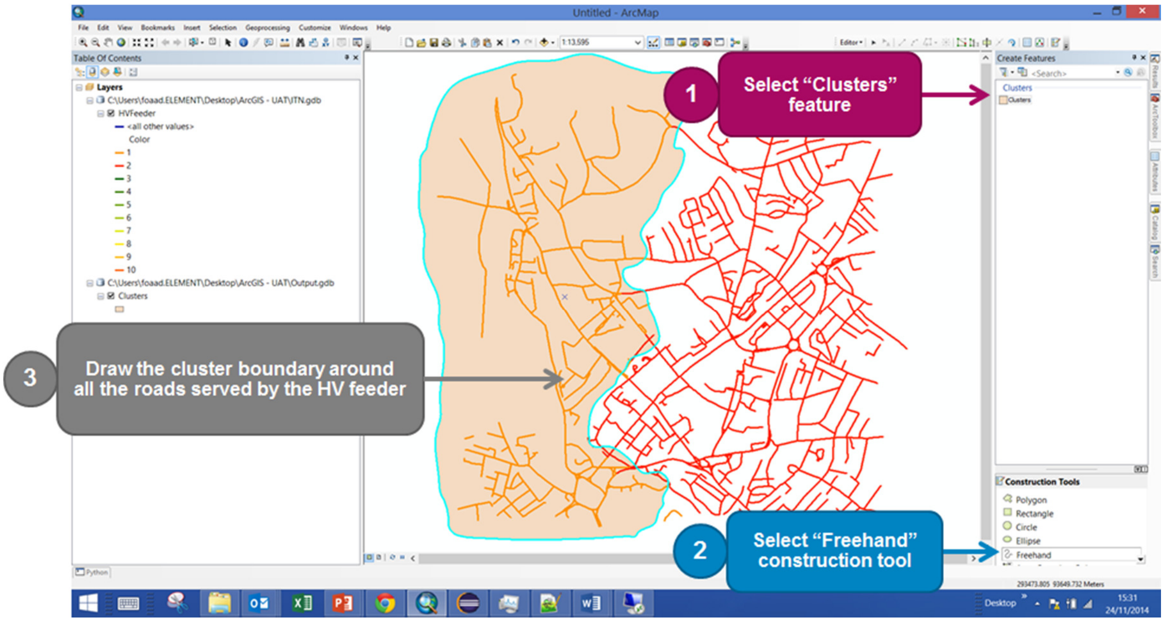
- **METHOD 1)** Draw polygon boundaries manually to define the clusters[34]
  - The final clusters would be based on these polygons and automated minimum overlap thresholds e.g. only network components with more than 50% overlap with the user defined cluster polygons are included, to avoid the user having to define the clusters precisely to ensure they include all features. Also, the remaining individual items upstream of this point are created as separate clusters
- **METHOD 2)** Draw polygon boundaries around key higher-level parts of the electricity system topology (LV substation, HV feeder, HV substation) and use automated processing to map everything connected 'downstream' of this point within the cluster[35] (as well as create clusters for the remaining individual items upstream of this point). For example
  - Selecting single/multiple LV substation would include both these and all LV feeders connected to them
  - Selecting single/multiple HV feeder would include these, all LV substations connected to the HV feeder, and all LV feeders connected all LV substations on the initially selected HV feeder
  - Selecting single/multiple HV substation would include these, all HV feeders connected to the HV substation ,all LV substations connected to the HV feeder,

[33] Another option would be to use to auto-clustering step in NAM-2.
[34] This clustering method is commented out in the latest version of the tool.
[35] Including items indirectly connected to the electricity network topology such as a polygon package

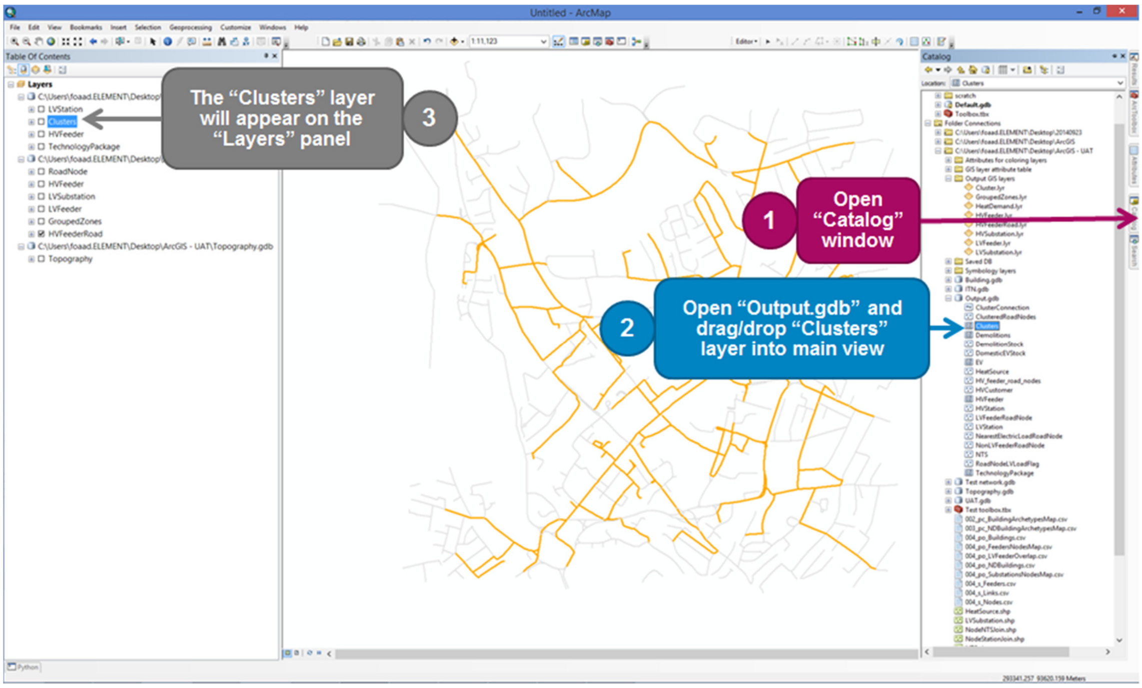
and all LV feeders connected all LV substations on the initially selected HV substation

Example of METHOD 1)

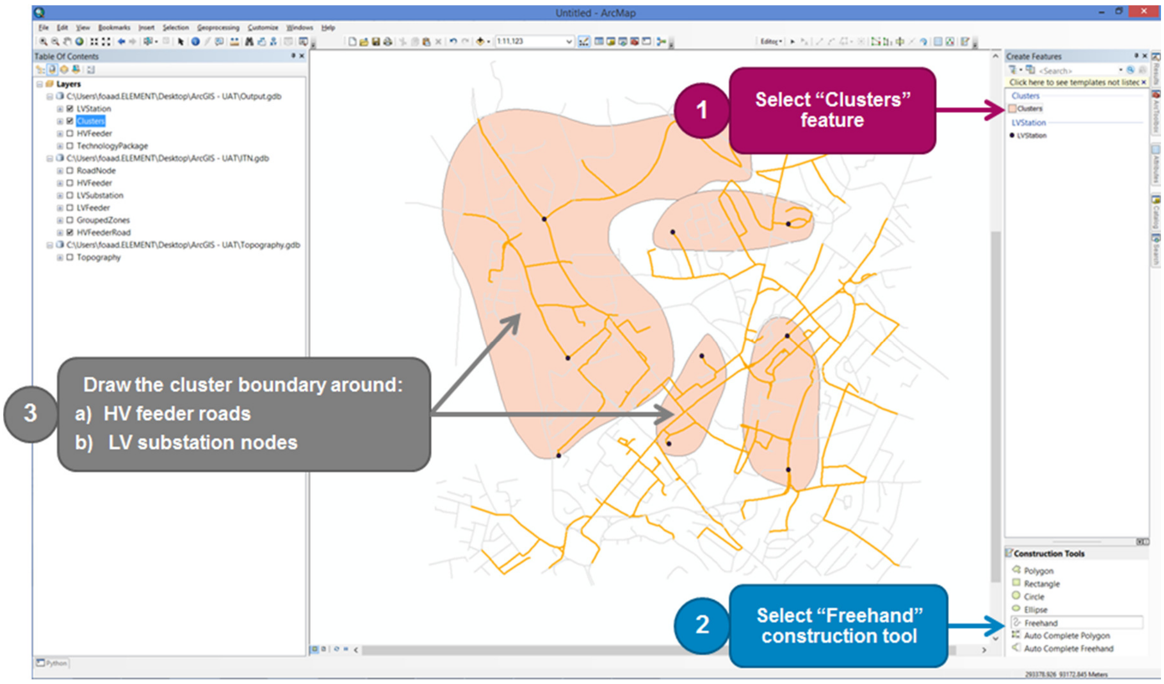




Example of METHOD 2)



The "Clusters" layer will appear on the "Layers" panel
3
1
Open "Catalog" window
2
Open "Output.gdb" and drag/drop "Clusters" layer into main view

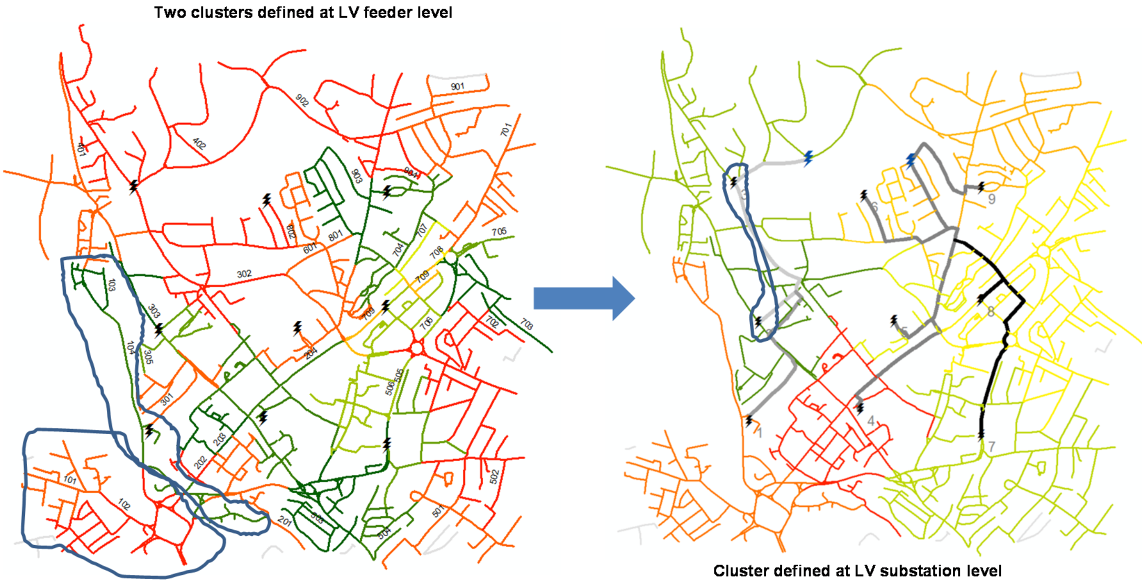


1
Select "Clusters" feature
3
Draw the cluster boundary around:
a) HV feeder roads
b) LV substation nodes
2
Select "Freehand" construction tool

Illustrative example of network topology at decreasing levels of geographic granularity (LV feeder to Distribution substation)

