

# The Database

XPSWMM maintains an internal database that integrates the spatial data associated with an object with the attribute data required by the model engine.

The graphical network creation and manipulation can be considered as a specialized function dealing with the purely graphical attributes of objects, such as; display symbols, spatial coordinates, and connectivity.

The method of creating the model-specific attribute data is through the [Dialog Box](#) of one of the many available import methods.

## The Permanent Database

The data managed by XP is permanently stored on disk in a "database" file, as an ordinary operating system file. This file should normally have an ".XP" extension. Data is only committed to this file during a Save command.

The database stores both the graphical and non-graphical attributes of all objects in the network, and also the non-specific or general control data associated with a network, such as job title and time steps for the solution procedure, mode of analysis, links to external interface files etc.

When using the Scenario Manager all of the model differences that create scenarios are stored in a Microsoft Access .MDB file. When sharing the model between users it will be necessary to give both the .XP and .MDB files.

The following table describes how databases change and modifications occur.

Application Files	Description
Model.xp	The XP file is the primary XPS model database.
Model.json	This the primary file for global rainfall input, ensembles, and runoff results.
Model.mdb	This file stores the Scenario settings and stores the differences between the Base Scenario and all child scenarios.
Model.sqlite	This database contains Sewershed information.

Each file is created upon opening the file, and is populated by Save or Auto Save at designated time intervals.

## The Working Database

To increase the size of networks able to be efficiently created and manipulated by XP, the program utilises a combination of memory and disk space to manage data.

In editing sessions, the Permanent Database stored on the disk is not interacted with directly. All changes made are done to an internal working copy of the Permanent Database, known as the Working Database which is stored in memory.

The Working Database is established when the Permanent Database is opened. The Permanent Database is updated only when the Working Database is "saved". Copies of the Working Database can be saved under different names at any time, the default name being the name of the Permanent Database when originally "opened".

The Working Database is the active database to which all data editing changes are made. There can only be one active database at any time.

Using the working database provides an error recovery procedure. Since almost all changes to the database are immediately recorded in the working database, catastrophic failures can be recovered from with a minimum of agony. Additionally, if you would like to revert to the permanent database and disregard changes made since the last save, choose Revert from the File pull-down menu.

Additional information on file types may be found in [File Extensions](#).

## Database Integrity

As far as possible, data committed to the Working Database via the Dialog Box interface is checked and filtered to maintain the integrity and consistency of the database.

**On this page:**

**On this section:**

- [The Model Structure](#)
- [Philosophy](#)
- [Strategy](#)
- [Is a 2D or 1D-2D Model Feasible](#)
- [Analytical Engine](#)
- [The Database](#)
- [Expert System Capabilities](#)

In general, text strings entered by users are the most dangerous type of dialog item because they are a human-readable encoding of some fundamental data type such as a number. This encoding and decoding to and from text strings done by users and computers is a rich source of errors. XP checks all strings at three levels to ensure they can be interpreted correctly.

1. Absolute Validity
2. A numerical string, for example, cannot contain invalid non-numeric characters.
3. Absolute Range Checking – Outside of range generates an error
4. Once a numerical string can be interpreted properly, its value is checked for validity in the context of the model, eg. A negative pipe length would not be accepted.
5. "Reasonable Range" Checking – Outside of range generates a warning
6. If a data item is within the absolute range it is also compared to a reasonable range defined by the Expert Engineer. If the data is between the absolute and reasonable range, a warning is generated and the user is asked to confirm that the data entered is correct and is not a typographical error. The user can ignore the warning and proceed to enter more data. If the data is outside of the absolute range, an error is generated and the user is required to fix the offending input before committing the data with the OK button on the dialog and continuing to enter additional data.

These checks are applied to individual data items and are classed as low-level consistency checks. High-level consistency checks are also made, which generally involve placing constraints on relationships between data items. Because these checks rely on the existence of independent data items, often there is insufficient data for XP to perform these high-level checks interactively. These integrity checks are therefore performed off-line at the point of generating the data file for the model to solve.