

Scenario Management (Part 1)

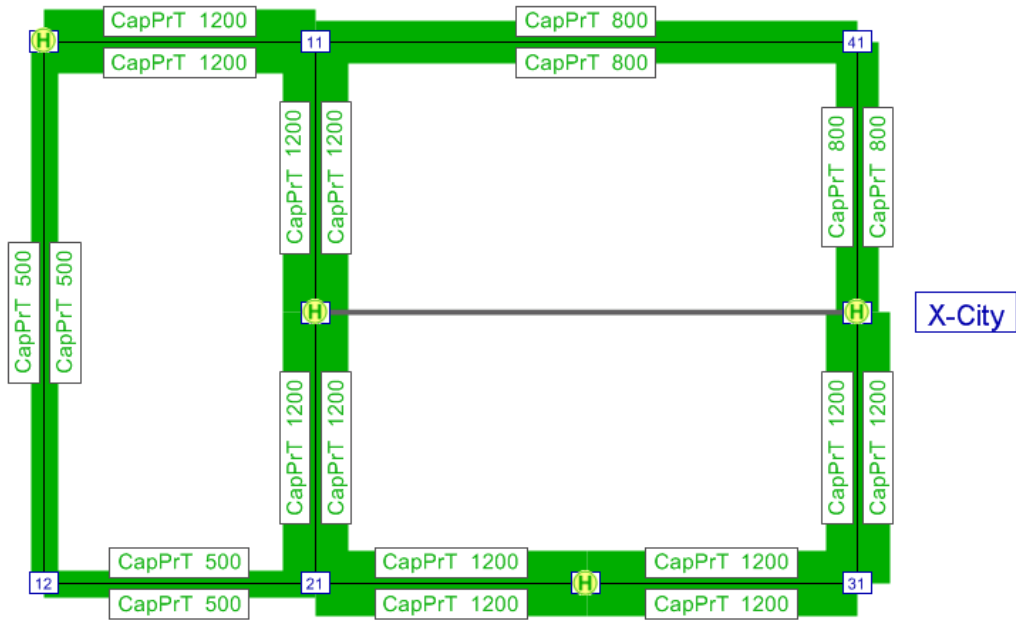
Many users requested a more explicit support for the data management of several scenarios within one project. While most users sooner or later develop their own workflow for maintaining related scenarios, and may automate it to varying degrees, VISUM can and should take care of most of it. We are currently finishing the design of a scenario manager inside VISUM which will:

- ▶ organise variants of a model (related to input data for both supply and demand) in a non-redundant way,
- ▶ let users define scenarios as combinations of model blocks building on each other,
- ▶ allow batch runs of the model for some or all of its defined scenarios,
- ▶ support scenario comparison by cross-tabulating selected network-wide measures of effectiveness.

The package as a whole is too large to be implemented within a single release, so that we will deliver it in two stages, beginning with VISUM 11.5. The first part focuses on the introduction of model-transfer files, a file format which captures the difference between two scenarios of a model in a single ASCII file.

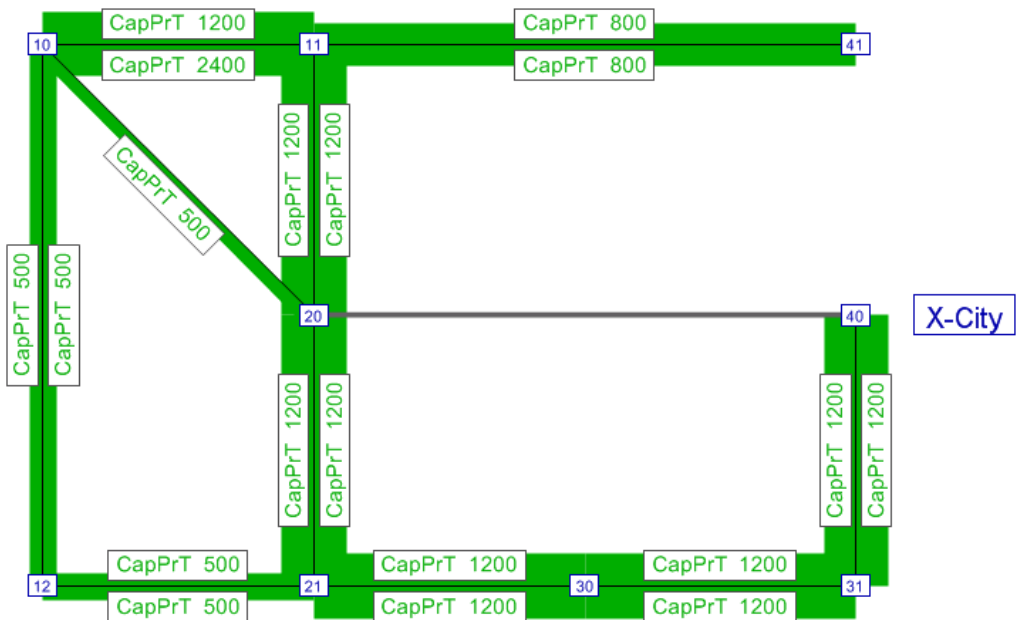
Let us assume that the base case of our model is the network shown below (stored in file 1.ver):