1 Roads color coded to represent individual LV feeders

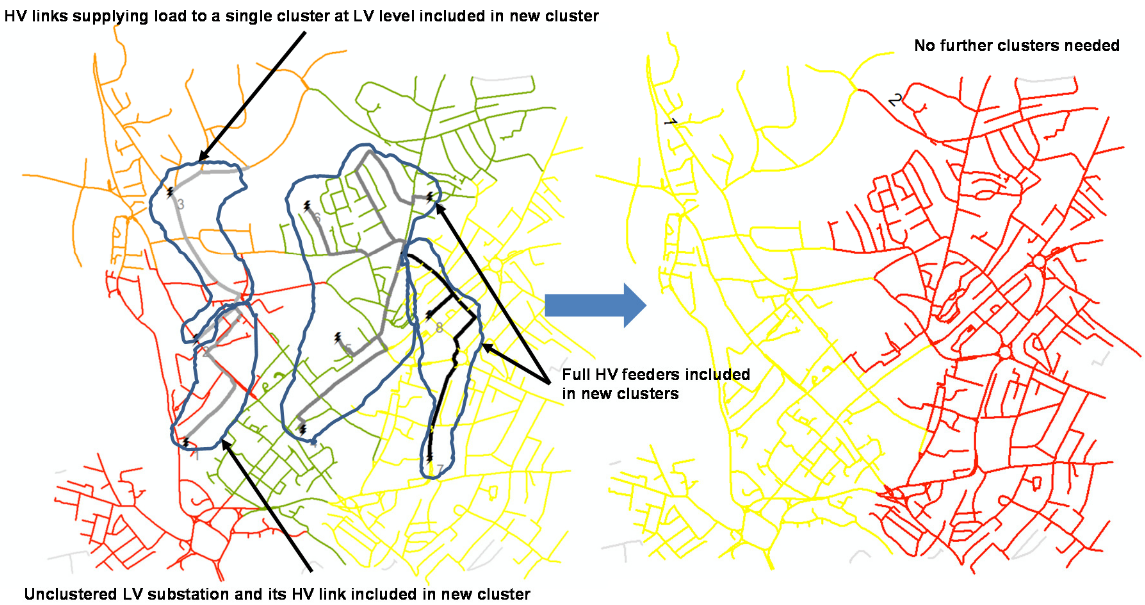
2 Roads colour coded to represent HV feeders and LV feeders connected to individual LV substations



Illustrative example of network topology at decreasing levels of geographic granularity (LV feeder upwards to Primary substation)

3 Roads color coded to represent HV feeders and LV feeders connected to individual HV feeders

4 Roads color coded to represent LV feeders connected to individual HV substations

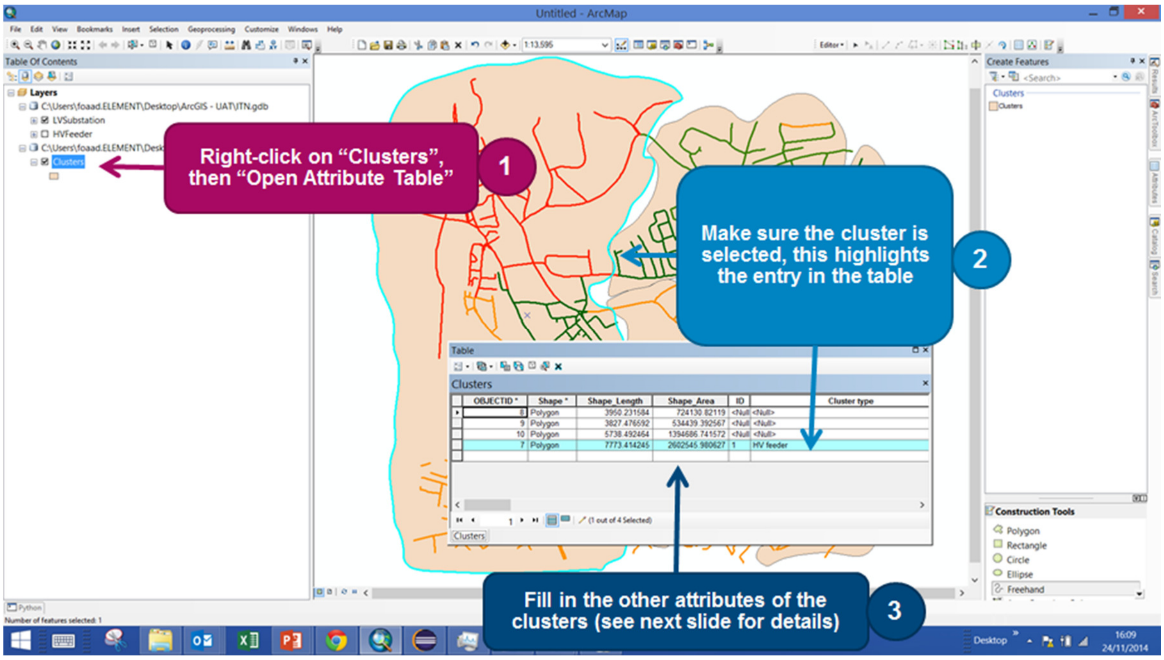


Assigning attributes to clusters after defining them

The combination of the two approaches allows the user to span the spectrum of precise cluster definition for any (valid) combination of clusters to a very quick process, which utilises the building blocks of the electricity system topology to define them.

Once the clusters have been defined the user:

- Must assign relevant attributes to each cluster
  - Unique ID
  - Cluster type (e.g. LV feeder, LV substation, LV substation and downstream etc)
  - The cables status (e.g. over/underground)
  - The locality type (e.g. rural, suburban, urban)
- Assign mean height difference (optional - used for load flow modelling of heat networks)
- Assign (multiple) constraint classes to a cluster (optional)
  - Constraint classes will be defined separately through the main Excel UI in advance (e.g. no wind before year X, or force buildings to be on district heating by year Y)



Table

Clusters

	OBJECTID *	Shape *	Shape_Length	Shape_Area	ID	Cluster type	Locality	Cable status	Mean height difference	Constraint_Upgrade1	Constraint_Upgrade2	Cc
▸	2	Polygon	5189.279014	1209193.898882	1	HV feeder + LV substation + LV feeder	U	UG	0	Conservation	Heritage	<Null>
	4	Polygon	1647.817378	151370.472551	2	LV substation + LV feeder	U	UG	0	StorageClassA	ForceSolar	<Null>
	5	Polygon	1310.734388	99686.761941	3	LV substation + LV feeder	U	UG	0	NetworkClassA	<Null>	<Null>
	6	Polygon	1829.03592	206993.351096	4	LV substation + LV feeder	U	UG	0	TechnologyClassA	<Null>	<Null>

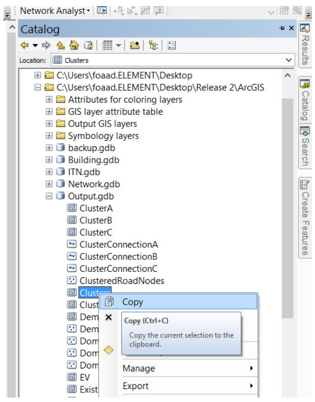
(1 out of 4 Selected)

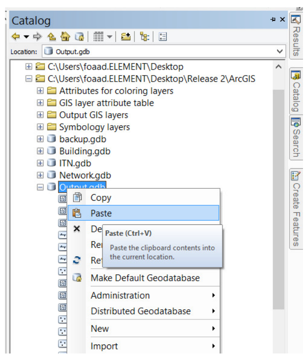
Clusters

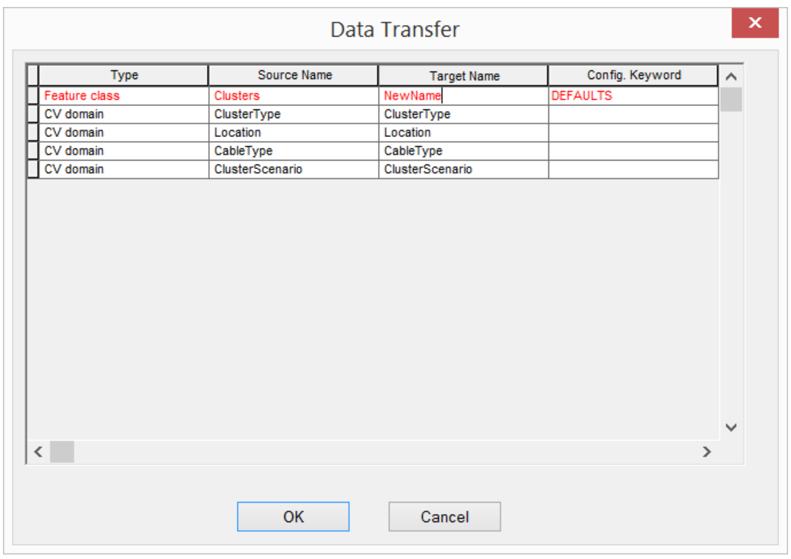
We can then save edits and stop editing using the “Editor” menu.

9.8.1. Defining multiple sensitivity cluster definitions

The user has the ability to define multiple definitions of the clusters, which may vary in geometry (i.e. size) of cluster definition as well as the constraints applied to individual clusters. The user can create multiple layers by copying the ‘Clusters’ layer that is created by python script and re naming it to the relevant cluster for a particular scenario.

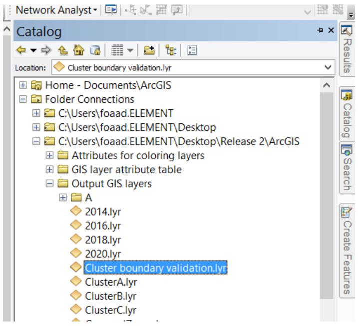


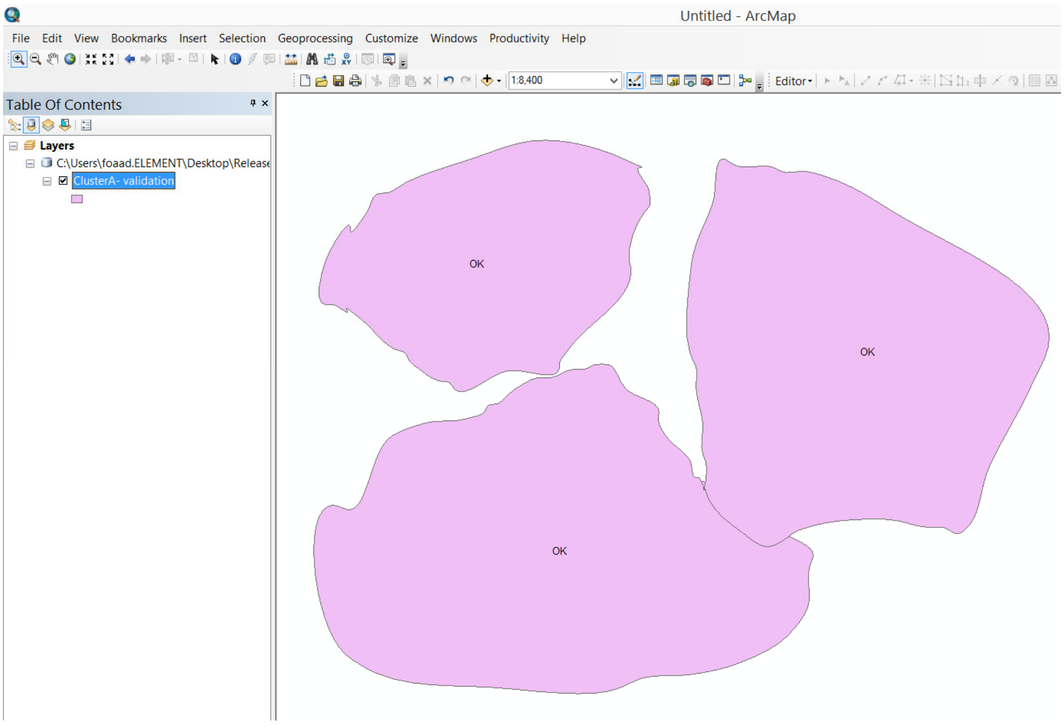




9.8.2. Automated feedback on the cluster definition

The user defined cluster definition is processed to cluster the network elements (LV/HV feeders, LV/HV substations), topology (roads, road nodes) as well as the connected buildings and technology packages. However, the clustering process must obey the network topology rules i.e. two network elements can only be clustered if they have the same upstream element and all of their downstream elements are also included in the same cluster. A cluster definition validation layer is created in the Output GIS layers folder which informs the user where these rules are violated and gives simple suggestions to correct the cluster layer.



