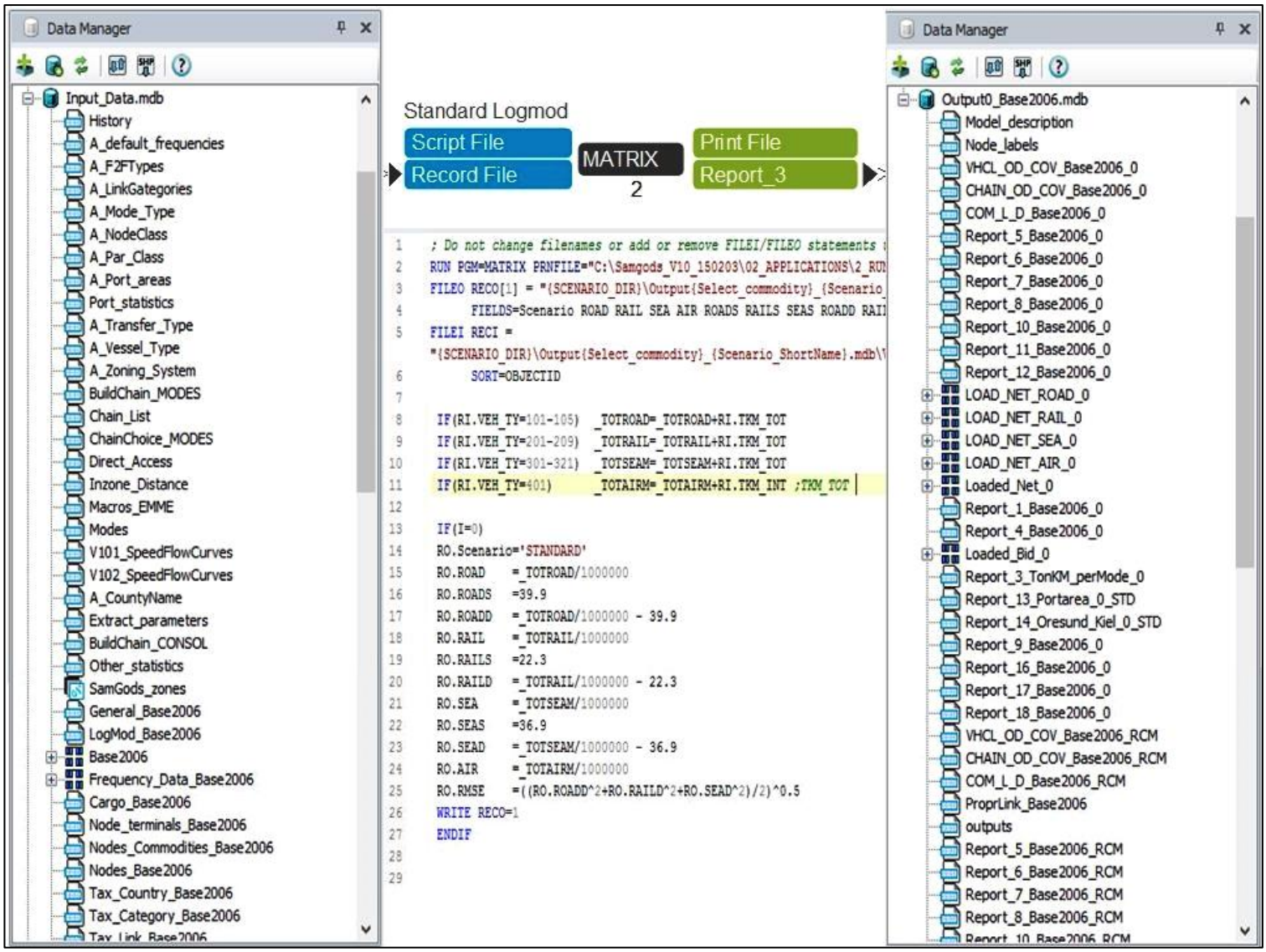
TRAFIKVERKET

REPORT

Samgods Technical documentation V1.0



TMALL 0004 Rapport generell v 1.0

Document title: Samgods Technical documentation V1.0
Created by: Gabriella Sala, Daniele Romanò, see list of contributors
Document type: Report
Case number: 2014/26249

Date of publication: 2015-04-01
Publisher: Trafikverket (The Swedish Transport Administration)
Contact person: Petter Hill
Responsible: Peo Nordlöf
Distributor: Trafikverket (The Swedish Transport Administration)

Content

INTRODUCTION 7

GLOSSARY 8

THE FILE STRUCTURE 9

INPUT AND OUTPUT FILE REFERENCE 16

File summary 16

2.1.1.1. A_CountyName 23
2.1.1.2. A_default_frequencies 23
2.1.1.3. A_F2FTypes 24
2.1.1.4. A_LinkCategories 24
2.1.1.5. A_Mode_Type 25
2.1.1.6. A_NodeClass 25
2.1.1.7. A_Par_Class 26
2.1.1.8. A_Port_areas 27
2.1.1.9. A_Transfer_Type 27
2.1.1.10. A_Vessel_type 28
2.1.1.11. A_Zoning_System 28
2.1.1.12. BuildChain_CONSOL 29
2.1.1.13. BuildChain_MODES 30
2.1.1.14. Chain_List 31
2.1.1.15. ChainChoice_MODES 31
2.1.1.16. Direct_Access 32
2.1.1.17. Extract_parameters 33
2.1.1.18. Inzone_Distance 34
2.1.1.19. Macros_EMME 35
2.1.1.20. Modes 36
2.1.1.21. Other_statistics 36
2.1.1.22. Port_statistics 37
2.1.1.23. SamGods_zones 38
2.1.1.24. V101_SpeedFlowCurves 39
2.1.1.25. V102_SpeedFlowCurves 40
2.1.2. Base data 41
2.1.2.1. Base2006 44
2.1.2.2. Frequency_data_Base2006 45
2.1.2.3. Cargo_Base2006 48
2.1.2.4. Node_terminals_Base2006 49
2.1.2.5. Nodes_Base2006 50
2.1.2.6. Nodes_Commodities_Base2006 51
2.1.2.7. PropLink_Base2006 54
2.1.2.8. Rail_Capacity_Base2006 54
2.1.2.9. Tax_Category_Base2006 55
2.1.2.10. Tax_Country_Base2006 56
2.1.2.11. Tax_Link_Base2006 56

2.1.2.12. Toll_Link_Base2006 ... 57
2.1.2.13. Vehicles_exception_Base2006 ... 58
2.1.2.14. Vehicles_parameters_Base2006 ... 59
2.1.3. Scenario specific data ... 60
2.1.3.1. General_{SCENARIO_SHORTNAME} ... 61
2.1.3.2. LogMod_{SCENARIO_SHORTNAME} ... 62
2.1.4. Manage database ... 64
2.1.4.1. History ... 64
2.1.4.2. Model_description ... 64

Calibration data ... 65
Port Area parameters per STAN group ... 65
Parameters_Kielcalibration.dbf ... 65
Parameters_portcalibration.dbf ... 66

Output data ... 66
General tables in geodatabase ... 67
Model description ... 67
Outputs ... 67
Node_labels ... 68
ProprLink_{SCENARIO_SHORTNAME} ... 68
Network format in geodatabase ... 70
Load_net_Road_0 and Load_net_Road_RCM ... 70
Load_net_Rail_0 ... 72
Load_net_Sea_0 and Load_net_Sea_RCM ... 75
Load_net_Air_0 and Load_net_Air_RCM ... 79
Loaded_net_0 ... 81
Loaded_Bid_0 and Loaded_Bid_RCM ... 88
COMPARE_LOAD0_{Scenario_short_name}_{Scenario_for_comparison) and COMPARE_LOADRCM ... 89
Load_net_Rail_RCM ... 91
Load_net_RCM ... 94
Rail_BiDir ... 101
Compare_Bid ... 104
Table format in geodatabase ... 107
CHAIN_OD_COV_Base2006{SCENARIO_SHORTNAME}_0 and CHAIN_OD_COV_{SCENARIO_SHORTNAME}_RCM ... 107
COM_L_D_{SCENARIO_SHORTNAME}_0 and COM_L_D_{SCENARIO_SHORTNAME}_RCM ... 108
VHCL_OD_COV_{SCENARIO_SHORTNAME}_0 and VHCL_OD_COV_{SCENARIO_SHORTNAME}_RCM ... 109
Report_1_{SCENARIO_SHORTNAME}_0 and Report_1_{SCENARIO_SHORTNAME}_RCM ... 110
Report_3_TonKM_perMode_0 and Report_3_TonKM_perMode_RCM ... 111
Report_4_{SCENARIO_SHORTNAME}_0 and Report_4_{SCENARIO_SHORTNAME}_RCM ... 113
Report_5_{SCENARIO_SHORTNAME}_0 and Report_5_{SCENARIO_SHORTNAME}_RCM ... 114
Report_6_{SCENARIO_SHORTNAME}_0 and Report_6_{SCENARIO_SHORTNAME}_RCM ... 115
Report_7_{SCENARIO_SHORTNAME}_0 and Report_7_{SCENARIO_SHORTNAME}_RCM ... 116
Report_8_{SCENARIO_SHORTNAME}_0 and Report_8_{SCENARIO_SHORTNAME}_RCM ... 117
Report_9_{SCENARIO_SHORTNAME}_0 and Report_9_{SCENARIO_SHORTNAME}_RCM ... 118
Report_10_{SCENARIO_SHORTNAME}_0 and Report_10_{SCENARIO_SHORTNAME}_RCM ... 119
Report_11_{SCENARIO_SHORTNAME}_0 and Report_11_{SCENARIO_SHORTNAME}_RCM ... 120
Report_12_{SCENARIO_SHORTNAME}_0 and Report_12_{SCENARIO_SHORTNAME}_RCM ... 121
Report_13_Portarea_0_STD and Report_13_Portarea_RCM ... 122
Report_14_Oresund_Kiel_0_STD and Report_14_Oresund_Kiel_RCM ... 123

Report_15_RailLinksRCM 124
Report_16_{SCENARIO_SHORTNAME}_0 and Report_16_{SCENARIO_SHORTNAME}_RCM 125
Report_17_{SCENARIO_SHORTNAME}_0 and Report_17_{SCENARIO_SHORTNAME}_RCM 126
Report_18_{SCENARIO_SHORTNAME}_0 and Report_18_{SCENARIO_SHORTNAME}_RCM 128
Results in matrix format 130
LOS matrices 130
Trip Matrices from Standard Logistic Module 137
Trip Matrices from Rail Capacity Management 144
Other matrices 151
Other output related to matrices 152
Report files 152
Temporary data to visualize scenario data 152
LogMod folder 153
Other files with different purposes 158
Calibration procedure files 158
Exported files 158
Reports from STD Logistics Module and Rail Capacity Management Module 159
Warm start procedure 159
Debugging 160

THE MODEL STRUCTURE 163

THE APPLICATION STRUCTURE 175

"Installation" application 175

"Create the editable files" application 175

"Edit the data" application 176

"SamGods" application 180

"Compare scenarios" application 200

"Handling scenario" 200

"PWC_matrices" application 203

"Change matrix format" application 203

APPLIED METHODS IN THE MODEL 204

Manage different loading times, costs and capacities for different commodities 204

Manage different consolidation factors for different commodity groups 205

Capacity constraints on sea mode 208
Capacity constrains in Kiel Canal 208
Capacity constrains in Vänern Lake 210

Rail Capacity Management procedure 211

LP0 step 212
LP1 step 214
Adjust Capacity Procedure 216
Final process (FIN) 216

Rail Assignment 217
Capacity constraints 217
Domestic flows allowed only on domestic links 218

Calibration procedures 219
Forbidden over-seas ship transports to domestic small ports 219
Rescaling factors for LOS matrices in sea mode 220
Rescaling factor for Kiel Canal 221

Samgods Parallelization Module 222

PROGRAMS AND LICENSES REQUIRED 224

Programs 224
Cube software 224
JAVA software 224
LP Solve 224

Licence requirements 225
5.1.1. Cube software 225
5.1.2. JAVA software 225
5.1.3. LP solve 225

REFERENCES 226

Introduction

This technical document is written for Trafikverket in order to help users to perform analysis with the Samgods GUI. It contains a technical description of each element of the model in terms of input data and output data, procedures and methods applied.

The Samgods GUI forms an integrated interface for several external programs developed along the years. They are related to the Standard Logistics Module and Rail Capacity Management Module. For details on the external programs, their control files and their usage and meaning of input and output, we refer to their own technical documentation listed in chapter 7. The focus is on the logistics model for Sweden. The documentation is not written for programmers in the first place, but mainly for users of the model.