A-Village



In the course of a project we develop a new scenario 2.ver that looks like this:

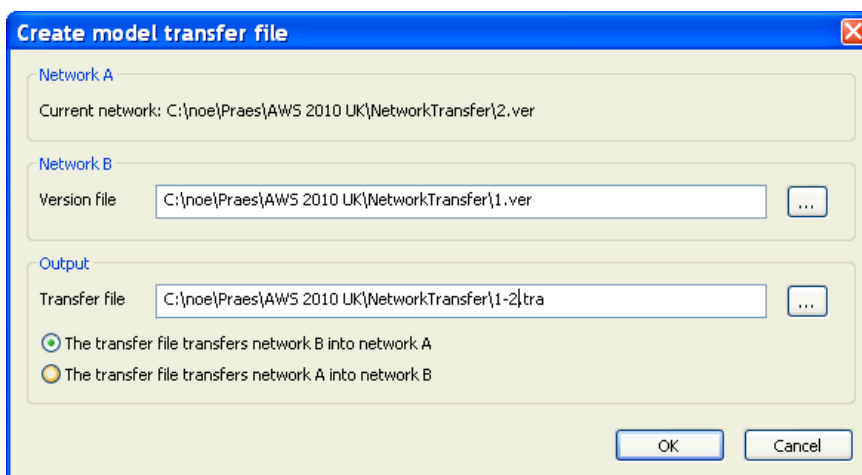
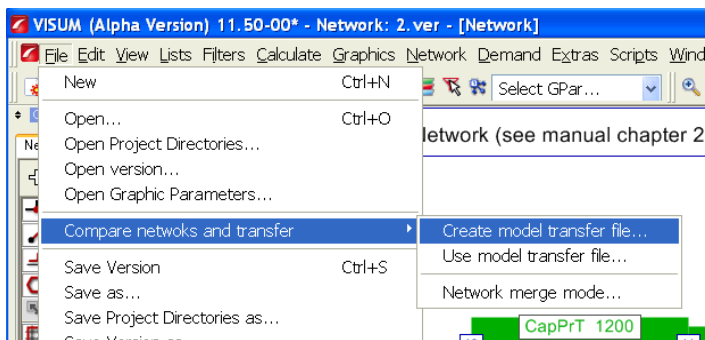
A-Village



A quick glance tells us that a new link was added between nodes 10 and 20, the link 10 – 11 changed its capacity and the link 41 – 40 was deleted. Storing the two scenarios as separate

version files is redundant, because much of the information, e.g. everything about the lower half of the network is duplicated. If you discovered a mistake in that part of the base version, you would need to fix it in each scenario version file, too. Instead we would like to capture the difference between scenarios 1 (= base) and 2 in a file.

The new menu item *Create model transfer file ...* opens a dialog where you select the two model snapshots to be compared.



It then creates a model transfer file (extension .tra) which when applied to B (1.ver) will produce A (2.ver).