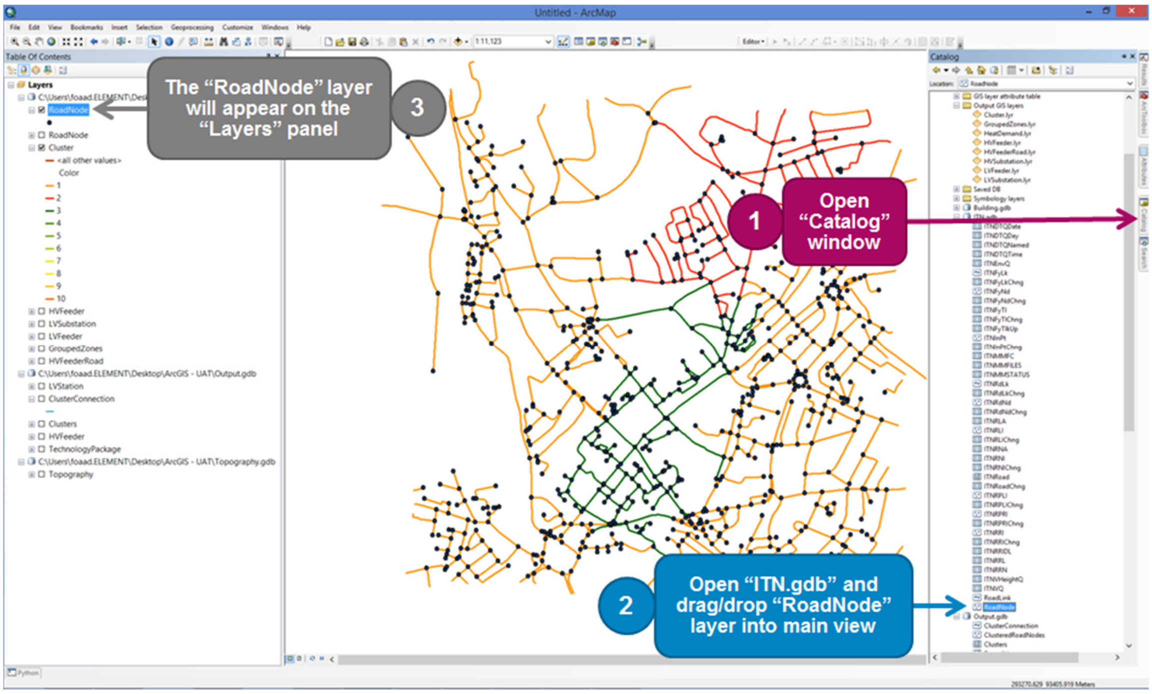


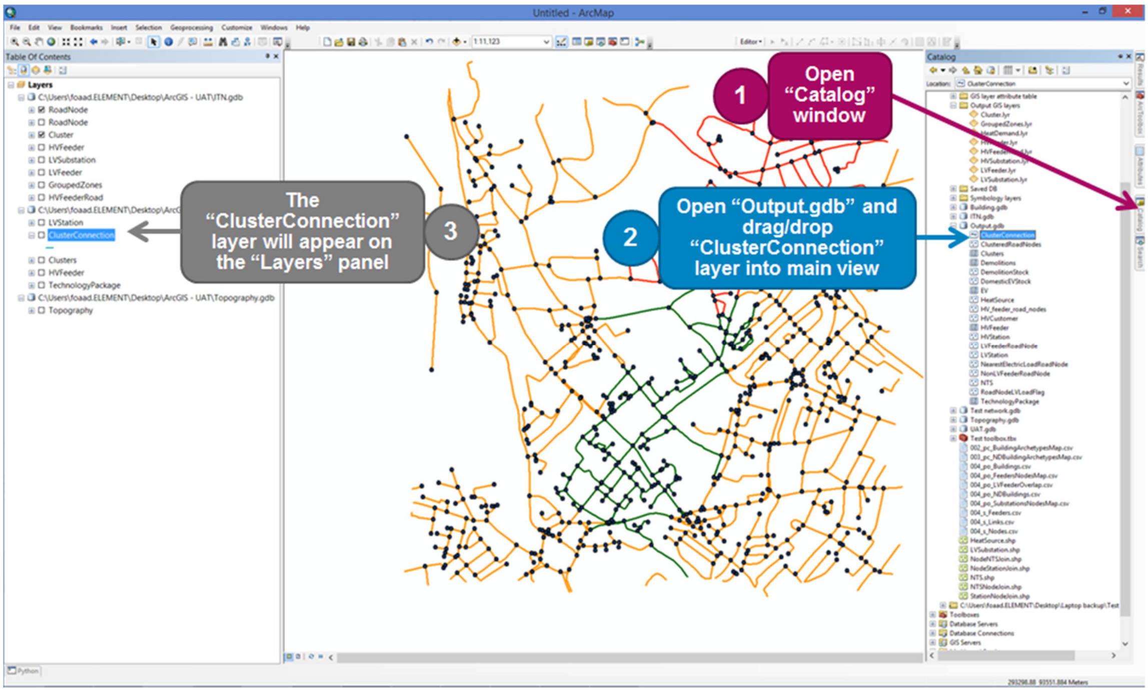
9.9. ArcGIS [I] Define inter-cluster connections and mesh options

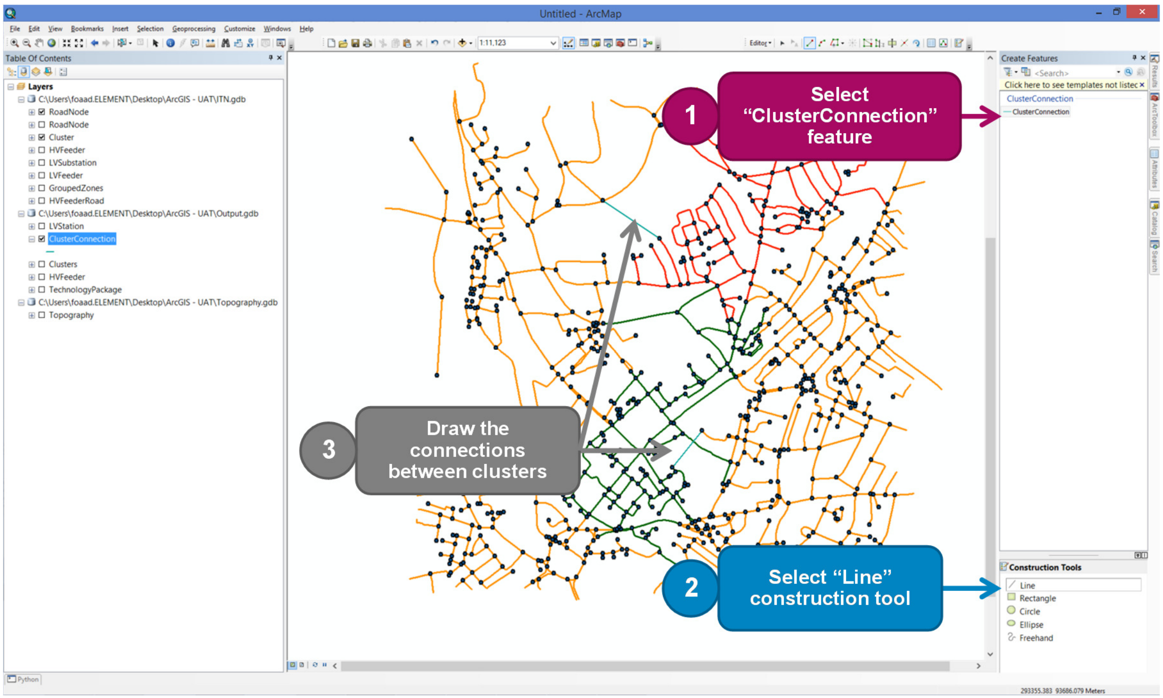
The user is required to define the possible *new* meshing options for the electricity network, as well as the existing or possible inter cluster connections for heat and gas networks. For existing electricity networks, connections between clusters are defined automatically by the existing network topology.

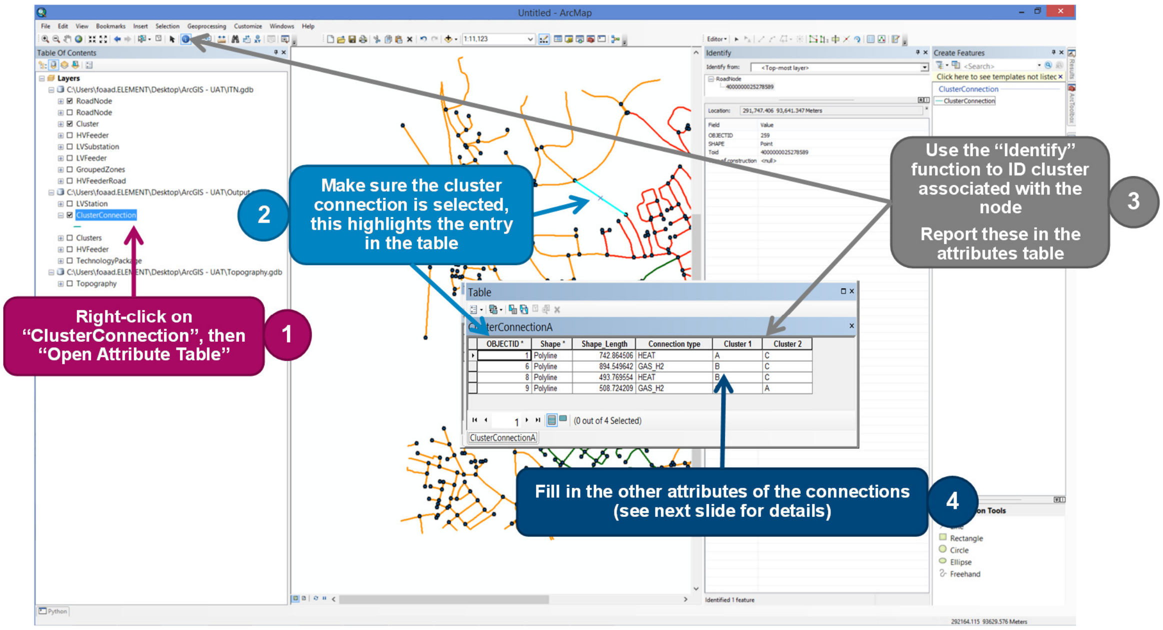
- Import “RoadNode” layer in ArcGIS

- Import “ClusterConnection” layer in ArcGIS
- Draw the cluster connections
- Define attributes for the cluster connections
- Import “Meshing” layer in ArcGIS
- Draw the meshing connections
- Define attributes for the meshing connections









In the attributes table, key in relevant properties:

Table

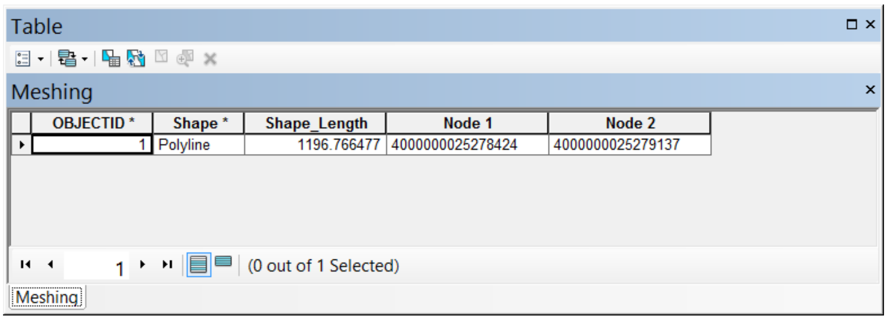
ClusterConnectionA

OBJECTID *	Shape *	Shape_Length	Connection type	Cluster 1	Cluster 2
1	Polyline	742.864506	HEAT	A	C
6	Polyline	894.549642	GAS_H2	B	C
8	Polyline	493.769554	HEAT	B	C
9	Polyline	508.724209	GAS_H2	C	A

1 (0 out of 4 Selected)

ClusterConnectionA

An analogous process is undertaken to define potential meshing options via the "Meshing" layer, but noting that these are defined via the attributes as node-to-node connections, as per the example below (rather than cluster-to-cluster connections).