The document has the following structure:

- **Chapter 1**: the file structure
- **Chapter 2**: input and output files reference
- **Chapter 3**: description of model structure
- **Chapter 4:** description of applications
- **Chapter 5**: Method applied in the model
- **Chapter 6**: License required
- **Chapter 7**: References

List of contributors

Gabriella Sala	Citilabs	Developer of GUI and model setup and co-author of the program documentation
Daniele Romanò	Citilabs	Co-author of the program documentation
Henrik Edwards	Sweco	Developer, tester and specialist of the logistics module and RCM module
Petter Hill	The Swedish Transport Administrator	Project Manager

Glossary

Below some important glossaries used in this manual are collected. Observe that for some glossaries the explanation is specific for this context.

BuildChain - Build Chain procedure associated to BuildCHain.exe program

ChainChoi - Chain Choi procedure associated to CHainChoi.exe program

consol - consolidation

Extract - extraction procedure of tonnes, empty and loaded vehicles matrices associated to extract.exe program

LogMod or STD - Standard Logistics Module (as per version 0.8)

MAT - extension for matrix file produced in Cube (binary format), referred also as Voyager Matrix.

MergeRep - procedure to merge all .rep files produced by ChanChoi in a unique .rep file. Associate to Mergerep.exe program

Mode - aggregate mode used in chains. See table A-2 3 for further details

MSA algorithm - Method of Successive Averages algorithm

ORIGCAP - Initial capacity in rail links provided with input capacity table

PWC matrices - Production Warehouse Consumption matrices

S/A user - Standard User / Advance User**.**

RCM - Rail Capacity Management Module

Submode - submode (vehicle type) used in mode. See table A-2 3 for further details

SCBSTANN - region code for Swedish zones and univocal number for the other countries.

Vessel or Vessel Type - vehicle type for sea mode

Vehicle type - type of vehicle described in the model. 5 types for road mode, 8 types for rail mode, 21 types for sea mode and one for air.

VY - Voyager

Warm start - start the model from a defined LP solution instead of start from scratch.

314 - extension for matrix file in Emme format, e.g. format derived by EMME software

{CATALOG_DIR} - the directory where the .cat file is located

{SCENARIO_DIR} - the directory where the scenario folder is located

{SCENARIO_CODE} - the code associated to a specific scenario and specified during the creation under scenario window

{SCENARIO_SHORTNAME} - the scenario name under the scenario tree present in scenario window

V/C – flow volume over capacity ratio for rail flows.

The file structure

The file structure for the model is shown in the following table:

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
C:\SamGods_v10	SamGods.cat				Catalog file: main file to access to the entire model
C:\SamGods_v10	01_Programs				Folder for the external programs (LOGMOD and RCM). At this level present Samgods_Parallelization_Module.jar and its input files. The other external programs reside in subfolders.
C:\SamGods_v10	01_Programs	LogMod			Folder and subfolders with executables for different programs. Executables are placed in respective subfolders together with control files. The files in LogMod folder are: • execution_times_LogMod.txt - run time associated to each commodity. Used in parallelize LogMod run • execution_times_extract.txt - ID per each vehicle class. Used to parallelize extract procedure • Samgods_Parallelization_Module.jar - program to parallelize LogMod and extract procedure
C:\SamGods_v10	01_Programs	LogMod	BUILDCHAIN		Folder containing executable files: • BuildChain.exe - First program of Standard Logistics Module • BuildChain4RCM.exe - First program of RCM for generation of transport chain alternatives with derived marginal costs for railway and control file: • select.dat for selection of detailed cost and volume output. The following

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					control files: • BuildChain_common.ctl • BuildChainNN.ctl • logselect.dat are produced by the model and saved under scenario folder. Therefore they are not present in the general folder.
C:\SamGods_v10	01_Programs	LogMod	ChainChoi		Folder containing executable files: • ChainChoi.exe - Second program of Standard Logistics Module • ChainChoi4RCM.exe - Second program of RCM for generation of the optimal transport chain alternatives with derived marginal costs for railway • LP2CC.exe - Last program in RCM to derive the final solution in Standard Logistics module format and a control file: • Select.dat for selection of detailed cost and volume output. The following control files: • ChainChoi_common.ctl • ChainChoiNN.ctl are produced by the model and saved under scenario folder. Therefore they are not present in the general folder
C:\SamGods_v10	01_Programs	LogMod	EXTRACT		Folder containing the executable file extract.exe file to extract OD vehicles, empty vehicles and tonnes in EMME format. are produced by the model and saved under scenario folder. Therefore they are not present in the general folder
C:\SamGods_v10	01_Programs	LogMod	INPUT	PWC\2006 PWC\2030 PWC\2030 BnM	Different sets of PWC matrices. The selection in the model is done via catalog key {Year} catalog key " Year of PWC matrices"
C:\SamGods_v10	01_Programs	LogMod	INPUT	PWC\2006\VY_F PWC\2030\VY_F PWC\2030	PWC matrices in Voyager format. Files present if PWC_Matrices has been run

The file structure

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
				BnM\VY_F	
C:\SamGods_v10	01_Programs	LogMod	LOG		Folder containing report files from Logistics Module and RCM.
C:\SamGods_v10	01_Programs	LogMod	MERGEREP		Folder containing the executable MergeRep.exe. Its purpose is to merge all ChainChoiXX.rep (XX= commodity group) in a common ChainChoi.rep
C:\SamGods_v10	01_Programs	LogMod	RCM		Folder containing the executables files: • lp_solve.exe - free software for linear programming problem • MPS.jar - executable to prepare inputs for lp_solve.exe and to extract results from output Main control file for LP: • MPS_LP.ctl • MPS_LOS.ctl And input files for functionality of locking transport chain solutions: • LockedSTDLogMod_empty.txt • LockedSTDLogMod_Sc2030.txt See user manual for further details on those two files.
C:\SamGods_v10	02_Applications				Main folder for all the applications defined in the model
C:\SamGods_v10	02_Applications	1_Editing			Folder for “Create the editable files” and “Edit the data” applications
C:\SamGods_v10	02_Applications	1_Editing	Capacity		Folder for the subgroup “Rail Capacity Checks” in the “Edit the data” application
C:\SamGods_v10	02_Applications	1_Editing	Comparison		Folder for the subgroup “Compare Nodes and Node_terminals” in the “Edit the data” application
C:\SamGods_v10	02_Applications	1_Editing	Numbering		Folder for the subgroup “Numbering System” in the “Edit the data” application
C:\SamGods_v10	02_Applications	1_Editing	Regions		Folder for the subgroup “Regions for links” in the “Edit the data” application
C:\SamGods_v10	02_Applications	2_Run			Folder for the main application “Samgods Model”
C:\SamGods_v10	02_Applications	2_Run	Ass_VY		Folder for the subgroup “Assignment” in the “Samgods

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					Model" and other subgroups for this application: • Conversion from LogMod to VY • Road Assignment • Rail Assignment • Sea Assignment; • Air Assignment
C:\SamGods_v10	02_Applications	2_Run	Calibr		Folder for the subgroup "Parameters Calculation" in the "Samgods Model" application
C:\SamGods_v10	02_Applications	2_Run	Costs_VY		Folder for the subgroup "LOS Calculation" in the "Samgods Model" and other subgroups for this application: • Data preparation ○ Tax Calculation • Road • Rail • Sea • Air • Conversion from VY-matrices to LogMod-matrices in emme format
C:\SamGods_v10	02_Applications	2_Run	Ini		Folder for the subgroup "Initialization" in the "Samgods Model" application
C:\SamGods_v10	02_Applications	2_Run	LogisticModVY		Folder for the subgroup "Logistic Module" in the "Samgods Model" and other subgroups for this application: • Prepare data • Prepare data second part • Prepare data third part • Port Area LOS input (STAN) • Run the logistic model • Save reports

C:\SamGods_v10	02_Applications	2_Run	Results		Folder for the subgroup "Results" in the "Samgods Model" and other subgroups for the application:
					• Results 1.0
					  ○ Port Areas
					  ○ Öresund Kiel and Jylland
					  ○ Reports per geographical aggregation

C:\SamGods_v10	02_Applications	3_Compare			Folder for the "Compare" application and subgroups: • LOS matrices

The file structure

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					• Assignment
C:\SamGods_v10	02_Applications	3_Compare	OD_matrices		Folder for the subgroup “OD matrices” inside “Compare” in the “Samgods Model”

C:\SamGods_v10	02_Applications	4_Handling			Folder for the main application “Handling Scenario” and subgroups:
					• Delete
					• Compact
					• Scenario export
					  ○ Model Export
					  ○ General tables
					  ○ Create the new base
					  ○ Create the new scenario specific tables
					• Scenario import

C:\SamGods_v10	02_Applications	4_Handling	Original		Folder with the empty catalog used as template during the creation of a new model
C:\SamGods_v10	02_Applications	4_Handling	Scenario_tables		Folder with scenario tables used in comparison with the new base. File produced during "Handling Scenario" application
C:\SamGods_v10	02_Applications	5_Installation			Folder for the “Installation” application
C:\SamGods_v10	02_Applications	6_PWC_matrices			Folder for the “PWC_Matrices” application
C:\SamGods_v10	02_Applications	7_Matrix_form			Folder for the “Change matrix format” application
C:\SamGods_v10	02_Applications	09_RCM1			Folder for the subgroup “Rail Capacity Management” in the “Samgods Model” and other subgroups for this application: • Warm start
C:\SamGods_v10	02_Applications	09_RCM1	Ass_VY		Folder for the subgroup “RCM Assignment” in the “Samgods Model” and other subgroups for this application: • Conversion from LogMod matrices to VY matrices • Road Assignment RCM • Rail Assignment RCM • Sea Assignment RCM • Air Assignment RCM
C:\SamGods	02_Applica	09_RCM1	Dataprep_L		Folder for the subgroup “Data

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
_v10	tions		P		Preparation LP" inside "Rail Capacity Management" in the "Samgods Model"
C:\SamGods_v10	02_Applications	09_RCM1	Final		Folder for the subgroup "Run Final Process" inside "Rail Capacity Management" in the "Samgods Model" and other subgroups for this application: • Prepare control files • Run the final process • Save Reports
C:\SamGods_v10	02_Applications	09_RCM1	LP		Folder for the subgroup "LP Loop" inside "Rail Capacity Management" in the "Samgods Model"
C:\SamGods_v10	02_Applications	09_RCM1	LP	LP0	Folder for the subgroup "LP0 step" inside "LP Loop" in the "Samgods Model" and other subgroups for this application: • Prepare CTL files - Extract procedure
C:\SamGods_v10	02_Applications	09_RCM1	LP	LP1	Folder for subgroup "LP1+ step" inside "LP Loop" in the "Samgods Model" and other subgroups for this application: • Prepare CTL files • Run BuildChain4RCM and ChainChoi4RCM • Run MPS, LP and Extract • Check convergence

C:\SamGods_v10	02_Applications	09_RCM1	LPAdjCap		Folder for subgroup "LP1 Adjust Capacities" inside "Rail Capacity Management" in the "Samgods Model" and other subgroups for this application:
					• Data Preparation
					  ○ Conversion from LogMod to VY
					  ○ Empty Assignment
					  ○ Setup Capacity Values
					• LP Adjust Capacity Loop
					  ○ LP1b step
					  ○ Prepare CTL files - Extract procedure
					  ○ Conversion from LogMod to VY
					  ○ Rail assignment
					  ○ Setup Capacity Values
					  ○ Convergence

The file structure

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					Criteria

C:\SamGods_v10	02_Applications	09_RCM1	Results		Folder for subgroup “Results RCM” in the “Samgods Model” and other subgroups for the application:
					• Report 1.0
					  ○ Port Areas
					  ○ Öresund Kiel and Jylland
					  ○ Rail Capacity
					  ○ Reports per geographical aggregation

C:\SamGods_v10	03_Gis_Data				Folder for shape files Empty and layer file (.lyr)
C:\SamGods_v10	04_Media				Folder for banners used in the interface
C:\SamGods_v10	05_Input_Data				Folder for the main geodatabase, containing all the input data (networks, tables and scenario specific tables)
C:\SamGods_v10	05_Input_Data	Calibration			Folder containing calibration files
C:\SamGods_v10	06_Reports				Folder containing Rtf files with the model operating instructions and report files (.rep) shown in data panel Scenario Reports section
C:\SamGods_v10	07_Python				Folder with Python scripts
C:\SamGods_v10	07_Python	Original			Folder with Python scripts used to install the model
C:\SamGods_v10	Scenario_Tree				Folder for results from model; it contains several subfolders specific by scenario

Table 1 – Folder structure of the model.

Input and output file reference

File summary

The file list is organised according to location. There is a main folder for the input data and an output folder for each scenario.