The model transfer file is an ASCII file which uses the familiar syntax of .net and .dmd files, with two important new features. Firstly, all tables from .net and .dmd files can be mixed freely in one model transfer file (effectively ending the separation between the two file types). Secondly, unlike a .net file in additive mode, a model transfer file can add, change, and delete network objects. Below is the model transfer file 1-2.tra:

```

$VISION
$VERSION:VERSNR;FILETYPE;LANGUAGE;UNIT
8.100;Trans;ENG;KM

*
*
* Table: Links (Removed)
$-LINK:NO;FROMNODENO;TONODENO
9;40;41

*
*
* Table: Links (Edited)
$*LINK:NO;FROMNODENO;TONODENO;CAPPRT;FROMNODEORIENTATI...
1;10;11;2400;;
10;10;12;;W;
10;12;10;;;W

*
*
* Table: Links (Created)
$+LINK:NO;FROMNODENO;TONODENO;NAME;TYPENO;TSYSSET;USER...
12;10;20;;0;C;0;7.071;1;0;500;4;360000000s;360000000s;...
12;20;10;;0;W;0;7.071;1;0;0;4;360000000s;360000000s;63...

*
*
* Table: Turns (Created)
$+TURN:FROMNODENO;VIANODENO;TONODENO;TYPENO;TSYSSET;CA...
11;10;20;3;B,C,H,T;99999;0s;0;0;0;1;3,66;0;0;4.000;0;3...
11;20;10;1;B,C,H,T;99999;0s;0;0;0;1;3,66;0;0;4.000;0;3...
21;20;10;3;B,C,H,T;99999;0s;0;0;0;1;3,66;0;0;4.000;0;3...
...

```

After a header the file contains three link tables. The first one (\$-LINK) lists the keys of all links which are deleted, the second one (\$*LINK) lists changes to existing links, and the third one (\$+LINK) lists all new links. There is a further table at the bottom which lists new turns (created as a byproduct of new link 10-20). Splitting the link table into three parts has important benefits. The complete list of attributes is only necessary in \$+LINK, saving space and improving readability. And when we later apply the model transfer file, we can check additional constraints: link 10-20 should not exist already, because we want to add a new link with that key. Conversely, link 41-40 should exist, because we expect to delete it. Otherwise

we would flag an error, because chances are that the model transfer file is applied to the wrong base version.

After saving the model transfer file we can delete file 2.ver. If we need the model input for scenario 2, we can instead load 1.ver, then apply 1-2.tra to it using the new menu item *Use model transfer file ...*

In the past we recommended a super-network approach to (do-it-yourself) scenario management, where a single network contained data for all scenarios, and parts of the network were disabled before the model was run. Model transfer files, by comparison, support the exact opposite. Only the minimal base scenario exists as a version file, everything else is stored as small deltas which are loaded on demand as model transfer files. Multiple model transfer files, for both supply and demand, can be applied in succession before a model run, and they will all take effect – as long as these effects do not overlap. (A few additional restrictions apply where changes to one network objects cascade to other network objects, e.g. in public transport or in the junction model, but the general principle still holds.)

In the next release after VISUM 11.5 we will wrap the model transfer file concept with a user-friendly GUI. You will be able to graphically organize scenarios in a tree rooted at the base scenario, and activate scenarios by clicking a node in the tree. VISUM then takes care of the necessary file load operations relieving you of managing model transfer files on disk. From the scenario tree you will be able to run the model for a selected subset of all scenarios and generate comparative reports for the model results.

Part 1 of scenario management also includes a first step for the comparison of scenario results (output). With a new operation you will be able to pull selected attributes from one or several version files (containing results for one scenario each) into one master version file. VISUM will create UDAs for these values in the master version file which take their name from the name of the original attribute plus a code identifying the scenario. After this operation you could e.g. have link attributes `VolVehPrT_Build1(AP)`, `VolVehPrT_Build2(AP)`, `VolVehPrT_Build3(AP)` in the master version file. Assume that the master version file contains the results of the NoBuild scenario. You can then immediately display comparative listings or plots of the four (including NoBuild) sets of link volumes. VISUM also predefines attributes for the absolute or relative difference between the master version attribute and any of the imported attributes, so the value of

$$(\text{VolVehPrT_Build1(AP)} - \text{VolVehPrT (AP)}) / \text{VolVehPrT (AP)}$$

is also directly available in listings or network graphics without any need for multi-edit. Compared to VISUM 11's difference network mode, the new operation does not place VISUM in a special mode in which many operations are banned.