We can then save edits and stop editing using the "Editor" menu.

9.9.1. Defining multiple sensitivities for inter-cluster connection definitions

The user has the ability to define multiple definitions of the inter-cluster connections. The user can create multiple layers by copying the 'ClusterConnection' layer that is created by python script and re-naming it to the relevant inter-cluster connection for a particular scenario.

9.10. ArcGIS [J] View spatial results from POM

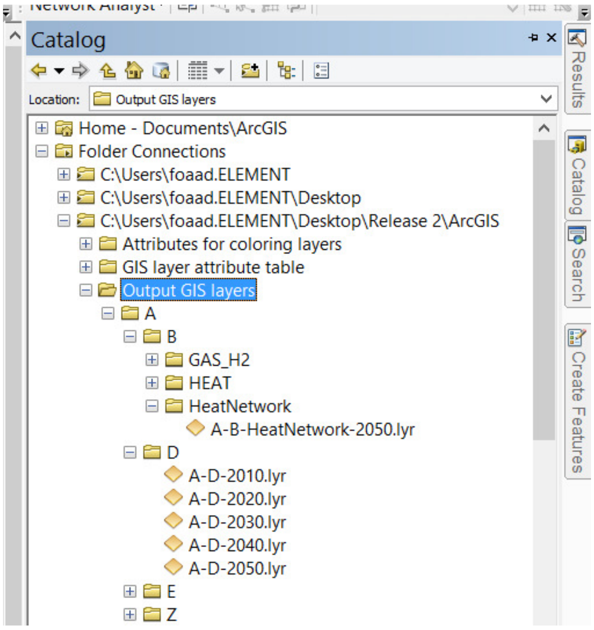
The POM generates several outputs at a cluster level at the conclusion of its optimisation. These outputs relate to decisions on building upgrades, heating technology uptake, investment in the electricity network, installation and expansion of heating network etc. An automated python script loads a predefined set of these outputs into GIS layers and allow the user to view these spatially via colour coding, labels or symbology as appropriate:

- There are three main ArcGIS results structures
  - r_ClusterMetrics
    - Reporting results to underlying cluster e.g. Number of buildings with X heating technology
  - r_ClusterConnectionMetrics
    - Inter-cluster connection metrics, flows / capacity for heat etc
  - r_HeatNetworkSize
    - Reporting results to underlying district heating network e.g. size and coverage of pipes in 2020
- Plus supplementary reporting parameters to cover
  - Labelling of r_ClusterLabels / r_PolygonLabels (for packages)

The MCF uploads multiple layers of results to a sub-folder structure in the "Output GIS layer" folder, based on the unique layer name assigned to the results e.g:

- Sim0|ArchetypeHeatingByCluster|GasBoiler|2020
- Sim0|ArchetypeHeatingByCluster|GasBoiler|2030.

Each '|' represents a further sub-folder generated in ArcGIS, for example in the above case the final subfolder relates to GasBoiler-related results and it would contain 2-layers, one with results for 2020 and the other for 2030.



9.11. Define number of simulations (and generate Monte Carlo inputs)

The following input parameters can be simulated from R2 and later

- Capital costs (domestic heating and efficiency measures, non-domestic heating options, network options, other technology and storage costs)
- Product costs
- Insulation U-Values
- Domestic heating system efficiencies
- Non-domestic energy benchmarks

For most of these parameters

- Indices are modelled in percentage terms
- One custom index is assigned for each simulated parameter
- Generic indices are user-defined and can be assigned to several simulated parameters to ***indirectly*** drive correlations (as unlike custom indices, generic indices can be explicitly correlated to each other)
- The final simulated parameter is a weighted average between one custom and one or several generic indices (sum of indices weights should equal to 1)

As an exception, some indices and the correlations between them are extracted from the ESME run for boundary conditions: these are in the tables:

- [esme].[pi_GenericIndicesMap]
- [esme].[pi_ProductMap]
- [esme].[pi_GenericTechMap]

These procedures are used for extracting simulated information from the ESME boundary DB as input to the EPN model:

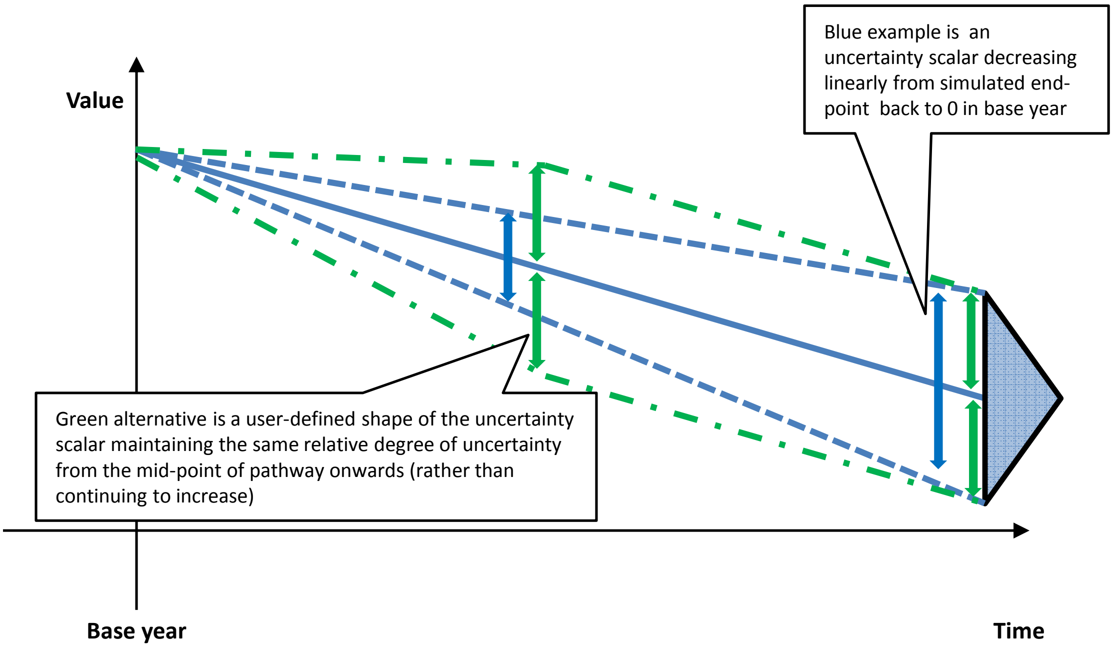
- [esme].[sp_WriteGenericIndicesCorelations]
- [esme].[sp_WriteGenericIndicesSimulated]
- [esme].[sp_WriteIndicesProfiles]
- [esme].[sp_WriteIndicesSimSettings]

The use of both custom and generic indices gives the user the flexibility to simulate and correlate parameters without the need to define large correlation matrices covering all possible custom indices.

The base (deterministic) time period data for the parameters that can be simulated reflects their annual evolution. As part of the Monte Carlo simulation process

- We only simulate around the 2050 end-point (via a triangular distribution) and effectively scale the deterministic pathway data from the base year to this simulated end-point (as opposed to simulating every intermediate year)
- The scaling applies a user-defined shape, which can be changed according to a view of how the *relative* cone of uncertainty is likely to evolve over time. The example below shows the simulated end-point and two alternative pathway shapes - one reflecting a linearly increasing shape of uncertainty from the base year to the end-point, and another where the maximum cone of *relative* uncertainty increases linearly to the mid-point of the pathway, but is then maintained until the end of the pathway[36]. The green line is obtained by updating the values in [monte_carlo].[pi_CustomIndicesProfiles], starting with value 1 in 2014, linearly increasing to 2 in 2030, then linearly decreasing to 1 in 2050.

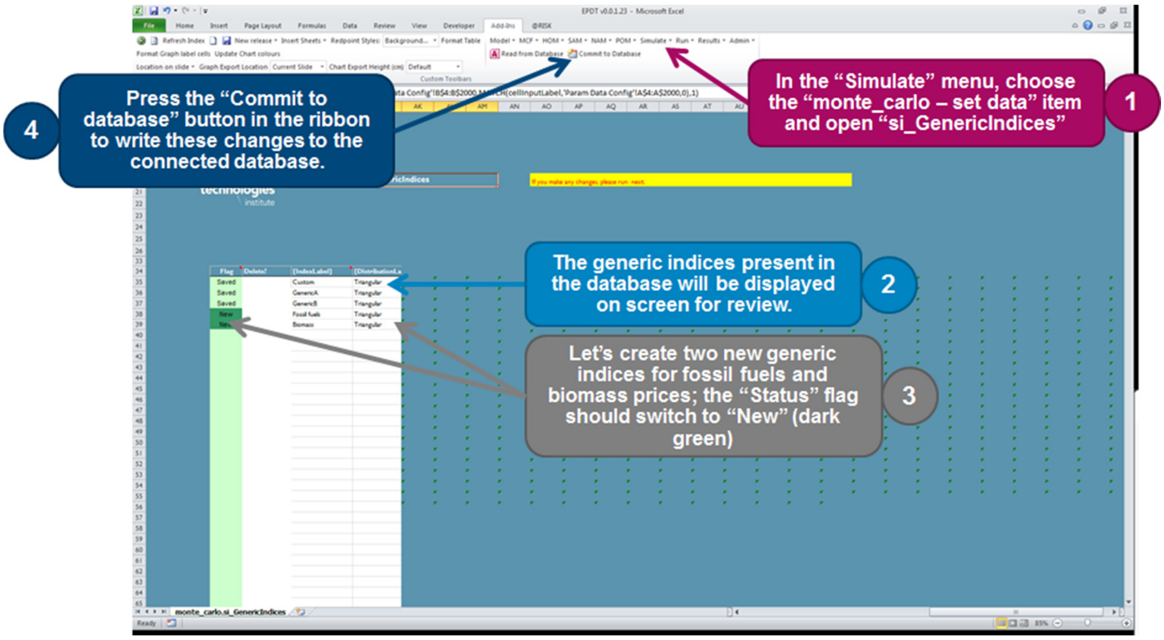
[36] The latter may be more applicable to parameters such as fossil fuel cost projections where our maximum view of the uncertainty range (min to max of e.g. -/+ 50%) is hit in the relatively near term (~10-15 years). Instead of having to assume that the range of uncertainty continues to increase regardless of the length of the pathway, we can maintain this relative uncertainty level beyond the 10 year horizon.