Under the folder C:\SamGods_v10\05_Input_data\Input_Data.mdb there is the main geodatabase with all the input data. The input data could be classified in the following classes:

1. General tables to define lookup values
2. Tables that represent the base values used in the model
3. Tables that represent the values that differ from the base values and are related to a specific scenario. These are denoted scenario specific tables
4. Tables to manage database information

The following table lists the input data and specifies what the class for each table is:

Input name	Format	Class
A_CountyName	Table	1
A_defaul_frequencies	Table	
A_F2FTypes	Table	
A_LinkCategories	Table	
A_Mode_Type	Table	
A_NodeClass	Table	
A_Par_Class	Table	
A_Port_areas	Table	
A_Transfer_Type	Table	
A_Vessel_Type	Table	
A_Zoning_system	Table	
BuildChain_CONSOL	Table	
BuildChain_MODES	Table	
ChainChoice_MODES	Table	
Direct_Access	Table	
Extract_parameters	Table	
Inzone_Distance	Table	
Macros_EMME	Table	
Modes	Table	
Other_statistics	Table	
Port_statistics	Table	
SamGods_zones	Stand-alone Feature class polygons	
V101_SpeedFlowCurves	Table	
V102_SpeedFlowCurves	Table	

Input name	Format	Class
Base2006	Network	2
Frequency_Data_Base2006	Network	
Cargo_Base2006	Table	
Node_terminals_Base2006	Table	
Nodes_Base2006	Table	
Nodes_Commodities_Base2006	Table	
PropLink_Base2006	Table	
Rail_Capacity_Base2006	Table	
Tax_Category_Base2006	Table	
Tax_Country_Base2006	Table	
Tax_Link_Base2006	Table	
Toll_Link_Base2006	Table	
Vehicles_exception_Base2006	Table	
Vehicles_parameters_Base2006	Table	
General_Base2006	Table	3
LogMod_Base2006	Table	
Sc_Base2006_Cargo	Table	
Sc_Base2006_Frequency_Data_Link	Table	
Sc_Base2006_Frequency_Data_Node	Table	
Sc_Base2006_Link	Table	
Sc_Base2006_Node	Table	
Sc_Base2006_Node_Terminals	Table	
Sc_Base2006_Nodes	Table	
Sc_Base2006_Nodes_commodities	Table	
Sc_Base2006_PropLink	Table	
Sc_Base2006_Rail_Capacity	Table	
Sc_Base2006_Tax_Category	Table	
Sc_Base2006_Tax_Country	Table	
Sc_Base2006_Tax_Link	Table	
Sc_Base2006_Toll_Link	Table	
Sc_Base2006_Vehicles_exc	Table	
Sc_Base2006_Vehicles_parameters	Table	
History	Table	4
Model_description	Table	

Table 2 – Input data in geodatabase.

Under the folder C:\SamGods_v10\05_Input_data\Calibration three files:

- PortAreasParameters_22092014_L1_Base2006R21_RUN20.txt - this file is a result of the calibration procedure carried out for port areas.
- Parameters_Kielcalibration.dbf and Parameters_portcalibration.dbf - in the calibration process the parameters for a new loop are estimated in a calculation that requires some control parameters. These tables hold the parameters used.

Description of the calibration process is provided in Chapter 5.8, and description of the file structure in Chapter 2.3.

Under each scenario folder, for instance SamGods\Scenario_tree\Base2006, will be present:

- one or more LogMod_Y folders, where Y refers to the cycling process for calibration. For a standard user this will always be LogMod_1. This folder contains all the standard outputs from the Logistics and Rail Capacity Management modules. A description of the file system and structure is provided in chapter 0
- the following output files (the ones actually produced depends on user choices). In this example the scenario is Base2006. A different scenario will have the same output files with the scenario name included in the file name, for instance Output0_Base2006.mdb will be Output0_Base2006R62.mdb if the scenario is Base2006R62. In the next table and the rest of the document we will refer to {SCENARIO_SHORTNAME} to highlight this concept
- Note that in the table below X represents the commodity number

File name	Output name	Format
Output0_{SCENARIO_SHORTNAME}.mdb	Model_description	Table
	Outputs	Table
	Node_labels	Table
	ProprLink_{SCENARIO_SHORTNAME}	Table
	Load_net_Road_0	Network
	Load_net_Rail_0	Network
	Load_net_Sea_0	Network
	Load_net_Air_0	Network
	Loaded_Net_0	Network
	Loaded_Bid_0	Network
	CHAIN_OD_COV_{SCENARIO_SHORTNAME}_0	Table
	COM_L_D_{SCENARIO_SHORTNAME}_0	Table
	VHCL_OD_COV_{SCENARIO_SHORTNAME}_0	Table
	COMPARE_LOAD0_*{Scenario_short_name}*_{Scenario_for_comparios)	Network
	Report_1_{SCENARIO_SHORTNAME}_0	Table
	Report_3_TonKM_perMode_0	Table
	Report_4_{SCENARIO_SHORTNAME}_0	Table
	Report_5_{SCENARIO_SHORTNAME}_0	Table
	Report_6_{SCENARIO_SHORTNAME}_0	Table
	Report_7_{SCENARIO_SHORTNAME}_0	Table
	Report_8_{SCENARIO_SHORTNAME}_0	Table
	Report_9_{SCENARIO_SHORTNAME}_0	Table
	Report_10_{SCENARIO_SHORTNAME}_0	Table
	Report_11_{SCENARIO_SHORTNAME}_0	Table
	Report_12_{SCENARIO_SHORTNAME}_0	Table
	Report_13_Portarea_0_STD	Table
	Report_14_Oresund_Kiel_0_STD	Table
	Report_16_{SCENARIO_SHORTNAME}_0	Table
	Report_17_{SCENARIO_SHORTNAME}_0	Table
	Report_18_{SCENARIO_SHORTNAME}_0	Table
	Load_net_Road_RCM	Network
	Load_net_Rail_RCM	Network
	Load_net_Sea_RCM	Network
	Load_net_Air_RCM	Network
	Loaded_Net_RCM	Network
	Rail_BiDir	Network
	Compare_Bid	Network
	Loaded_Bid_RCM	Network
	CHAIN_OD_COV_{SCENARIO_SHORTNAME}_RCM	Table
	COM_L_D_{SCENARIO_SHORTNAME}_RCM	Table

File name	Output name	Format
	VHCL_OD_COV_{SCENARIO_SHORTNAME}_RCM	Table
	COMPARE_LOADRCM_*{Scenario_short_name}*_{Scenario_for_comparios)	Network
	Report_1_{SCENARIO_SHORTNAME}_RCM	Table
	Report_3_TonKM_perModeRCM	Table
	Report_4_{SCENARIO_SHORTNAME}_RCM	Table
	Report_5_{SCENARIO_SHORTNAME}_RCM	Table
	Report_6_{SCENARIO_SHORTNAME}_RCM	Table
	Report_7_{SCENARIO_SHORTNAME}_RCM	Table
	Report_8_{SCENARIO_SHORTNAME}_RCM	Table
	Report_9_{SCENARIO_SHORTNAME}_RCM	Table
	Report_10_{SCENARIO_SHORTNAME}_RCM	Table
	Report_11_{SCENARIO_SHORTNAME}_RCM	Table
	Report_12_{SCENARIO_SHORTNAME}_RCM	Table
	Report_13_Portarea_RCM	Table
	Report_14_Oresund_Kiel_RCM	Table
	Report_15_RailLinksRMC	Table
	Report_16_{SCENARIO_SHORTNAME}_RCM	Table
	Report_17_{SCENARIO_SHORTNAME}_RCM	Table
	Report_18_{SCENARIO_SHORTNAME}_RCM	Table
Output0_{SCENARIO_SHORTNAME}.vpr		Visual project file (GRAPHICS)
COST_ROAD_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
COST_RAIL_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
COST_SEA_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
COST_AIR_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
ROAD_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
ROAD_TON0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_TON0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_TON0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_TON0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
ROAD_EMP0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_EMP0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_EMP0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_EMP0_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
ROAD_VHCLFLOW_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_VHCLFLOW_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_VHCLFLOW_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_VHCLFLOW_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix

Input and output file reference

File name	Output name	Format
ROAD_TON_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_TON_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_TON_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_TON_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
ROAD_EMP_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_EMP_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_EMP_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_EMP_FIN_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
Scenario_Data.mdb		Database
Scenario_Data.vpr		Visual project file (GRAPHICS)
Input_data.mxd		Visual project map file (GIS)
Input_data.mxr		File index
CalParameter_Loop_1.txt		Text File
CalParameter_Loop_2.txt		Text File
CalParameterNextLoop.txt		Text File
FINAL_NETWORK_B06.NET		Binary Network
KielNextValue.txt		Text File
Log_File.log		Text File
logfile_scenario.dat		Dat file
ParKiel_BS06.txt		Text File
REPORT_ED_BS06.txt		Text File
REPORT_SG_BS06.txt		Text File
Differences_BS06.csv		CSV File
PortArea_report_BS06_Loop1.csv		CSV File
PortArea_report_RCM_BS06_Loop1.csv		CSV File
EMME_NET_{SCENARIO_SHORTNAME}.211		Emme file
V101_102_LINK_SPEEDS.DAT		DAT File
Logfile_scenario.dat		DAT File
RENUM_NODES.DBF		Dbase File
Rail_capacities_STDBS06.DBF		Dbase File
General_Table_{SCENARIO_SHORTNAME}.DBF		Dbase File
Capacity_table_BS06.DBF		Dbase File
ChainChoi0_{SCENARIO_SHORTNAME}.rep		Report File
ChainChoi0_{SCENARIO_SHORTNAME}FIN.rep		Report File
OutputX_{SCENARIO_SHORTNAME}.mdb	Model_description	Table
	Outputs	Table
	Node_labels	Table
	Load_net_Road_X	Network
	Load_net_Rail_X	Network

File name	Output name	Format
	Load_net_Sea_X	Network
	Load_net_Air_X	Network
	Loaded_Net_X	Network
	Loaded_Bid_X	Network
	CHAIN_OD_COV_{SCENARIO_SHORTNAME}_X	Table
	COM_L_D_{SCENARIO_SHORTNAME}_X	Table
	VHCL_OD_COV_{SCENARIO_SHORTNAME}_X	Table
	COMPARE_LOADX_*{Scenario_short_name}*_{Scenario_for_comparios)	Network
	Report_1_{SCENARIO_SHORTNAME}_X	Table
	Report_3_TonKM_perMode_X	Table
	Report_4_{SCENARIO_SHORTNAME}_X	Table
	Report_5_{SCENARIO_SHORTNAME}_X	Table
	Report_6_{SCENARIO_SHORTNAME}_X	Table
	Report_7_{SCENARIO_SHORTNAME}_X	Table
	Report_8_{SCENARIO_SHORTNAME}_X	Table
	Report_9_{SCENARIO_SHORTNAME}_X	Table
	Report_10_{SCENARIO_SHORTNAME}_X	Table
	Report_11_{SCENARIO_SHORTNAME}_X	Table
	Report_12_{SCENARIO_SHORTNAME}_X	Table
	Report_13_Portarea_X_STD	Table
	Report_14_Oresund_Kiel_X_STD	Table
	Report_16_{SCENARIO_SHORTNAME}_X	Table
	Report_17_{SCENARIO_SHORTNAME}_X	Table
	Report_18_{SCENARIO_SHORTNAME}_X	Table
ROAD_VHCLFLOWX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_VHCLFLOWX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_VHCLFLOWX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_VHCLFLOWX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
ROAD_TONX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_TONX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_TONX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_TONX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
ROAD_EMPX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_EMPX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
SEA_EMPX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
AIR_EMPX_{SCENARIO_SHORTNAME}.MAT		Voyager Matrix
RAIL_COST_DISTX_{SCENARIO_SHORTNAME}.CSV		DBF file
RAIL_VHCLFLOW_FLOWX_{SCENARIO_SHORTNAME}.CSV		CSV file

Table 3 – List of outputs.

Input data

Lookup tables

2.1.1.1. A_CountyName

05_Input_data\Input_data.mdb\A_CountyName

Table Format:

Field	Description	Data Format	Example
COUNTY	Numerical identifier	Double	1
NAME	County name	Text	Stockholm

Table 4 – Format of "A_CountyName" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Used in reporting phase to associate the county description to reports.