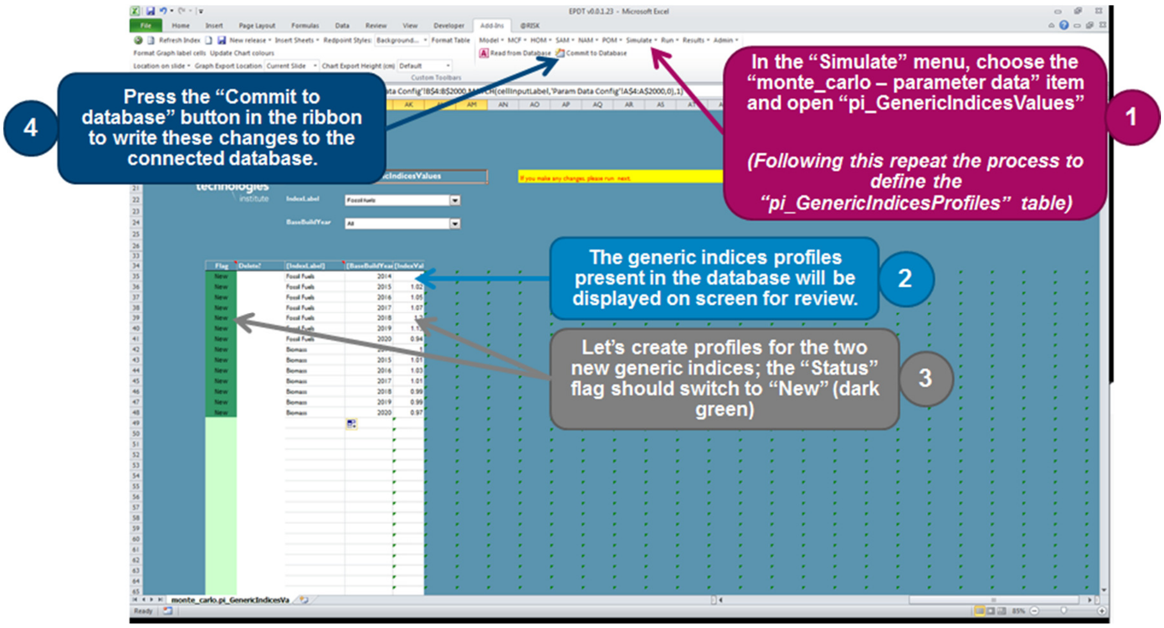


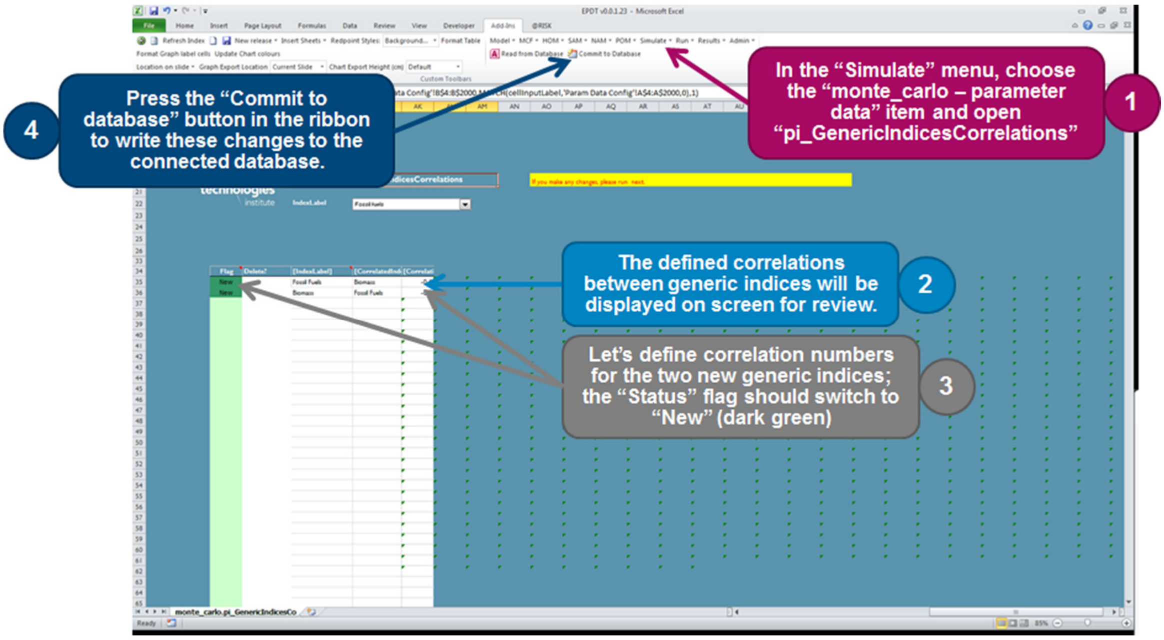
Example of creating and correlating two new generic indices

The example below describes the setup of new, correlated generic indices, to which custom indices are assigned (via a weighting); via the following these steps:

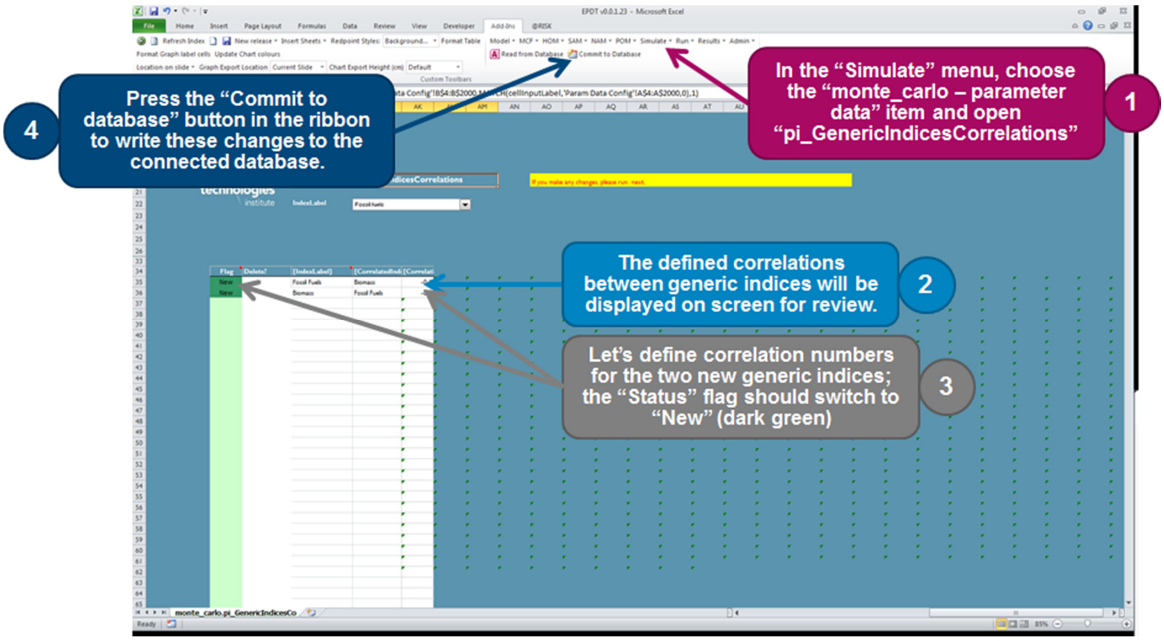
1. Create new generic indices
2. Define their values and uncertainty profiles across the modelling horizon
3. Specify the correlations between these generic indices
4. Configure custom indices
5. Allocate or "weight" these generic indices to simulated parameters
6. Define the parameters of the triangular distribution
7. Run the simulations using @Risk
8. Review simulated input parameters in the Excel UI before running POM
9. [Optional] Some simulated parameters are used in other parts of the model before POM so the simulation should be run before the relevant sub-modules:
    - E.g. U-values for HOM-1,
    - E.g. heating system costs for Monte Carlo simulation of archetype costs after HOM-4,
    - E.g. non-domestic building retrofitting costs for SAM-7.


In the "Simulate" menu, choose the "monte_carlo – set data" item and open "si_GenericIndices"
1
The generic indices present in the database will be displayed on screen for review.
2
Let's create two new generic indices for fossil fuels and biomass prices; the "Status" flag should switch to "New" (dark green)
3
Press the "Commit to database" button in the ribbon to write these changes to the connected database.
4

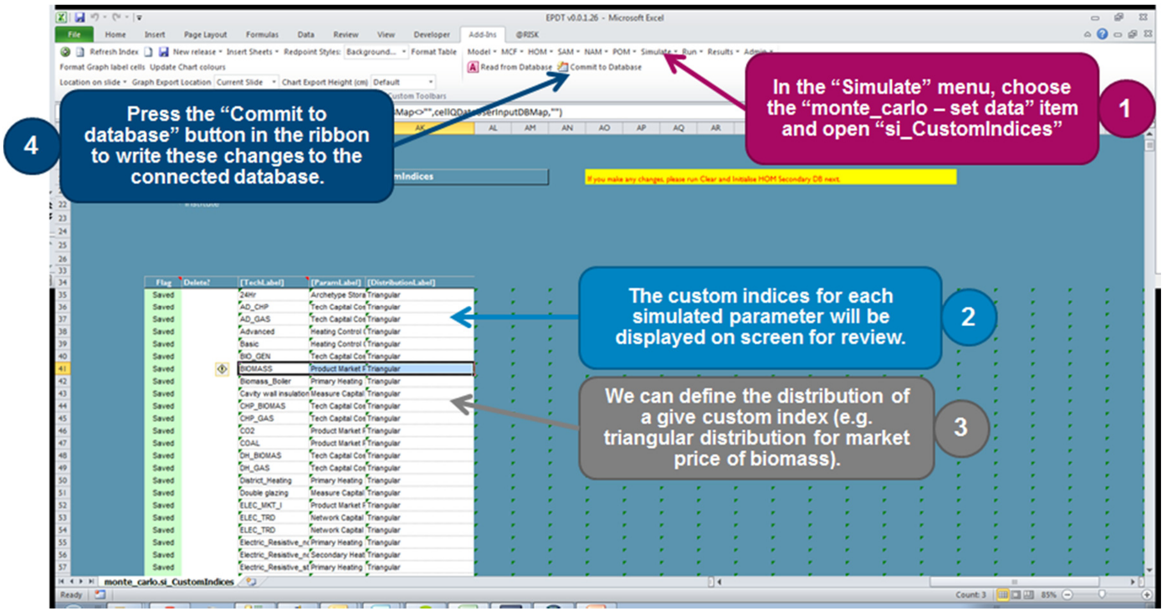

In the "Simulate" menu, choose the "monte_carlo – parameter data" item and open "pi_GenericIndicesValues"
(Following this repeat the process to define the "pi_GenericIndicesProfiles" table)
1
The generic indices profiles present in the database will be displayed on screen for review.
2
Let's create profiles for the two new generic indices; the "Status" flag should switch to "New" (dark green)
3
Press the "Commit to database" button in the ribbon to write these changes to the connected database.
4