Editing Options: No need to be updated . (only a new definition of counties classification will require changes on this)

Used by: "Reports per geographical aggregation" subgroup under "Samgods Model\Results\Results 1.0" and under "Samgods Model\Results RCM\Reports 1.0"

2.1.1.2. A_default_frequencies

05_Input_data\Input_data.mdb\A_defaul_frequencies

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_REL	Counter	Integer	1
TER_TYPE_O	Terminal code for origin	Integer	0
TER_TYPE_D	Terminal code for destination	Integer	0
SYSTEM	Frequency System	Double	0
LORRY	Frequency Lorry	Double	84
WAGONLOAD	Frequency WaggonLoad	Double	0
RAILFERRY	Frequency RailFerry	Double	0
ROADFERRY	Frequency RoadFerry	Double	0
CON_VESSEL	Frequency ContainerVessel	Double	0
OTH_VESSEL	Frequency Other Vessels	Double	0.5
RORO_VESS	Frequency RororVessel	Double	0.2
COMBI	Frequency Combi	Double	0
AIR	Frequency Air	Double	5

Table 5 – Format of "A_default_frequencies" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Define the default frequencies for the frequency matrices based on the terminal type (road, rail, sea, ferry or air)

Editing Options: Only needs to be updated in case of changes of the default frequencies

Used by: "Frequency matrices" subgroup under "Edit the data\Create the tables specific of scenario" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.3. A_F2FTypes

05_Input_data\Input_data.mdb\A_F2FTypes

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_F2F	Code	Integer	0
DESCRIPTIO	Description of the firm to firm relation	Text	singular or transit

Table 6 – Format of "A_F2FTypes" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Define the type of relations between firm to firm in the PWC matrices

Editing Options: No need to be updated

Used by: "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.4. A_LinkCategories

05_Input_data\Input_data.mdb\A_LinkCategories

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
CATEGORY	Category code	Double	11
MODE_T	Description of the mode	Text	Road
MODE	Code for allowed modes	Integer	1
ID_COUNTRY	Country code	Double	1
GENERAL_MO	Description of the general mode	Text	Road
DESCRIPTIO	Type of infrastructure	Text	Oresund Bridge
AREA	Description of geographical covered area	Text	Balstic Sea

Table 7 – Format of "A_LinkCategories" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: List of categories defined in the network and the associated country code. The main use is to derive the ID_COUNTRY code for each link in the network, e.g. geographical classification. Second use is to derive statistics at different levels in the reporting stage, such as vehiclekms and tonnekms per E10 roads (CATEGORY=11)

Editing Options: Add a new line for each new category defined in the network

Used by: "Editable map data" and "Editable tables" subgroups under "Create the editable files", "Emme Edits" subgroup under "Edit the data\Create the tables specific of scenario", "Tax calculation" subgroup under "Samgods Model\LOS calculation\Data Preparation" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.5. A_Mode_Type

05_Input_data\Input_data.mdb\A_Mode_Type

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	5
MODE	Code for allowed modes	Double	4
DESCRIPTIO	Description of mode	Text	Ferry
GEN_MODE	General mode code	Integer	3
DESCR_GENMODE	Description of general mode	Text	Sea

Table 8 – Format of "A_Mode_Type" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: List of modes defined in the network and related MODE_L, MODE_N attributes in node and link tables.

Editing Options: No need to be updated . (only the introduction of new general will require changes on this. This will impact all the model structure and require several amendments)

Used by: "Editable map data" subgroup under "Create the editable data", "General tables" subgroup under "Handling scenario\Scenario Export" and graphical editor during the editing of the network. Is one of the codes mandatory to edit properly the network (other two are the country code and the region code)

2.1.1.6. A_NodeClass

05_Input_data\Input_data.mdb\A_NodeClass

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	10

Field	Description	Data Format	Example
ID	Counter	Integer	10
DESCRIPTIO	Description of node class	Text	Road - regional system
RULE	Rule to define node number	Text	From 20000 to 259999
MIN_V	Minimum value for node number	Double	20000
MAX_V	Maximum value for node number	Double	260000
MODE	Code for allowed modes	Integer	1
DES_MODE	Description of mode	Text	Road – Domestic

Table 9 – Format of "A_NodeClass" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: List of node classes defined in the network. The initial network used has been developed in EMME with a list of rules for numbering. Those rules are described in this table. To derive new EMME numbers when the network is modified in Cube GIS window, those rules are applied.

Editing Options: No need to be updated . (only the introduction of new general will require changes on this. This will impact all the model structure and require several amendments)

Used by: "Numbering System Voyager" subgroup under "Edit the data\Create the tables specific of scenario", "Tax calculation" subgroup under "Samgods Model\LOS calculation\Data Preparation" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.7. A_Par_Class

05_Input_data\Input_data.mdb\A_Par_Class

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
CLASS	ID of previous table	Integer	12
MIN_R	Minimum value for zones in CLASS	Integer	1
MAX_R	Maximum value for zones in CLASS	Integer	399

Table 10 – Format of "A_Par_Class" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: The hierarchic rule in EMME numbering has in the first three digits the region code. In this table are represented the minimum and maximum country code associated to the class of nodes present in A_NodeClass

Editing Options: No need to be updated . (only the introduction of new numberig system will require changes on this. This will impact all the model structure and require several amendments)

Used by: Not used in the system. Illustration purposes

2.1.1.8. A_Port_areas

05_Input_data\Input_data.mdb\A_Port_areas

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	9
PortNodes_PortAreaNbr	Port area number	Double	8
Description	Description of port area	Text	Visby (Gotland)

Table 11 – Format of "A_Port_areas" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: List of defined domestic port areas. The calibration of the model carried out, requires as input a table with scaling factors by port area and STAN group respectively. Each domestic port belongs to a specific port area. This table defines them

Editing Options: No need to be updated. (only a new definition of port area classification will require changes on this)

Used by: "Port Areas" subgroup under "Samgods Model\Results\Results 1.0" and under "Samgods Model\Results RCM\Reports 1.0" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.9. A_Transfer_Type

05_Input_data\Input_data.mdb\A_Transfer_type

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
TRANSF_TYP	Transfer type code	Integer	1
DESCRIPTIO	Description of transfer type	Text	TransferRoadRoad

Table 12 – Format of "A_Transfer_Type" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: List of defined transfer types used in Nodes_Commodities_Base2006 table

Editing Options: No need to be updated. (only a new definition of transfer type will require changes on this)

Used by: Editable map data" subgroup under "Create the editable files" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.10. A_Vessel_type

05_Input_data\Input_data.mdb\A_Vessel_type

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
VES_TYPE	Vehicle Code	Integer	1
DESCRIPTIO	Description of vehicle	Text	containers

Table 13 – Format of "A_Vessel_type" table.

Software required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: List of vehicle types defined in the logistic chains (container and non container)

Editing Options: No need to be updated. (only a new definition of special vehicle type will require changes on this)

Used by: "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.11. A_Zoning_System

05_Input_data\Input_data.mdb\A_Zoning_System

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	2
COUNTRY_CO	Country code	Double	1
REGION_COD	Region code	Integer	115
COUNTRY	Country description	Text	Sweden
REGION	Region description	Text	Vallentuna
CENTROID_C	Centroid code	Integer	711500
SCBSTANN	Scbstann code	Integer	115
SWEDEN	Boolean variable indicating if the zone is in Sweden	Integer	1

Table 14 – Format of "A_Zoning_System" table.

Software required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: List of zones and related country, region and scbstann codes

Editing Options: No need to be updated (a revision of zoning system would impact all the model, therefore for normal usage of the model is not recommended to change this)

Used by: “Editable tables” subgroup under “Create the editable files”, “Numbering System Voyager” subgroup under “Edit the data\Create the tables specific of scenario”, “Tax calculation” subgroup under “Samgods Model\LOS calculation\Data Preparation”, “Reports per geographical aggregation” subgroup under “Samgods Model\Results\Results 1.0” and under “Samgods Model\Results RCM\Reports 1.0” and “General tables” subgroup under “Handling scenario\Scenario Export”

2.1.1.12. BuildChain_CONSOL

05_Input_data\Input_data.mdb\BuildChain_CONSOL

Table Format:

Field	Description	Data Format	Example
ID	Counter	Integer	1
MODE_C	<mode>	Text	A
CONSOL_L	CONSOL<mode> lower bound	Double	0.1
CONSOL_U	CONSOL<mode> upper bound	Double	0.3
Mode	Description	Text	Road
CONSOL_L2	CONSOL<mode> for STAN2 lower bound	Double	0.1
CONSOL_U2	CONSOL<mode> for STAN2 upper bound	Double	0.3
CONSOL_L8	CONSOL<mode> for STAN2 lower bound	Double	0.1
CONSOL_U8	CONSOL<mode> for STAN2 upper bound	Double	0.3
CONSOL_L9	CONSOL<mode> for STAN9 lower bound	Double	0.1
CONSOL_U9	CONSOL<mode> for STAN9 upper bound	Double	0.3

Table 15 – Format of “BuildChain_CONSOL” table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: The consolidation factors, specified as two values representing an allowed range, in this table are specified for all the commodities (CONSOL_L and CONSOL_U) and for specific STAN groups. Four general control files for BuildChain will be produced:

- BuildChain_Common.ctl (with CONSOL_L, CONSOL_U values)
- BuildChain_STAN2.ctl (with CONSOL_L2, CONSOL_U2 values)
- BuildChain_STAN8.ctl (with CONSOL_L8, CONSOL_U8 values)
- BuildChain_STAN9.ctl (with CONSOL_L9, CONSOL_U9 values)

In each BuildChainXX.CTL (control file specific for each XX commodity) the control parameter INCL=xxx will refer to the proper general BuildChain Control file.

For instance:

Commodity	INCL control parameter
1	INCL=BuildChain_common.ctl
33	INCL=BuildChain_STAN9.ctl

Editing Options: Instructions on setting up different values for different scenarios has been explained in User Manual 6.1. The table could be copied via Data Manager and opened in dbf window in GUI interface. Its reference is controlled by catalog key " Consolidation factors table"

Used by: "Prepare Data third part" subgroup under "Samgods Model\Logistics Module" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.13. BuildChain_MODES

05_Input_data\Input_data.mdb\BuildChain_MODES

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_COMM	Commodity class	Integer	1
A	Main vehicle type for chain type A	Integer	104
B	Main vehicle type for chain type B	Integer	102
C	Main vehicle type for chain type C	Integer	104
D	Main vehicle type for chain type D	Integer	201
E	Main vehicle type for chain type E	Integer	202
F	Main vehicle type for chain type F	Integer	208
G	Main vehicle type for chain type G	Integer	202
H	Main vehicle type for chain type H	Integer	208
I	Main vehicle type for chain type I	Integer	204
J	Main vehicle type for chain type J	Integer	0
K	Main vehicle type for chain type K	Integer	0
L	Main vehicle type for chain type L	Integer	0
M	Main vehicle type for chain type M	Integer	310
N	Main vehicle type for chain type N	Integer	315
O	Main vehicle type for chain type O	Integer	317
P	Main vehicle type for chain type P	Integer	319
Q	Main vehicle type for chain type Q	Integer	321
R	Main vehicle type for chain type R	Integer	401
T	Main vehicle type for chain type T	Integer	205
U	Main vehicle type for chain type U	Integer	206

Table 16 – Format of "BuildChain_MODES" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Main vehicle type used in the BuilChain process for each logistic chain and commodity class. It is the translation in access table of information present in document listed under Section 0.3 Table 5. Note that the values differ for Vehicle Type 40, since new road chains were added (104 and 105 for C and 318, 319, 320 for P) The values in the model are the most up-to-date

Editing Options: No need to be updated.

Used by: “Prepare Data third part” subgroup under “Samgods Model\Logistics Module” and “General tables” subgroup under “Handling scenario\Scenario Export”

2.1.1.14. Chain_List

05_Input_data\Input_data.mdb\Chain_List

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	2
ID_C	Logistic chain code	Integer	2
CHAIN_TYPES	Description of logistic chain	Text	ADA

Table 17 – Format of “Chain_List” table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Specifies the possible chains in the model. Each letter represents a specific submode and vehicle type. It is the translation in the access table with information present in document listed under Section 0.3 Table 5. Note that the values are different from those in the report. The values in the model are the most up-to-date.

Editing Options: Needs to be updated when the chain types are revised.