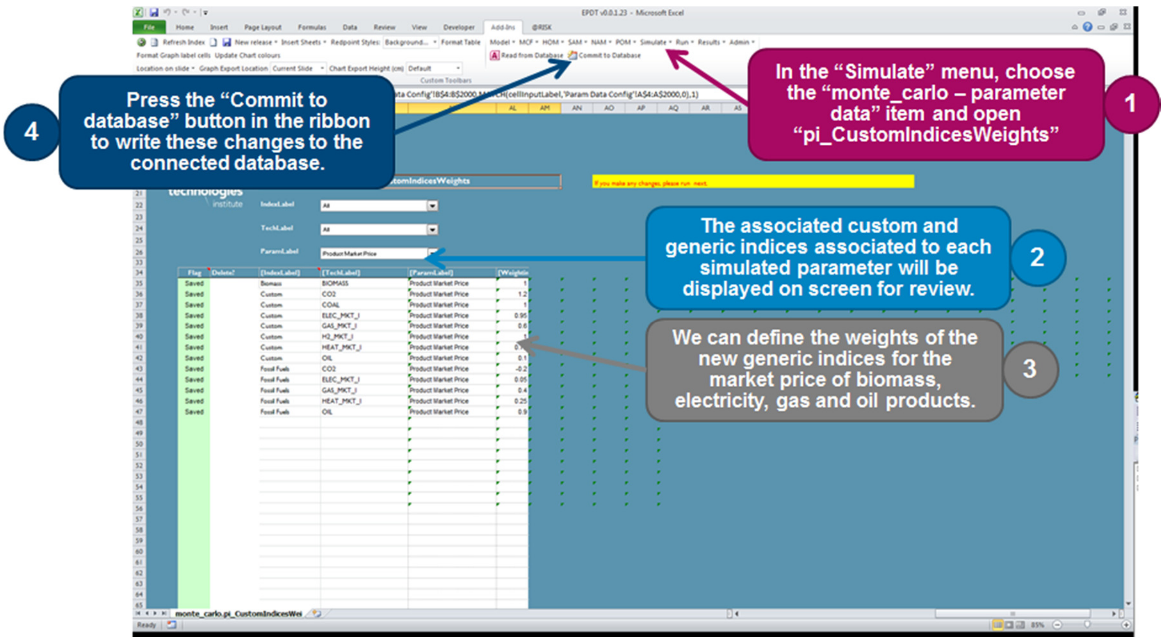


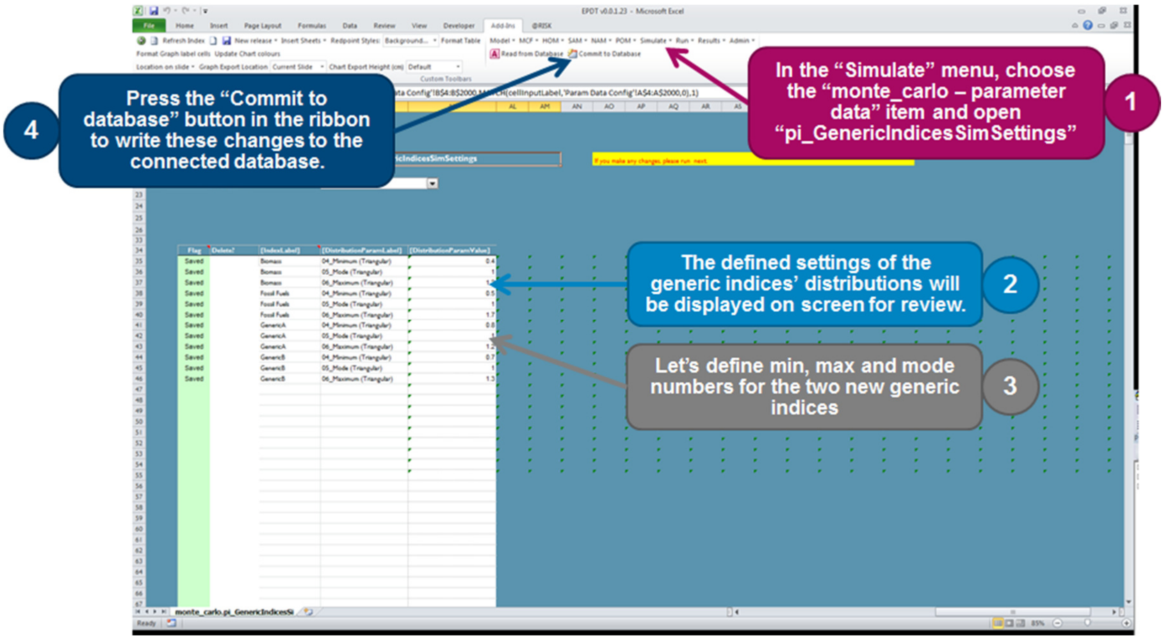
Press the "Commit to database" button in the ribbon to write these changes to the connected database.
4
In the "Simulate" menu, choose the "monte_carlo – parameter data" item and open "pi_GenericIndicesCorrelations"
1
The defined correlations between generic indices will be displayed on screen for review.
2
Let's define correlation numbers for the two new generic indices; the "Status" flag should switch to "New" (dark green)
3

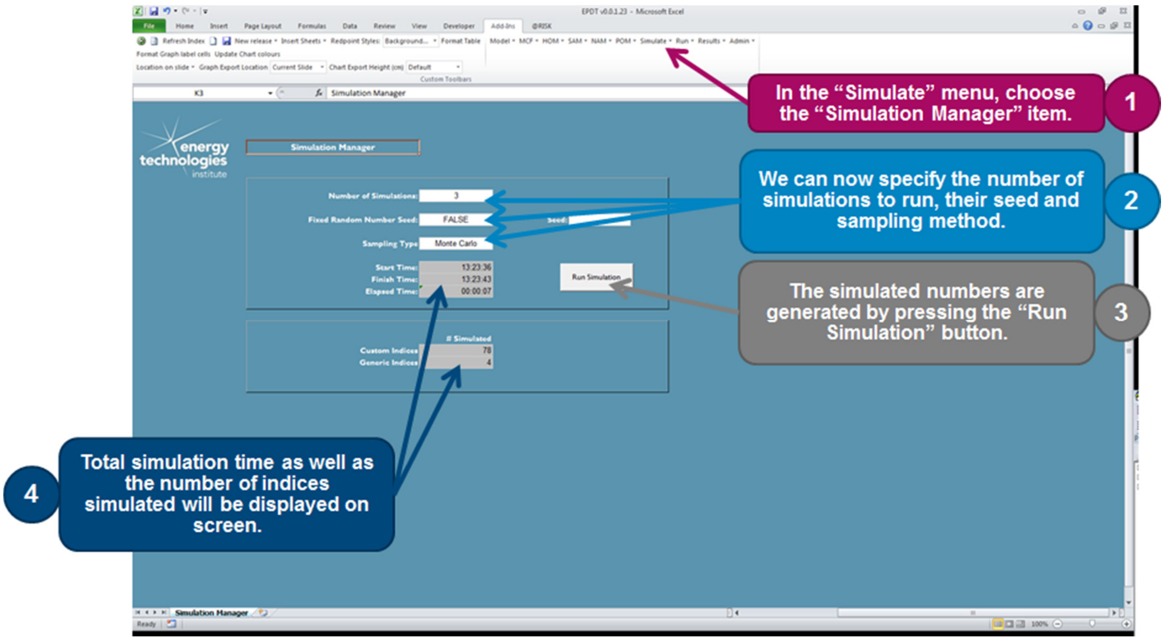


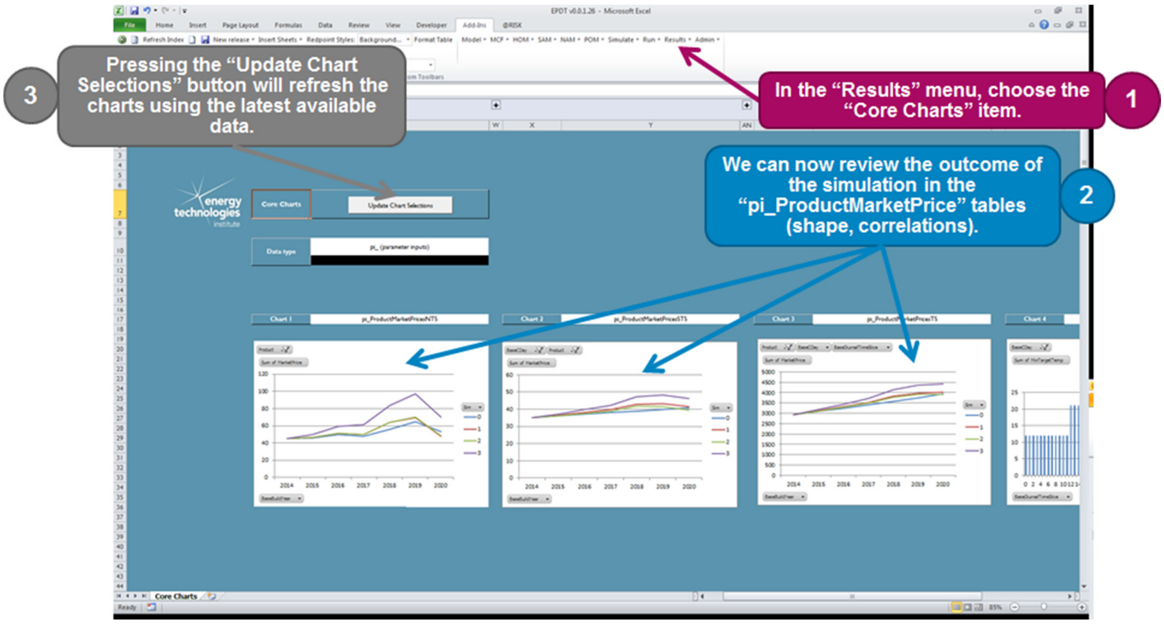
Press the "Commit to database" button in the ribbon to write these changes to the connected database.
4
In the "Simulate" menu, choose the "monte_carlo – parameter data" item and open "pi_GenericIndicesCorrelations"
1
The defined correlations between generic indices will be displayed on screen for review.
2
Let's define correlation numbers for the two new generic indices; the "Status" flag should switch to "New" (dark green)
3


In the "Simulate" menu, choose the "monte_carlo – set data" item and open "si_CustomIndices"
1
Press the "Commit to database" button in the ribbon to write these changes to the connected database.
4
The custom indices for each simulated parameter will be displayed on screen for review.
2
We can define the distribution of a give custom index (e.g. triangular distribution for market price of biomass).
3


In the "Simulate" menu, choose the "monte_carlo – parameter data" item and open "pi_CustomIndicesWeights"
1
Press the "Commit to database" button in the ribbon to write these changes to the connected database.
4
The associated custom and generic indices associated to each simulated parameter will be displayed on screen for review.
2
We can define the weights of the new generic indices for the market price of biomass, electricity, gas and oil products.
3


In the “Simulate” menu, choose the “monte_carlo – parameter data” item and open “pi_GenericIndicesSimSettings”
1
The defined settings of the generic indices’ distributions will be displayed on screen for review.
2
Let’s define min, max and mode numbers for the two new generic indices
3
Press the “Commit to database” button in the ribbon to write these changes to the connected database.
4


In the “Simulate” menu, choose the “Simulation Manager” item.
1
We can now specify the number of simulations to run, their seed and sampling method.
2
The simulated numbers are generated by pressing the “Run Simulation” button.
3
Total simulation time as well as the number of indices simulated will be displayed on screen.
4



9.12. Define temporal aggregation

HOM / SAM / NAM modules create model at granular level in terms of

- Time dimensions (time periods on pathway, characteristic days, diurnal time slices)
- Archetypes, packaged technologies, network options, spatial clusters

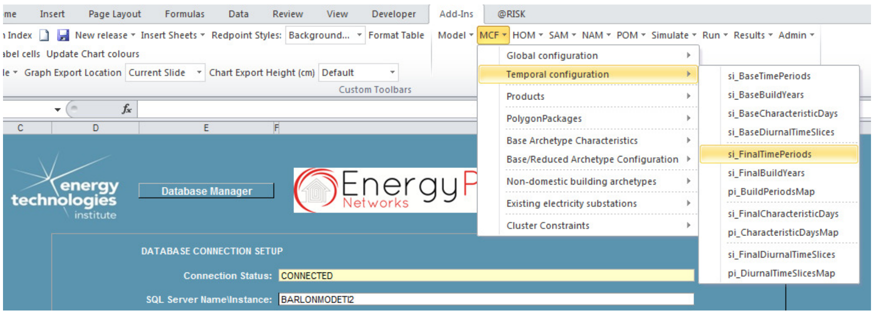
POM pre-processing in primary SQL database creates full dataset for POM

- Aggregates across time dimensions in line as part of pre-processing
- Removes some technology, archetype options according to constraints

Example of aggregating time periods / build years

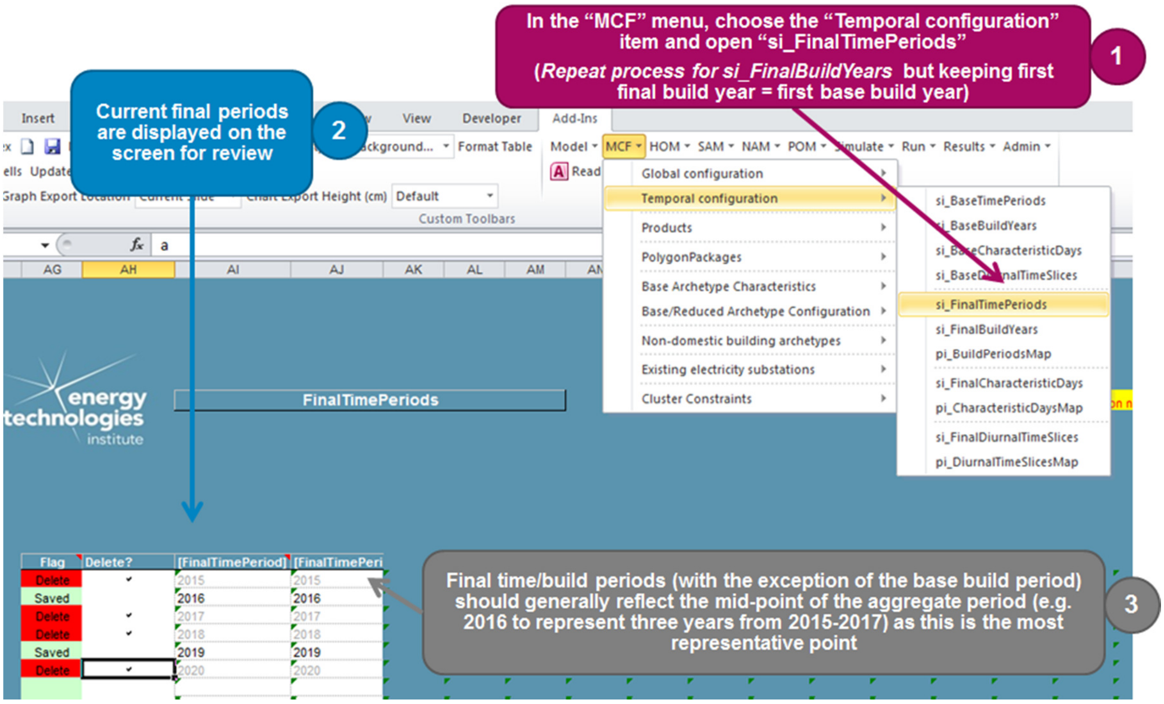
Time period aggregation defined by 3 key tables

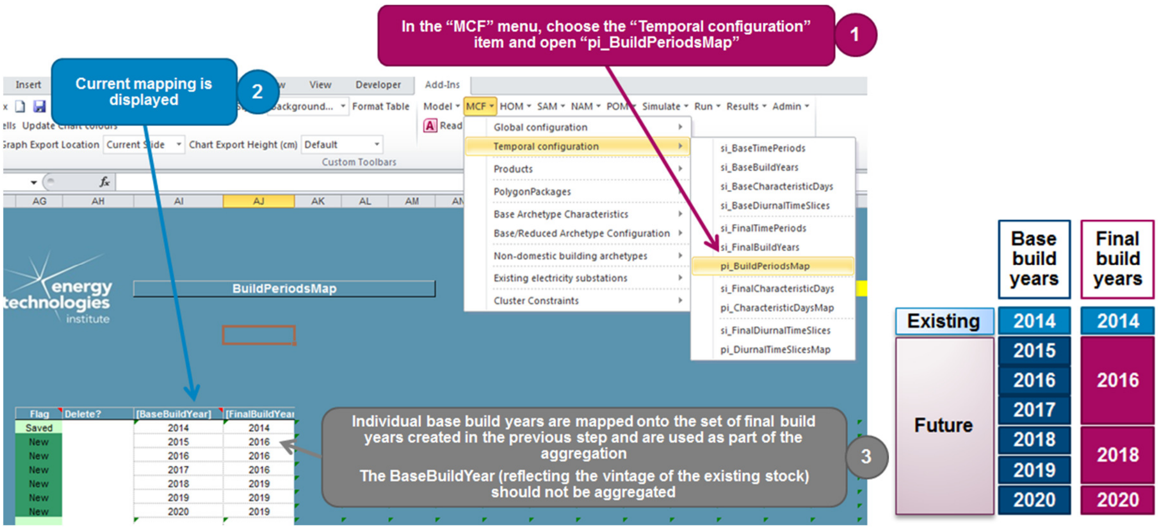
- mcf.si_FinalTimePeriods
  - Nb base time periods represent each annual year on the pathway
- mcf.si_FinalBuildPeriods
  - Identical to above except for "base year" which reflects vintage of the existing stock but is not an "active year" on the pathway
- mcf.pi_BuildPeriodsMap – defines base to final mapping
  - The equivalent TimePeriodMap auto-defined in reference to this

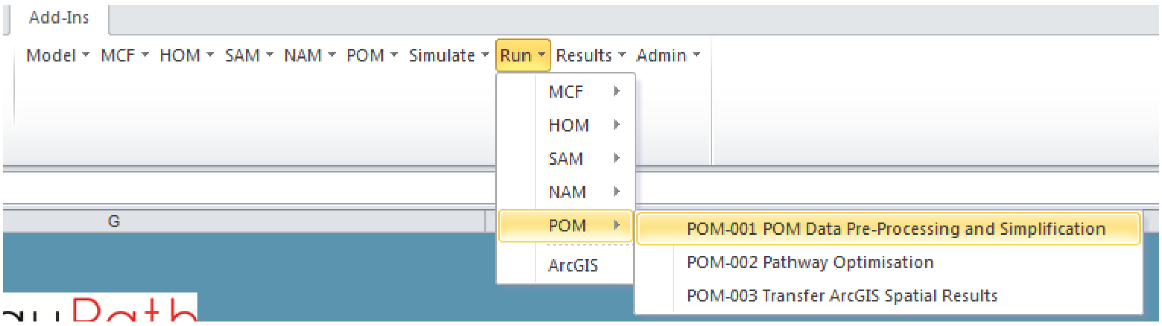


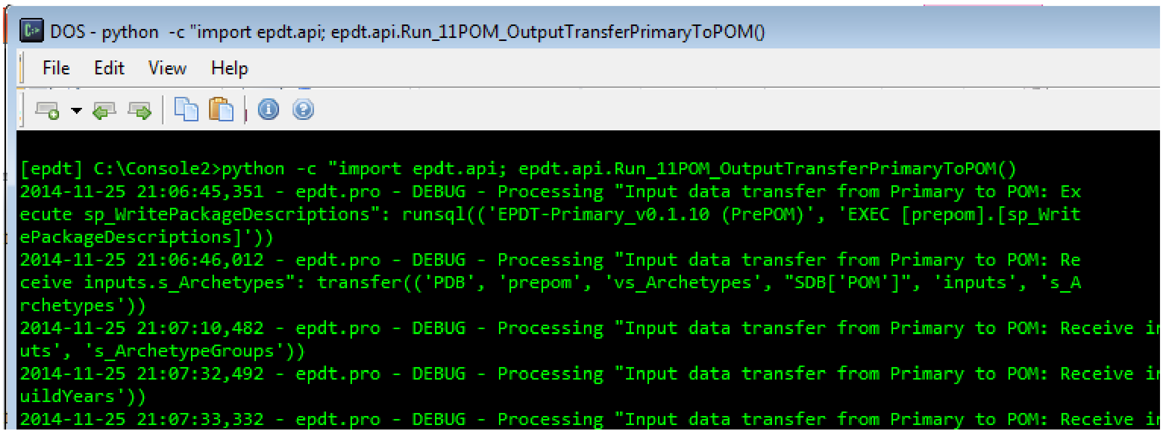
To update the timeperiod / buildperiod aggregation the user needs to:

1. Define updated set of final time periods/build years
2. Update mapping of base to final build periods
3. Run POM pre-processing




In the "MCF" menu, choose the "Temporal configuration" item and open "pi_BuildPeriodsMap"
1
Current mapping is displayed
2
Global configuration
Temporal configuration
Products
PolygonPackages
Base Archetype Characteristics
Base/Reduced Archetype Configuration
Non-domestic building archetypes
Existing electricity substations
Cluster Constraints
si_BaseTimePeriods
si_BaseBuildYears
si_BaseCharacteristicDays
si_BaseDiurnalTimeSlices
si_FinalTimePeriods
si_FinalBuildYears
pi_BuildPeriodsMap
si_FinalCharacteristicDays
pi_CharacteristicDaysMap
si_FinalDiurnalTimeSlices
pi_DiurnalTimeSlicesMap
BuildPeriodsMap
Individual base build years are mapped onto the set of final build years created in the previous step and are used as part of the aggregation
The BaseBuildYear (reflecting the vintage of the existing stock) should not be aggregated
3
Base build years
Final build years
Existing
Future
2014
2015
2016
2017
2018
2019
2020


Add-Ins
Model MCF HOM SAM NAM POM Simulate Run Results Admin
MCF
HOM
SAM
NAM
POM
ArcGIS
POM-001 POM Data Pre-Processing and Simplification
POM-002 Pathway Optimisation
POM-003 Transfer ArcGIS Spatial Results


DOS - python -c "import epdt.api; epdt.api.Run_11POM_OutputTransferPrimaryToPOM()
File Edit View Help
[epdt] C:\Console2>python -c "import epdt.api; epdt.api.Run_11POM_OutputTransferPrimaryToPOM()
2014-11-25 21:06:45,351 - epdt.pro - DEBUG - Processing "Input data transfer from Primary to POM: Execute sp_WritePackageDescriptions": runsql(('EPDT-Primary_v0.1.10 (PrePOM)', 'EXEC [prepom].[sp_WritePackageDescriptions]'))
2014-11-25 21:06:46,012 - epdt.pro - DEBUG - Processing "Input data transfer from Primary to POM: Receive inputs.s_Archetypes": transfer(('PDB', 'prepom', 'vs_Archetypes', "SDB['POM']", 'inputs', 's_Archetypes'))
2014-11-25 21:07:10,482 - epdt.pro - DEBUG - Processing "Input data transfer from Primary to POM: Receive inputs', 's_ArchetypeGroups'))
2014-11-25 21:07:32,492 - epdt.pro - DEBUG - Processing "Input data transfer from Primary to POM: Receive ildYears'))
2014-11-25 21:07:33,332 - epdt.pro - DEBUG - Processing "Input data transfer from Primary to POM: Receive i