Used by: “Prepare Data third part” and “Save Reports” subgroups under “Samgods Model\Logistics Module” and “General tables” subgroup under “Handling scenario\Scenario Export”

2.1.1.15. ChainChoice_MODES

05_Input_data\Input_data.mdb\ChainChoice_MODES

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
MODE_C	Chain type	Text	A
VHCL_NR	Vehicle type	Integer	104
VES_TYPE	Container/non container	Double	1

Table 18 – Format of “ChainChoice_MODES” table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of use: Specifies the vehicle type and vessel type (specified in A_Vessel_type) associated to each mode present in Chain_List. It is the translation in access table of information present in document listed under Section 0.3 Table 3

This table is used in building the ChainChoixx.ctl files for parameters VHCLA..VHCLR. See document listed under Section 0.3 pag. 32.An example is as follow (ChainChoi01.ctl):

```
VHCLA=104,105

VHCLD=201
```

Those values correspond to:

OBJECTID	MODE_C	VHCL_NR	VES_TYPE
1	A	104	1
2	A	105	1
3	D	201	1

Editing Options: Needs to be updated when chain types and associated vehicle and vessel types are revised

Used by: "Prepare Data third part" and "Save Reports" subgroups under "Samgods Model\Logistics Module" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.16. Direct_Access

05_Input_data\Input_data.mdb\Direct_Access

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_COM	Code of commodity class	Integer	1
F2F_0	Flag to direct access for F2F type 0	Integer	1
F2F_1	Flag to direct access for F2F type 1	Integer	0
F2F_2	Flag to direct access for F2F type 2	Integer	0
F2F_3	Flag to direct access for F2F type 3	Integer	1
F2F_4	Flag to direct access for F2F type 4	Integer	0
F2F_5	Flag to direct access for F2F type 5	Integer	0
F2F_6	Flag to direct access for F2F type 6	Integer	1
F2F_7	Flag to direct access for F2F type 7	Integer	1
F2F_8	Flag to direct access for F2F type 8	Integer	1
F2F_9	Flag to direct access for F2F type 9	Integer	1

Table 19 – Format of "Direct_Access" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Define per each type of f2f flow (the types are described in table A_F2FTypes) and commodity group if direct access is allowed.

This table is used when building the ChainChoixx.ctl files for the parameter DIRACC. See See document listed under Section 0.3 pg.

Each row has been translated from an access table:

Input and output file reference

OBJECTID	ID_COM	F2F_0	F2F_1	F2F_2	F2F_3	F2F_4	F2F_5	F2F_6	F2F_7	F2F_8	F2F_9
1	1	1	0	0	1	0	0	1	1	1	1

Inside the control file it looks like:

DIRACC=1,0,0,1,0,0,1,1,1,1

Editing Options: Needs to be updated when direct access for a specific f2f category and product group is revised

Used by: "Prepare Data third part" subgroup under "Samgods Model\Logistics Module" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.17. Extract_parameters

05_Input_data\Input_data.mdb\Extract_parameters

Table Format:

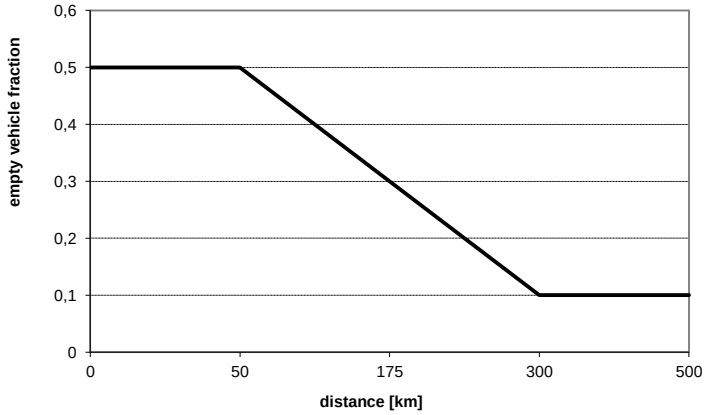
Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID	Counter	Integer	1
VEH_NR	Vehicle class	Double	101
DIST1	First threshold distance	Double	50
EMPFAC1	Add-on factor applied up to DIST1 determining the proportion of empty vehicle flows to loaded vehicle flows. Between DIST1 and DIST2 a linear interpolation is carried out between EMPFAC1 and EMPFAC2	Double	0.5
DIST2	Second threshold distance	Double	300
EMPFAC2	Add-on factor applied above DIST2 determining the proportion of empty vehicle flows to loaded vehicle flows	Double	0.1
ASYM	Threshold distance above which asymmetric loaded vehicle flows will generate the same empty vehicle flow	Double	50

Table 20 – Format of "Extract_parameters" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: The table provides, for each vehicle type, the function that will be applied in the extract procedure. The extract procedure reads the emptyfrac.dat placed under EXTRACT folder and applies the values specified in this table (DIST1, EMPFAC1, DIST2,EMPFAC2 etc for specification of a piecewise linear function). The ASYM control parameter present in each extractXXX.ctl file is filed out with values from this table

Editing Options: Needs to be updated when specifying different curves



An example to understand the parameters in this table, using the curve above, is:

- DIST1=50 km
- EMPFAC1=0.5
- DIST2=300 km
- and EMPFAC2=0.1

Used by: “prepare Data second part” and “Prepare Data third part” subgroups under “Samgods Model\Logistics Module” and “Prepare CTL Files” subgroup under “Samgods Model\Rail Capacity Management\LP Loop\LP0 step”, “Samgods Model\Rail Capacity Management\LP Loop\LP1+ step”, “Samgods Model\Rail Capacity Management\LP1 Adjust Capacities\LP Adjust Capacity Loop\LP1b step” and “Samgods Model\Rail Capacity Management\Run Final Process”

2.1.1.18. Inzone_Distance

05_Input_data\Input_data.mdb\Inzone_Distance

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	2
ZONEID	Zone code (NORIG value)	Integer	711500
DISTANCE	Intrazonal distance [km]	Double	5.53

Table 21 – Format of “Inzone_Distance” table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Inzone_Distance represents the default values for the intrazonal distances on domestic zones. This value will be added both for the distance, domestic distance and time skims during the cost calculation phases

Editing Options: No edits to be updated. (the revision of this is connected to the revision of zoning system. therefore in the normal usage of the model is not recomended to change this)

Used by: “Data Preparation” subgroup under “Samgods Model\LOS calculation” and “General tables” subgroup under “Handling scenario\Scenario Export”

2.1.1.19. Macros_EMME

05_Input_data\Input_data.mdb\Macros_EMME

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID	Counter	Double	1
VEH_NR	Vehicle class	Double	101
DESCRIPTIO	Vehicle description	Text	Lorry light LGV.< 3.5 ton
EMME2_C	Name of the macro in EMME for cost calculation	Text	~<v100.mac
MATIN	Name of the macro in EMME for load the OD matrix	Text	~<veh_matin_100.mac
EMME2_A	Name of the macro in EMME for assignment	Text	~<asg100.mac
ODMATRIX	Name of OD matrix in EMME format	Text	od_vhcl101.314
ODEMME	Matrix number in emme2bank	Text	mf11

Table 22 – Format of “Macros_EMME” table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: The table is used to specify the matrix label printed out in OD matrices when extracted from the transport chain solutions. Values in ODEMME column will specified the value for control parameter ID in the extract control file. For instance for Vehicle type 101 in control file we have:

```
ID=mf11
```

And this value comes from ODEMME attribute in this table:

OBJECTID	ID	VEH_NR	DESCRIPTIO	EMME2_C	MATIN	EMME2_A	ODMATRIX	ODEMME
1	1	101	Lorry light LGV.	~<v100.mac	~<veh_matin_1	~<asg100.mac	od_vhcl101.314	mf11

Editing Options: If user wishes to change the matrix label, the values could be updated.

Used by: “Prepare Data third part” subgroup under “Samgods Model\Logistics Module”, “Prepare CTL Files” subgroup under “Samgods Model\Rail Capacity Management\LP Loop\Lp0 step”, “Samgods Model\Rail Capacity Management\LP Loop\LP1+ step”, “Samgods Model\Rail Capacity Management\LP1 Adjust Capacity Loop\LP1b step” and “Samgods Model\Rail Capacity Management\Run Final Process”, “General tables” subgroup under “Handling scenario\Scenario Export”

2.1.1.20. Modes

05_Input_data\Input_data.mdb\Modes

Table Format:

Field	Description	Data Format	Example
OBJECTID	Count	Integer	1
CODE	Code for each mode	Text	x
DECR_MODE	Description of the vehicle class associated to mode	Text	All_m
VAL1	Code 1: code used in Voyager process to sort properly some tables during the run	Double	1
VAL2	Code 2: code used in Voyager process to sort properly some tables during the run	Double	1

Table 23 – Format of "Modes" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Define the modes in the network and some codes used in the Voyager application to properly sort records in intermediate files

Editing Options: Update with new codes if need to add a different vehicle type

Used by: "Editable map data" subgroup under "Create the editable files", "Numbering System Voyager" subgroup under "Edit the data\Create the tables specific of scenario", and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.1.21. Other_statistics

05_Input_data\Input_data.mdb\Other_statistics

Table Format:

Field	Description	Data Format	Example
ID	Counter	Integer	1
A	Start node of the link	Double	24604
B	End node of the link	Double	2566
DES	Description of statistics	Text	OresundRoad
Statistics	Unit used for statistics (number of vehicles or tons per year)	Text	Vehicles
ID_S	Identification number for unit in statistics (1 - number of vehicles, 2 - tons)	Double	1
Value	Tons or vehicles per year	Double	251589

Table 24 – Format of "Other_statistics" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: This table contains statistics on Oresund Bridge (both for rail and road components) and Kiel Canal and Jylland region. The values refer to base year 2006. The table also contains the links (A and B nodes) that represent those infrastructures or services. This table is used in setting up the comparisons between Base2006 scenario and statistics reported in Report #3.

Editing Options: Since these are the reference numbers for comparisons, no edits are required

Used by: "Oresund Kiel and Jylland" subgroup under "Samgods Model\Results\Results 1.0" and "Oresund Kiel and Jylland" under "Samgods Model\Results RCM\Reports 1.0"

2.1.1.22. Port_statistics

05_Input_data\Input_data.mdb\Port_statistics

Table Format:

Field	Description	Data Format	Example
PortAreaNbr	Port area code	Double	1
STAN1	Tonnes/1000 through port area of STAN group 1	Double	0.74
STAN2	Tons/1000 on port area and STAN group 2	Double	889.46
STAN3	Tons/1000 on port area and STAN group 3	Double	879.24
STAN4	Tons/1000 on port area and STAN group 4	Double	2.33
STAN5	Tons/1000 on port area and STAN group 5	Double	1722.14
STAN6	Tons/1000 on port area and STAN group 6	Double	594.2
STAN7	Tons/1000 on port area and STAN group 7	Double	5445.77
STAN8	Tons/1000 on port area and STAN group 8	Double	123.8
STAN9	Tons/1000 on port area and STAN group 9	Double	571.43
STAN10	Tons/1000 on port area and STAN group 10	Double	730.06
STAN11	Tons/1000 on port area and STAN group 11	Double	613.97
STAN12	Tons/1000 on port area and STAN group 12	Double	12.74

Table 25 – Format of "Port_statistics" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: This table contains statistics on port areas divided per STAN group. The values refer to base year 2006. This table is used in setting up the comparisons between Base2006 scenario and statistics reported in Report #13.

Editing Options: Since these are the reference numbers for comparisons, no edits are required

Used by: “Port Areas” subgroup under “Samgods Model\Results\Results 1.0”, “Port Areas” under “Samgods Model\Results RCM\Reports 1.0” and “General tables” subgroup under “Handling scenario\Scenario Export”

2.1.1.23. SamGods_zones

05_Input_data\Input_data.mdb\SamGods_zones

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	33
Shape	Feature class type	OLE Object	Long Binary Data
COUNTRY_CO	Country code	Integer	518
REGION_COD	Region code	Integer	201
COUNTRY	Description of country	Text	Norway
REGION	Description of region	Text	Akershus
CENTROID_C	Centroid code	Integer	960100
SCBSTANN	SCBSTANN code	Integer	2601
Shape_Length	Length of perimeter in meters	Double	393316.015
Shape_Area	Area of shape in square meters	Double	5030650696.500

Table 26 – Format of “SamGods_zones” table.

Software Required: Cube GIS Window (see Help>Cube Base>GIS window) Or ArcMap

Description of Use: Defines the zoning system in the model. It is used during the editing to assist the user in the definition of COUNTRY_CO and REGION_CO. It is also used to classify links on boundaries per SCBSTANN

Editing Options: No need to be updated (a revision of zoning system would impact all the model, therefore for normal usage of the model is not recomended to change this).

Used by: “Prepare temporary data” subgroup under “Create the editable files”, “Regions for links” subgroup under “Edit the data\Save changes in the main gdb” and GIS window to select the country and region codes during the edit of the network

2.1.1.24. V101_SpeedFlowCurves

05_Input_data\Input_data.mdb\V101_SpeedFlowCurves

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
NUM_CURVE	Function number	Integer	1
SPEEDBASE1	Base speed for first part of equation	Double	111
A1	A parameter for first part of equation	Double	4.211003
B1	B parameter for first part of equation	Double	107.2412
C1	C parameter for first part of equation	Double	0.009474
D1	D parameter for first part of equation	Double	0.00007
G1	G parameter for first part of equation	Double	0.001405
SPEEDBASE2	Base speed for second part of equation	Double	0
A2	A parameter for second part of equation	Double	0
B2	B parameter for second part of equation	Double	0
C2	C parameter for second part of equation	Double	0
D2	D parameter for second part of equation	Double	0
G2	G parameter for second part of equation	Double	0

Table 27 – Format of "V101_SpeedFlowCurves" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Define the delay functions based on the category of the link for vehicle class 101

Editing Options : No need to be updated

Used by: "Road" subgroup under "Samgods Model\LOS Calculation", "Road Assignment" subgroup under "Samgods Model\Assignment", "Road Assignment" subgroup under "Samgods Model\RCM Assignment" and "General tables" subgroup under "Handling scenario\Scenario Export"

NOTE: This table represents the coefficients used in delay curves defined for each category. This table will be used if there will be no speed values for the road mode in Sweden. Their formats in the programs are discussed below. There are two different structures for the delay functions, depending on the number of lanes on a link:

- First structure:

  TC[NUM_CURVE]=LI.UL2 / MIN (SPEEDBASE1,
  *(A1+(B1+(C1*exp(D1*(v/0.88)))/(1+G1*exp(D1*(v/0.88)))*

- Second structure:

  *TC[40]=CmpNumRetNum(LI.NLANES,'>',1.5,(LI.UL2/ (MIN(SPEEDBASE1,A1 + (B1 + C1 * exp(D1 *(V/0.12))) /(1 + G1 * exp(D1 *(V/0.12)))))), (LI.UL2/ (MIN(SPEEDBASE2,A2 + (B2 + C2 * exp(D2 *(V/0.12))) / (1 + G2 * exp(D2 *(V/0.12)))))))*

where:

- *LI.UL2* – Distance in km
- *V* – Volume of vehicle flow (per hour or units per day);
- *LI. NLANES* – Number of lanes.

2.1.1.25. V102_SpeedFlowCurves

05_Input_data\Input_data.mdb\V102_SpeedFlowCurves

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
NUM_CURVE	Function number	Integer	1
SPEEDBASE1	Base speed for first part of equation	Double	84.7
A1	A parameter for first part of equation	Double	0.29756424
B1	B parameter for first part of equation	Double	85.05170029
C1	C parameter for first part of equation	Double	0.01034036
D1	D parameter for first part of equation	Double	0.000069898342
G1	G parameter for first part of equation	Double	0.00058526
SPEEDBASE2	Base speed for second part of equation	Double	0
A2	A parameter for second part of equation	Double	0
B2	B parameter for second part of equation	Double	0
C2	C parameter for second part of equation	Double	0
D2	D parameter for second part of equation	Double	0
G2	G parameter for second part of equation	Double	0

Table 28 – Format of "V102_SpeedFlowCurves" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Define the delay functions based on the category of the link for vehicle class 102-105

Editing Options:: No need to be updated

Used by: "Road" subgroup under "Samgods Model\LOS Calculation", "Road Assignment" subgroup under "Samgods Model\Assignment", "Road Assignment" subgroup under "Samgods Model\RCM Assignment" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.2. Base data

Base data works in pair with scenario specific data to form a specific scenario.

The concept is that all the input data connected to a specific scenario is stored in the database as the differences between the current scenario and the base scenario (i.e., the reference scenario). The benefits of this are:

- A smaller amount of data is stored, that is, only the differences between the base scenario and the alternative scenario(s)
- It makes different alternative scenarios independent of each other, by letting the alternative scenarios having only the base scenario as a reference. The only redundancy of the system occurs when children are added to an alternative scenario since all scenarios (including child and parent scenarios) are independent of each other and only have the base scenario as a reference
- When changes are made to the base scenario, there is no need to change all the other scenarios. This facilitates the handling of the overall scenarios. Furthermore the changes and edits will be done only once, in the base scenario, avoiding risk of mistake when updating the correlated scenarios

To create the editable files for a specific scenario and the scenario specific tables, two applications work in conjunction:

- the *Create the editable files* application, that merges the base scenario with the scenario specific tables to view and edit a scenario, and
- the *Edit the data* application, that stores the scenario-specific tables into the database after any edits have been made

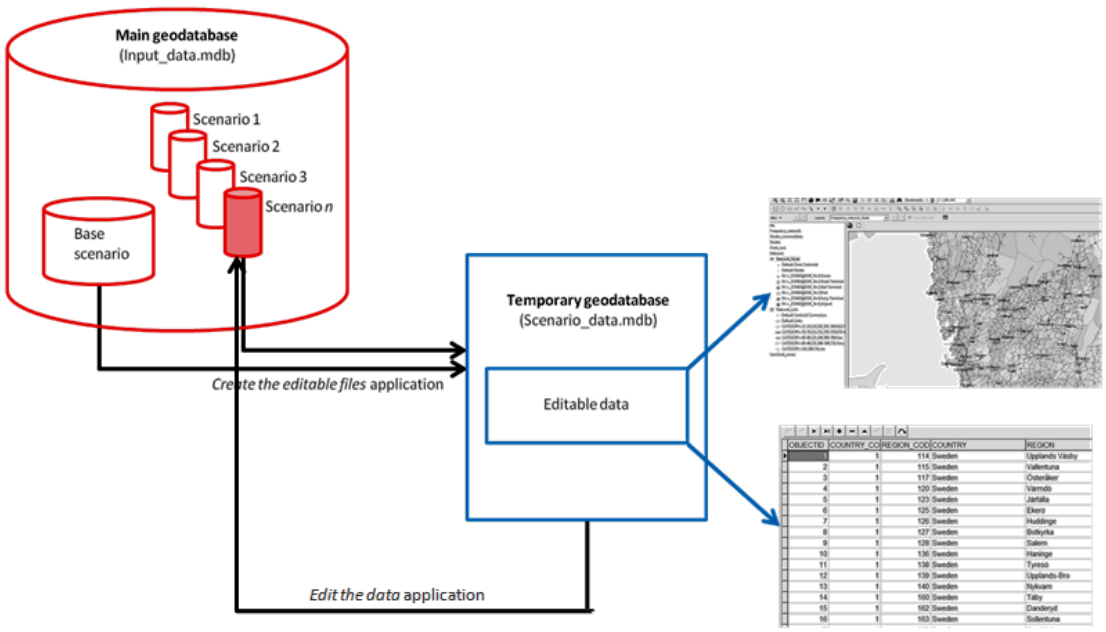


Figure 1 Database structure and scenario data of the Samgods GUI.

To create the editable file for a specific scenario and vice versa, create the scenario specific tables, two applications works in conjunction, the first one is “Create the editable files” with the purpose to merge the base scenario with the scenario specific table to consolidate a scenario, and “Edit the data” application, that after the edits stores the scenario specific table into the database.

The main database is always present in the model, under 05_Input_data folder; the temporary database is created in each run under the scenario folder and deleted on request by the user.

In this section we will discuss the tables and networks that form the base data, e.g. the reference scenario.

In the scenario folder all the base data and scenario data (e.g. differences with Base Scenario) are merged forming a new set of input data editable from interface. In the following table the correspondences among each feature class or table present in Input_data.mdb and Scenario_data.mdb are presented

Input_Data.mdb		**Scenario_data.mdb**	
Name	**Type**	**Name**	**Type**
Cargo_Base2006	Table	Cargo_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Cargo	Table		
Frequency_Data_Base2006	Network	Frequency_network	Network
Sc_{Scenario_short_name}_Frequency_Data_Link	Node Table		
Sc_{Scenario_short_name}_Frequency_Data_Node	Link Table		
General_Base2006	Table	General_{Scenario_short_name}	Table
LogMod_Base2006	Table	LogMod_{Scenario_short_name}	Table
Base2006	Network	Network	Network
Sc_{Scenario_short_name}_Link	Node Table		
Sc_{Scenario_short_name}_Node	Link Table		
Model_description	Table	Model_description	
Nodes_Base2006	Table	Nodes	Stand-alone Feature class points
Sc_{Scenario_short_name}_Nodes	Node Table		
Nodes_Commodities_Base2006	Table	Nodes_Commodities	Stand-alone Feature class points
Sc_{Scenario_short_name}_Nodes_commodities	Table		
Node_terminals_Base2006	Table	Ports_Swe	Stand-alone Feature class points
Sc_{Scenario_short_name}_Node_Terminals	Table		
Rail_Capacity_Base2006	Table	Rail_Capacity_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Rail_Capacity	Table		
SamGods_zones	Stand-alone Feature class polygons	SamGods_zones	Stand-alone Feature class polygons
Tax_Category_Base2006	Table	Tax_Category_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Tax_Category	Table		
Tax_Country_Base2006	Table	Tax_Country_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Tax_Country	Table		

Input_Data.mdb		Scenario_data.mdb	
Name	**Type**	**Name**	**Type**
Tax_Link_Base2006	Table	Tax_Link_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Tax_Link	Table		
Toll_Link_Base2006	Table	Toll_Link_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Toll_Link	Table		
Vehicles_exception_Base2006	Table	Vehicles_exception_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Vehicles_exc	Table		
Vehicles_parameters_Base2006	Table	Vehicles_parameters_{Scenario_short_name}	Table
Sc_{Scenario_short_name}_Vehicles_parameters	Table		

Table 29 – Correspondence between Input_data.mdb and Scenario_data.mdb.

The structure for each table is identical to the structure in Input_data.mdb with some exceptions:

- MODESTR, attribute present in the link table of the network, is converted into a set of attributes per link where the particular vehicle type is allowed/not allowed depending on the flag. For instance ' xc' is converted in two attributes ALL_M=1 and CAR=1. The correspondence is hold in Vehicle Parameters table under MODE_1 and MODE_2. Per each letter specified under MODESTR, a different attribute has specified. For further details refer to document listed under0.7
- Some new attributes are added to make the information more readable. For instance under Nodes_commodities a new field TRANSFER gives the information of which type of transfer the TRANSFER_TYPE number refers to. For instance TRANSFER_TYPE=1 means that TRANSFER= TransferRoadRoad

We refer to the user manual (document listed under0.7)for further details on the structure and rules for editing.

2.1.2.1. Base2006

05_Input_data\Input_data.mdb\Base2006

Base2006 represents the network for the cost calculation and assignment processes. Physically it is constructed from two tables, the link and the node table in the geodatabase.

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Count	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Integer	1
X	Coordinate x	Double	1620000
Y	Coordinate y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Integer	711400
SCBSTANN	SCBSTANN code	Integer	114
ID_COUNTRY	Country code	Integer	1
ID_REGION	Region code	Integer	114
MODE_N	Mode code	Integer	0
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Text	Upplands-Väsby

Table 30 – Format of "Base2006_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Count	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	1
B	End node	Integer	2659
MODESTR	String with all the allowed modes	Text	Xabc
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	50
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	50
CATEGORY	Link category	Double	110
FUNCTION	Index for travel time function	Double	61
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.4
UL3	Capacity for vessels on sea (dwell tons)	Double	0
Shape_Length	Length based on distance between coordinates in meters	Double	230.86792762

Table 31 – Format of "Base2006_Link" table.

Software required: Microsoft Access, Cube GIS Window (see Help>Cube Base>GIS window), ArcMap

Description of Use: Define the base network

Editing Options: No edits; all the changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: All the applications

2.1.2.2. Frequency_data_Base2006

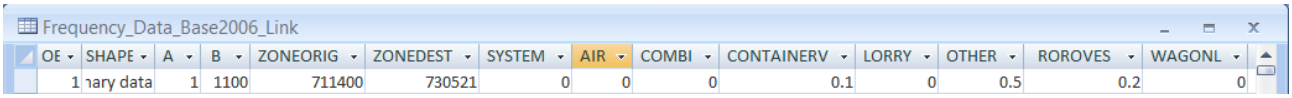
05_Input_data\Input_data.mdb\Frequency_data_Base2006

Frequency_data_Base2006 represents the frequency matrices for specific modes. Physically it is constructed from tables, the link and the node table in the geodatabase. The use of a network instead of a set of matrices is to save hard disk space (the network format is more compressed than the matrix format) and access time during the read/write phases.

Each relation between zones is represented by:

- a link, if the connection is between two different zones
- a node, if the connection is between the same zone (origin identical with destination)

For instance in the link table the following link:



Frequency_Data_Base2006_Link

OE	SHAPE	A	B	ZONEORIG	ZONEDEST	SYSTEM	AIR	COMBI	CONTAINERV	LORRY	OTHER	ROROVES	WAGONL
1	nary data	1	1100	711400	730521	0	0	0	0.1	0	0.5	0.2	0

will be translated into 3 rows in each file representing different frequencies. In the example:

- FREQCONTAINERVESSEL.314 will take the value 0.1 from CONTAINERV and origin 711400 from ZONEORIG and destination 730521 from ZONEDEST
- FREQOTHERVESSEL.314 will take the value 0.5 from OTHER and origin 711400 from ZONEORIG and destination 730521 from ZONEDEST

In each 314-file we will find the corresponding record:

File	Record
FREQCONTAINERVESSEL.314	711400 730521:0.1
FREQOTHERVESSEL.314	711400 730521:0.5

A similar process is carried out for the node table, where all the intrazonal frequencies are saved, i.e. values where origin is identical with destination. Values in node table with 0 in ZONEORIG_N and ZONEDEST_N represent that the intrazonal values are not available for that specific zone.