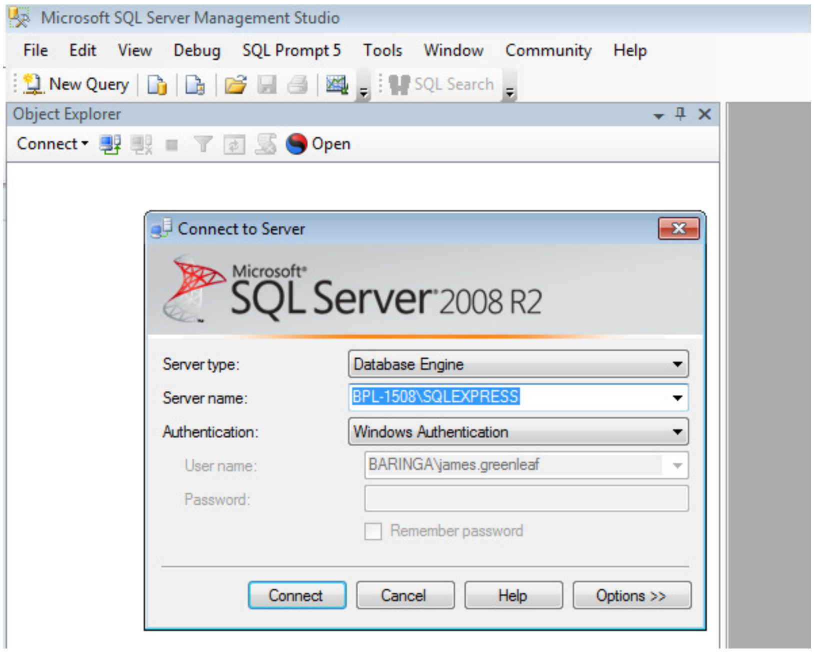
9.13. Generic use cases

9.13.1. Basic database configuration and management

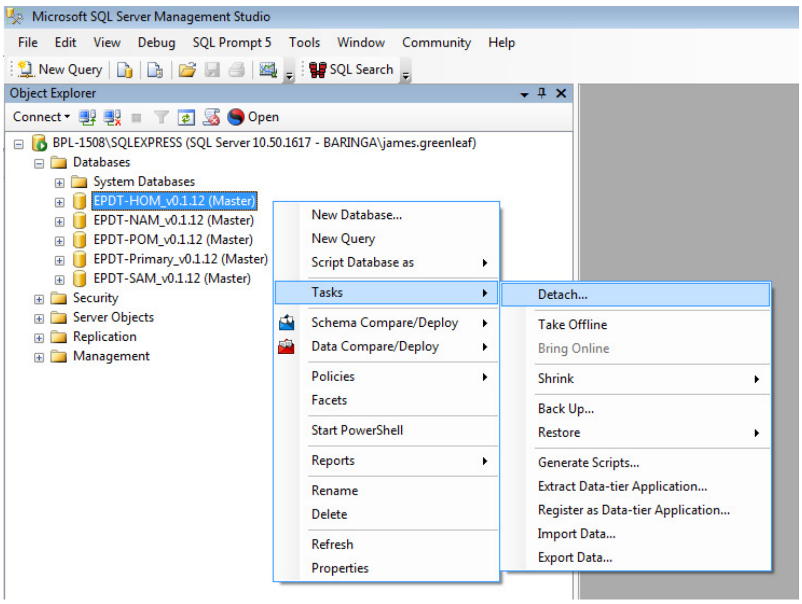
This case describes the basic process of attaching / detaching a database, creating a copy for backup and connection of the Excel UI.

Attaching / detaching SQL databases

First run “Microsoft SQL Server Management Studio and “Connect” to the appropriate server name that was configured as part of the installation of the EPN tool in 2.



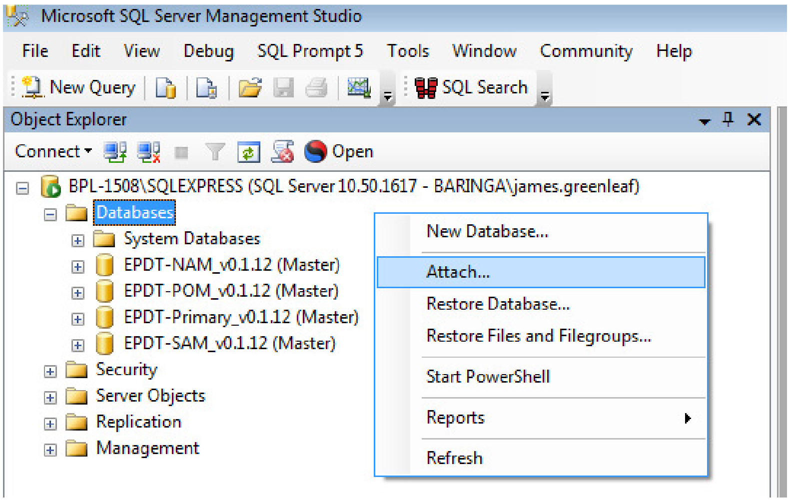
-
- To detach an existing database right click on the database name in the *“Object Explorer”* tree window and select *Tasks -> Detach.*



–

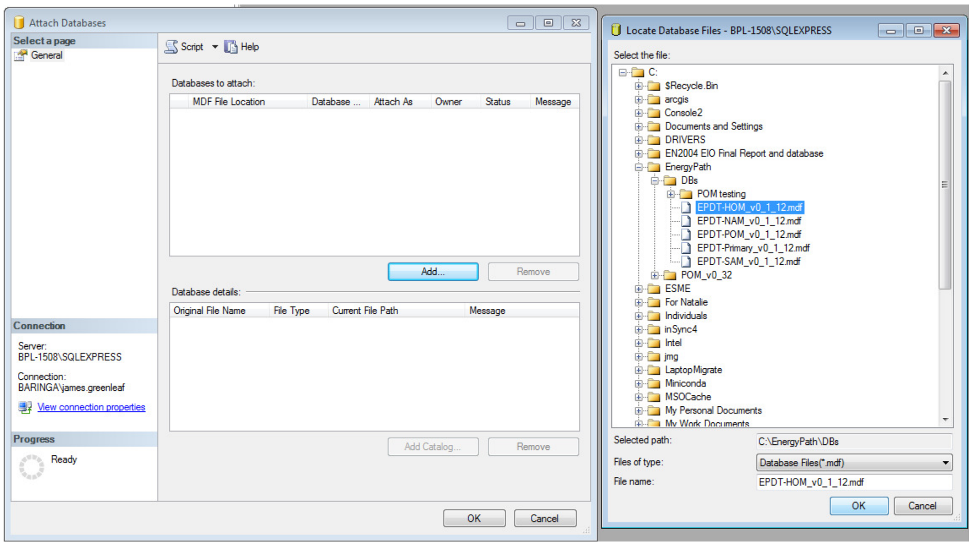
- In the subsequent pop-up window click ok.
- Note that it is necessary for a database to be detached to make a copy – e.g. for the purpose of archiving it (both the *.mdf and *.ldf files must be copied)

- To attach a database right click on the “Databases” section of the tree and select *Attach*



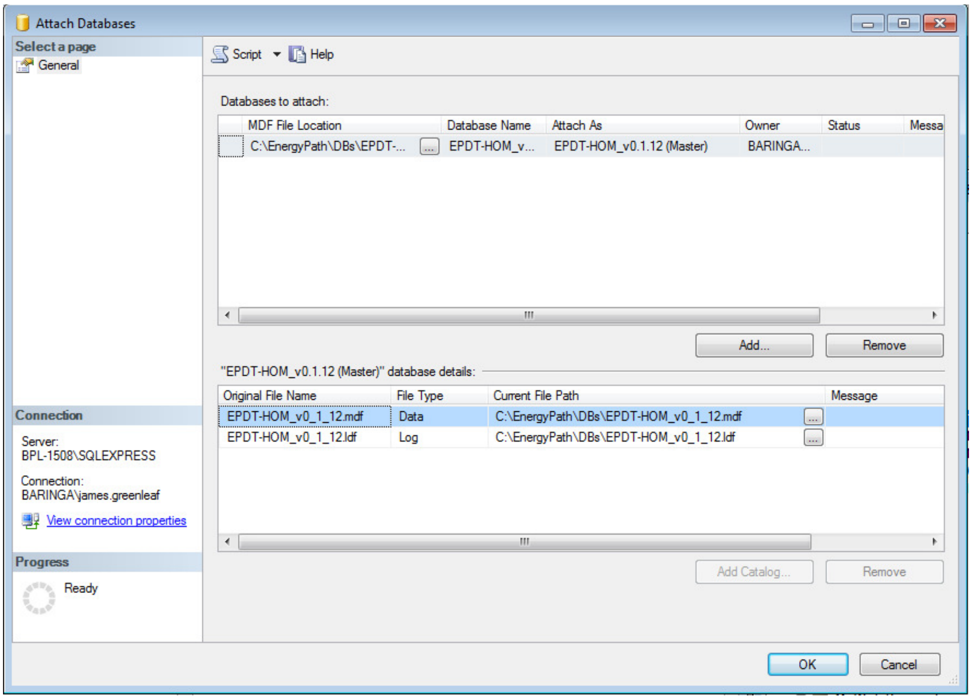
–

- In the subsequent pop-up window select *Add* -> locate and click on the relevant *.mdf database file on the network or local drive -> Select OK



–

- The original pop-up menu is now updated with details of the selected database to attach

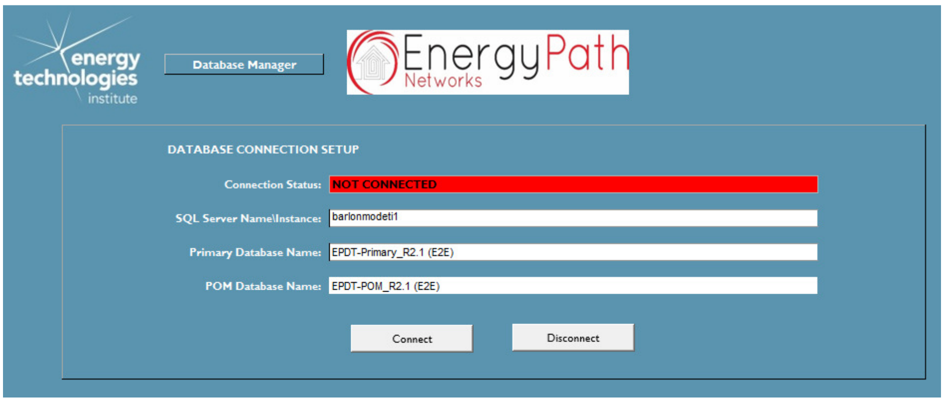


–

– Select *OK* and this database will appear in the list of databases in the “Object Explorer” tree.

Connecting / re-connecting the Excel UI

As described in section 6.1.10 the Excel UI is ***only*** connected to the ***Primary Database*** and not the secondary HOM, SAM, NAM, POM or supplementary SINCAL database. As described in this section the Excel UI is connected via the [Model] -> [Database Manager] sub-menu item



To connect to a different primary database simply update the “Database Name” box and click *Disconnect* and then *Connect* buttons. The “Connection Status” box will turn yellow and state “CONNECTED” if this is successful (save the Excel UI to retain this setting next time it is re-opened). The Excel UI does ***not*** need to be closed when attaching or detaching databases in SQL Server Management Studio.

10. Appendix – Database Documentation

This section provides a description of each primary database table extracted to html format.

EPN Primary DB.html

11. Appendix – ER Diagrams for Physical Data Model

These have been created using the *SQL Dependency Tracker* from RedGate[37], both the underlying *.sdvproj file and an exported PDF are attached.

11.1. Primary database

EPN PDB 2017-02-02.sdvproj

EPN PDB 2017-02-02.pdf

11.2. HOM Secondary database

EPN HOM DB 2017-02-02.sdvproj

EPN HOM DB 2017-02-02.pdf

11.3. SAM Secondary database

EPN SAM DB 2017-02-02.sdvproj

EPN SAM DB 2017-02-02.pdf

11.4. NAM Secondary database

EPN NAM DB 2017-02-02.sdvproj

EPN NAM DB 2017-02-02.pdf

11.5. POM Secondary database

EPN POM DB 2017-02-02.sdvproj

EPN POM DB 2017-02-02.pdf

[37] http://www.red-gate.com/products/sql-development/sql-dependency-tracker/

12. Appendix – Code documentation

The following sections provide the extracted comments for each sub-module.

12.1. MCF / HOM / NAM / SAM – python API code descriptions

The python code documentation is automatically extracted into html format. The files are located in the embedded zip, by running ***index.html***.

EPN API Descriptions.rar

12.2. POM – AIMMS identifier Descriptions

This section provides a description of each identifier in the full formulation of the POM AIMMS model, extracted to a text file.

POM AIMMS Identifier Descriptions.txt