The correspondence between 314-files and attributes in the network is as follow:

Attribute in node table	Attribute in link table	File under Input\LOS
SYSTEM_N	SYSTEM	FREQSYSTEM.314
AIR_N	AIR	FREQAIR.314
COMBI_N	COMBI	FREQCOMBI.314
CONTAINE_N	CONTAINERV	FREQCONTAINERVESSEL.314
LORRY_N	LORRY	FREQLORRY.314
OTHER_N	OTHER	FREQOTHERVESSEL.314
ROROVES_N	ROROVES	FREQROROVESSEL.314
WAGONL_N	WAGONL	FREQWAGGONLOAD.314
RAILFERR_N	RAILFERRY	FREQRAILFERRY.314
ROADFERR_N	ROADFERRY	FREQROADFERRY.314

Table 32 – Correspondence between information in Frequency_network and LOS files under Input folder for logistics module.

Node Table format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	25
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Integer	25
X	Coordinate x	Double	1593100
Y	Coordinate y	Double	6562900
ZONEORIG_N	Original node number for the origin	Double	714000
ZONEDEST_N	Original node number for the destination	Double	714000
SYSTEM_N	Frequency System	Double	0
AIR_N	Frequency air	Double	0
COMBI_N	Frequency Combi	Double	0
CONTAINE_N	Frequency ContainerVessel	Double	0
LORRY_N	Frequency Lorry	Double	84
OTHER_N	Frequency Other Vessels	Double	0
ROROVES_N	Frequency RororVessel	Double	0
WAGONL_N	Frequency WaggonLoad	Double	0
RAILFERR_N	Frequency Railferry	Double	0
ROADFERR_N	Frequency RoadFerry	Double	0

Table 33 – Format of "Frequency_data_Base2006_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	1
B	End node	Integer	1100
ZONEORIG	Original node number for the origin	Double	711400
ZONEDEST	Original node number for the destination	Double	730521
SYSTEM	Frequency System	Double	0
AIR	Frequency air	Double	0
COMBI	Frequency Combi	Double	0
CONTAINERV	Frequency ContainerVessel	Double	0.1
LORRY	Frequency Lorry	Double	0
OTHER	Frequency Other Vessels	Double	0.5
ROROVES	Frequency RororVessel	Double	0.2
WAGONL	Frequency WaggonLoad	Double	0
RAILFERRY	Frequency Railferry	Double	0
ROADFERRY	Frequency RoadFerry	Double	0
Shape_Length	Length based on distance between coordinates in meters	Double	6796881.71149

Table 34 – Format of "Frequency_data_Base2006_Link" table.

Software required: Microsoft Access, Cube GIS Window (see Help>Cube Base>GIS window), ArcMap

Description of Use: Define the base network

Editing Options: No edits; all the changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: All the applications

2.1.2.3. Cargo_Base2006

05_Input_data\Input_data.mdb\Cargo_Base2006

CargoBase2006 represents the default values for the commodities, with all the parameters used in the logistics model.

Table format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_COM	Commodity code	Integer	1
DESC_COMM	Description of commodity	Text	Cereals
VALUE_SEKT	Value in SEK per ton	Integer	1350
INV_COST	Inventory cost	Integer	466
ORD_COST	Order cost	Double	1829
NSTR_NR	NSTR code	Integer	10
STAN_NR	STAN number	Integer	1
AGGR_COM	Commodity aggregation (three classes Dry bulk, General cargo, Liquid bulk)	Text	Dry bulk
COM_TYPE	Code for the commodity aggregation (1 Dry bulk, 2 Liquid bulk, 3 general cargo)	Integer	1
PRORCOST	ProportionalOrderCosts Parameter used for the calculation of the annual demand dependent order costs. The annual demand dependent order costs are calculated as: OrderCosts = FixedOrderCosts + ProportionalOrderCosts x AnnualDemand^Alpha	Double	2.19109
ALPHA	Parameter used in the order cost calculation	Single	0.5
SHIP_SIZE	Typical shipment size for this commodity. This parameter is used to determine the shipment size q [tonnes per shipment] and calculate the logistic costs for the chains in the BuildChain phase	Text	41
OPTIP	Code for including all costs or only transport costs for producers 0=include all costs 1=include only transport costs	Integer	0
OPTIW	Code for including all costs or only transport costs for wholesale. 0=include all costs 1=include only transport costs	Integer	0
MFREQ	Minimum frequency that is used when the default frequency is not available in the vehicle cost file	Integer	1
SELECT_BC	Handle the existence of the select.dat file for BuildChain (0- no, 1 –yes)	Integer	0
SELECT_CC	Handle the existence of the select.dat file for ChainChoi (0- no, 1 –yes)	Integer	0

Table 35 – Format of "Cargo_Base2006" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcCatalog

Description of Use: Defines the values in INPUT\COST\CARGO.TXT and some control parameters (SELECT_BC and SELECT_CC) in builchain.ctl and ChainChoi.ctl control files. See document listed under Section 0.4 for details

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Create the tables specific of scenario" subgroup under "Edit the data", "Data Preparation" subgroup under "Samgods Model\LOS calculation" and "Create the new base" subgroup under "Handling scenario\Scenario Export"

2.1.2.4. Node_terminals_Base2006

05_Input_data\Input_data.mdb\Node_terminals_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Count	Integer	9
N	Node number	Integer	211
TERMINAL_N	Code number for port	Integer	808321
PILOT_V301	Pilot fee for vehicle type 301 [SEK]	Double	4932
PILOT_V302	Pilot fee for vehicle type 302 [SEK]	Double	7054
PILOT_V303	Pilot fee for vehicle type 303 [SEK]	Double	8240
PILOT_V304	Pilot fee for vehicle type 304 [SEK]	Double	11757
PILOT_V305	Pilot fee for vehicle type 305 [SEK]	Double	2393
PILOT_V306	Pilot fee for vehicle type 306 [SEK]	Double	3412
PILOT_V307	Pilot fee for vehicle type 307 [SEK]	Double	3766
PILOT_V308	Pilot fee for vehicle type 308 [SEK]	Double	4224
PILOT_V309	Pilot fee for vehicle type 309 [SEK]	Double	5639
PILOT_V310	Pilot fee for vehicle type 310 [SEK]	Double	7054
PILOT_V311	Pilot fee for vehicle type 311 [SEK]	Double	8240
PILOT_V312	Pilot fee for vehicle type 312 [SEK]	Double	10592
PILOT_V313	Pilot fee for vehicle type 313 [SEK]	Double	10592
PILOT_V314	Pilot fee for vehicle type 314 [SEK]	Double	11757
PILOT_V315	Pilot fee for vehicle type 315 [SEK]	Double	7054
PILOT_V316	Pilot fee for vehicle type 316 [SEK]	Double	8240
PILOT_V317	Pilot fee for vehicle type 317 [SEK]	Double	9406

Table 36 – Format of "Node_terminals_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Pilot fees for Swedish ports per vehicle type. Values are exported to INPUT\COST\PILOTFEES.TXT

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: “Editable map data” subgroup under “Create the editable file”, “Compare Nodes and Node_terminals” subgroup under “Edit the data\Create the tables specific of scenario”, “Prepare data” subgroup under “Samgods Model\Logistics module", and “Create the new base” subgroup under “Handling scenario\Scenario Export”

2.1.2.5. Nodes_Base2006

05_Input_data\Input_data.mdb\Nodes_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	10
N	Zone number	Integer	10
ZONEID	Code number for zone	Integer	712500
NAME	Description of the zone/terminal	Text	Zone: Ekerö
ZONET	Zone code for node location if direct access, 0 other cases	Double	712500
DOMESTIC	Dummy for domestic location (1=domestic, 0=non domestic)	Integer	1
C_TECH_FAC	Cost efficiency factor in terminals	Integer	1
T_TECH_FAC	Time efficiency factor in terminals	Integer	1
MAXDWTCONT	Load constraint for container vessels that the port can handle [tonnes]	Double	0
MAXDWTRORO	Load constraint for RoRo vessels that the port can handle [tonnes]	Double	0
MAXDWOTHE	Load constraint for other vessels that the port can handle [tonnes]	Double	0
SEAOUTPUT	Yearly sea output[tonne]. Used for consolidation determination.	Double	0
SEACONTO	Yearly container output [tonne] Used for consolidation determination.	Double	0
AIROUTPUT	Yearly air output [tonne] Used for consolidation determination.	Double	0
TranOceFac	Penalty factor applied during the LOS calculation for TIME. The final time will penalized by a factor in the following manner: TIME (from the network)* (1+OrigFactor)*(1+DestFactor) where OrigFactor is TranOceFac for the origin and DestFactor is TranOceFac for the destination	Double	0
PortAreaNr	Port area number for the specific port. Values from A_Port_areas table	Double	0
VanernCan	Capacity associated to a port in Vänern Lake. Value represents dwell units allowed in the particular port. See Chapter 0 for further details.	Double	0

Table 37 – Format of "Nodes_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: definition for each zone/terminal of different parameters representing capacities and location in \Input\Nodes\NODES.TXT file. TranOceFac is explicitly used in the Samgods application generating the impedance factors per each origin and destination in LOS calculation (Matrix step #3 and HIGHWAY step #7) applied to TIME skim

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable map data" subgroup under "Create the editable files", "Compare Nodes and Node_terminals" subgroup under "Edit the data\Create the tables specific of scenario", "Data Preparation" subgroup under "Samgods Model\LOS calculation", "Prepare Data second" subgroup under "Samgods Model\Logistics Module" and "Create the new base" subgroup under "Handling scenario\Scenario Export"

2.1.2.6. Nodes_Commodities_Base2006

05_Input_data\Input_data.mdb\Nodes_Commodities_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Count	Integer	21
N	Zone number	Integer	21
ZONEID	Code number for zone	Integer	713611
TRANSF_TYP	Transfer type code (referred to A_Transfer_type table)	Integer	14
COM_1	Dummy representing whether a transfer of a certain type is allowed for listed terminals etc for commodity nbr 1	Integer	1
COM_2	As above for commodity 2	Integer	2
COM_3	As above for commodity 3	Integer	3
COM_4	As above for commodity 4	Integer	4
COM_5	As above for commodity 5	Integer	5
COM_6	As above for commodity 6	Integer	6
COM_7	As above for commodity 7	Integer	7
COM_8	As above for commodity 8	Integer	8
COM_9	As above for commodity 9	Integer	9
COM_10	As above for commodity 10	Integer	10
COM_11	As above for commodity 11	Integer	11
COM_12	As above for commodity 12	Integer	12
COM_13	As above for commodity 13	Integer	13
COM_14	As above for commodity 14	Integer	14
COM_15	As above for commodity 15	Integer	15
COM_16	As above for commodity 16	Integer	16
COM_17	As above for commodity 17	Integer	17
COM_18	As above for commodity 18	Integer	18
COM_19	As above for commodity 19	Integer	19

Field	Description	Data Format	Example
COM_20	As above for commodity 20	Integer	20
COM_21	As above for commodity 21	Integer	21
COM_22	As above for commodity 21	Integer	22
COM_23	As above for commodity 23	Integer	23
COM_24	As above for commodity 24	Integer	24
COM_25	As above for commodity 25	Integer	25
COM_26	As above for commodity 26	Integer	26
COM_27	As above for commodity 27	Integer	27
COM_28	As above for commodity 28	Integer	28
COM_29	As above for commodity 29	Integer	29
COM_30	As above for commodity 39	Integer	30
COM_31	As above for commodity 31	Integer	31
COM_32	As above for commodity 32	Integer	32
COM_33	As above for commodity 33	Integer	33
COM_34	As above for commodity 34	Integer	34
COM_35	As above for commodity 35	Integer	35

Table 38 – Format of "Nodes_Commodities_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Define per transfer type the list of terminals where at least one commodity may be subject to transfer. It is the source to produce files under Input\Nodes. Based on each transfer type.

For instance, in the table below the following records:

OBJECTID	N	ZONEID	TRANSF_TYP	COM_1	COM_2	COM_3	COM_4	COM_5	COM_6	CON
11	21	713611	3	1	2	3	4	5	6	
12	21	713611	14	1	2	3	4	5	6	
13	21	713611	16	0	0	0	0	0	0	

will be translated into 3 rows in each file representing different transfer types. In the example:

- TRANSFERROADTRAIN.TXT (TRANSF_TYPE=3) will be given the commodity number for all the commodities where COM_X<>0 and ZONEID=713611
- DIRECTFEEDERTRAIN.TXT (TRANSF_TYPE=14) will be given the commodity number for all the commodities where COM_X<>0 and ZONEID=713611
- DIRECTFEEDERTRAIN.TXT (TRANSF_TYPE=16) will be given the commodity number for all the commodities where COM_X<>0 and ZONEID=713611

In each txt-file we will find the corresponding records:

File	Record					
TRANSFERROADTRAIN.TXT	713611	Rail: Jordbro	1 5	2 6	3 7	4
FREQOTHERVESSEL.314	713611	Rail: Jordbro	1 5	2 6	3 7	4

DIRECTFEEDERTRAIN.TXT	713611	Rail: Jordbro 0	0 0	0 0	0	0

In the following table the correspondence between Transfer Type and related input file in the logistics module:

TRANSF_TYP	**DESCRIPTIO**	**File under Input\Nodes**
1	TransferRoadRoad	TRANSFERROADROAD.TXT
2	TransferRoadCombi	TRANSFERROADCOMBI.TXT
3	TransferRoadTrain	TRANSFERROADTRAIN.TXT
4	TransferRoadSea	TRANSFERROADSEA.TXT
5	TransferRoadRoadFerry	TRANSFERROADROADFERRY.TXT
6	TransferRoadAir	TRANSFERROADAIR.TXT
7	TransferCombiSea	TRANSFERCOMBISEA.TXT
8	TransferFeederTrainWagonload	TRANSFERFEEDERTRAINWAGONLOAD.TXT
9	TransferWagonloadSea	TRANSFERWAGONLOADSEA.TXT
10	TransferWagonloadRailFerry	TRANSFERWAGONLOADRAILFERRY.TXT
11	TransferSystemTrainSea	TRANSFERSYSTEMTRAINSEA.TXT
12	TransferSeaSea	TRANSFERSEASEA.TXT
13	ContainerHandling	CONTAINERHANDLING.TXT
14	DirectFeederTrain	DIRECTFEEDERTRAIN.TXT
15	DirectSea	DIRECTSEA.TXT
16	DirectSystemTrain	DIRECTSYSTEMTRAIN.TXT
17	DirectWagonLoad	DIRECTWAGONLOAD.TXT

Table 39 – Correspondence between file system and Nodes_commodities table.

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: “Editable map data” subgroup under “Create the editable files”, “Compare other tables” subgroup under “Edit the data”, “Prepare Data second part” subgroup under “Samgods Model\Logistics Module” and “Create the new base” subgroup under “Handling scenario\Scenario Export”

2.1.2.7. PropLink_Base2006

05_Input_data\Input_data.mdb\PropLink_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	696
A	Start node	Double	2793
B	End node	Double	2794
REGION	SCBSTANN code	Double	136
WEIGHT	Weight	Double	100

Table 40 – Format of “PropLink_Base2006” table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Table with all road and rail links that share a different SCBSTANN area. Each record provides the proportion of the link length belonging to the listed SCBSTANN area. The proportion is used to classify tonneskm and vehicleskm statistics per region and county. (See reports #9, 16, 17 ,18)

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: “Reports per geographical aggregation” subgroup under “Samgods Model\Results\Results 1.0”, “Samgods Model\Results RCM\Results 1.0”, “Create the new base” subgroup under “Handling scenario\Scenario Export” and “Create the scenario tables” subgroup under “Handling scenario\Scenario Export\Create the new scenario specific tables”

2.1.2.8. Rail_Capacity_Base2006

05_Input_data\Input_data.mdb\Rail_Capacity_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_LINK	Identification code for the link	Double	1
A	Starting voyager node	Double	2000
B	Ending voyager node	Double	2001
ORIGCAP	Original value for capacity in bidirectional trains per day	Double	418

Table 41 – Format of “Rail_Capacity_Base2006” table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: This table represents the main input for Rail Capacity Management module, since the number of total vehicles on rail links is compared with the ORIGCAP value. The links with ratio V/C equals or greater than the value in catalog key " Cut off criteria for RCM process (will consider links with V/C > XX% where XX is the value defined)" will be selected as part of the linear programming problem. For details on edits of this table we refer to User Manual 0.7). It is also a mandatory input for the Rail Assignment under "Samgods\RCM Assignment\Rail Assignment RCM" since in this case the assignment is based on capacity constraints, e.g. the delay functions are related to the ratio V/C

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Rail Capacity Checks" subgroup under "Edit the data\Create the tables specific of scenario", "Data Preparation" subgroup under "Samgods Model\LOS calculation", "Create the new base" subgroup under "Handling scenario\Scenario Export" and "Create the scenario tables" subgroup under "Handling scenario\Scenario Export\Create the new scenario specific tables"

2.1.2.9. Tax_Category_Base2006

05_Input_data\Input_data.mdb\Tax_Category_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	17
LINKTYPE	Link category	Double	580
ID_VEHICLE	Vehicle type	Double	317
SEK	Track fee or tax [SEK]	Double	53700

Table 42 – Format of "Tax_Category_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Specifies a link based tax defined by the link category and vehicle type. In the current model the only tax applied is for Kiel Canal (CATEGORY=570)

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Compare other tables" subgroup under "Edit the data\Create the tables specific of scenario", "Tax calculation" subgroup under "Samgods Model\LOS calculation\Data Preparation" and "Create the new base" subgroup under "Handling scenario\Scenario Export"

2.1.2.10. Tax_Country_Base2006

05_Input_data\Input_data.mdb\Tax_Country_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_COUNTRY	Country code	Double	-1
ID_VEHICLE	Vehicle type	Double	201
SEK_KM	Country tax per km [SEK/km]	Double	4.47

Table 43 – Format of "Tax_Country_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Specifies a link based tax defined by country and vehicle type. (NB. -1 values refer to countries outside Sweden).

The current definition per country is:

ID COUNTRY	COUNTRY	101	102	103	104	105	201	202	204	205	206	207	208	209
1	Sweden						4.47	2.68	4.47	5.2	24.39	3.26	4.47	5.63
516	Check Republic						40.23	24.12	40.23	46.8	219.51	29.34	40.23	50.67
517	Poland						64.815	38.86	64.815	75.4	353.655	47.27	64.815	81.635
518	Norway						6.705	4.02	6.705	7.8	36.585	4.89	6.705	8.445
519	Denmark						35.76	21.44	35.76	41.6	195.12	26.08	35.76	45.04
520	Germany		1.08	1.17	1.17	1.17	29.055	17.42	29.055	33.8	158.535	21.19	29.055	36.595
521	Austria	1.395	1.953	2.9295	2.9295	2.9295	35.76	21.44	35.76	41.6	195.12	26.08	35.76	45.04
522	Switzerland	3.692	3.692	3.692	6.039	6.039								
527	Netherlands						7.8225	4.69	7.8225	9.1	42.6825	5.705	7.8225	9.8525
-1	All other countries						4.47	2.68	4.47	5.2	24.39	3.26	4.47	5.63

Table 44 – Values specified in the model.

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Compare other tables" subgroup under "Edit the data\Create the tables specific of scenario", "Tax calculation" subgroup under "Samgods Model\LOS calculation\Data Preparation" and "Create the new base" subgroup under "Handling scenario\Scenario Export"

2.1.2.11. Tax_Link_Base2006

05_Input_data\Input_data.mdb\Tax_Link_Base2006

Table Format:

Field	Description	Data Format	Example

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	4
A	Start node	Double	1007
B	End node	Double	1009
VEH_NR	Vehicle class	Double	101
SEK	Track fee or tax [SEK]	Double	18

Table 45 – Format of "Tax_Link_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Specifies a link based tax by vehicle type

Editing Options: No edits; all the Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Compare other tables" subgroup under "Edit the data\Create the tables specific of scenario", "Tax calculation" subgroup under "Samgods Model\LOS calculation\Data Preparation" and "Create the new base" subgroup under "Handling scenario\Scenario Export"

NOTE: For the same link it is possible to define tax values based on country criteria, category criteria and also its selection (three tables above). However the final result will be based on the following rule:

TAX_COUNTRY < TAX_CATEGORY < TAX_LINK

The value in the Tax_Link table will override the other values, so the final result will come from the evaluation of an OR logical condition. Zero values also have a meaning, inserting a zero in the table value will set the tax to zero. For instance 0 in the Tax_Link table will set the tax value for the specific link to zero.

2.1.2.12. Toll_Link_Base2006

05_Input_data\Input_data.mdb\Toll_Link_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	4
A	Start node	Double	2001
B	End node	Double	25742
VEH_NR	Vehicle class	Double	205
SEK	Link toll [SEK] per passing vehicle	Double	5070
DESCRIPT	Description of the infrastructure (in general bridges)	Text	Toll Oresund

Table 46 – Format of "Toll_Link_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Defines a toll applied to a link per vehicle type. The current values in the model refer to:

Toll [SEK]	101	102	103	104	105	201	202	204	205	206	207	208	209
Oresund	650	960	960	960	960	5070	5070	5070	5070	5070	5070	5070	5070
Stora Belt	620	985	985	985	985	6710	6710	6710	6710	6710	6710	6710	6710
Svinesund	20	100	100	100	100								

Table 47 – Values specified in the model.

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Compare other tables" subgroup under "Edit the data\Create the tables specific of scenario", "Tax calculation" subgroup under "Samgods Model\LOS calculation\Data Preparation" and "Create the new base" subgroup under "Handling scenario\Scenario Export"

2.1.2.13. Vehicles_exception_Base2006

05_Input_data\Input_data.mdb\Vehicles_exception_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
VEH_NR	Vehicle class	Double	205
DESCRIPTIO	Vehicle class description	Text	System train STAX 25
LABEL	Label for vehicle class	Text	SYS25
KMCOST_17	Km cost [SEK/km] for Commodity 17	Double	104.09
HRCOST_17	Hour cost [SEK/h] for Commodity 17	Double	3094
KMCOS10S11	Km cost [SEK/km] for STAN group 10 and 11	Double	186.2
NCLCD10S11	Non container Load Cost Dry Bulk for STAN groups 10 and 11 (as parameter NC_LCO_DRY in Vehicle_parameters_Base2006)	Double	-1
NCLTD10S11	Non container Load Time Dry Bulk for STAN groups 10 and 11 (as parameter NC_LTI_DRY in Vehicle_parameters_Base2006)	Double	-1
COLCO10S11	Container Load COST for STAN groups 10 and 11 (as parameter CONT_LCO in Vehicle_parameters_Base2006)	Double	-1
COLTI10S11	Container Load Time for STAN groups 10 and 11 (as parameter CONT_LTI in Vehicle_parameters_Base2006)	Double	-1
CAP_15	Maximum Load Capacity for commodity 15	Double	1098

Table 48 – Format of "Vehicles_exception_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: In the current setup some special values have been specified for STAN product groups or commodities. This table contains all the differences between the default values present in Vehicles_parameters_Base2006 table and specific values per defined vehicle types. For those values that are equal to -1, the values will be derived from Vehicles_parameters_Base2006. For those values greater or equal to zero, the value will come from this table. For details on editing on this table see User Manual Chapter 5.5 (document listed under Section 0.7). For details on how different values per different commodities will be set up see Chapter0

Editing Options: No edits; all the changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Compare other tables" subgroup under "Edit the data\Create the tables specific of scenario" and "Prepare data" subgroup under "Samgods Model\Logistics Module"

2.1.2.14. Vehicles_parameters_Base2006

05_Input_data\Input_data.mdb\Vehicles_parameters_Base2006

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	4
ID	Index for vehicle class	Double	4
VEH_NR	Vehicle number	Double	104
DESCRIPTIO	Description of vehicle type	Text	Lorry HGV 25-40 tonne
LABEL	Short name for the vehicle type	Text	HGV24
KM_COST	Km cost [SEK/km]	Double	3.7794
HOURS_COST	Hour cost [SEK/h]	Double	288
NC_LCO_DRY	Non container Load Cost Dry Bulk	Double	3
NC_LCO_LIQ	Non container Load Cost Liquid Bulk	Double	3
NC_LCO_GC	Non container Load Cost General Cargo	Double	20.7
DFLTFREQ	Default frequency (per week)	Double	84
NC_LTI_DRY	Non container Load Time Dry Bulk	Double	2
NC_LTI_LIQ	Non container Load Time Liquid Bulk	Double	2
NC_LTI_GC	Non container Load Time General Cargo	Double	2
CONT_LCO	Container Load Cost	Double	17
CONT_LTI	Container Load Time	Double	1
F_DUES_VH	Fairway Dues per Vehicle	Double	0
F_DUES_TON	Fairway Dues per Tonne	Double	0
CAPACITY	Maximum Load Capacity	Double	28
VESSELTYPE	Vessel Type (container/non container)	Double	0
ONFER_H_C	On Ferry Hour Cost	Double	488.4
ONFER_KM_C	On Ferry Km Cost	Double	2.9
POSICOST	Positioning Cost	Double	0

SPEED	Speed (only sea modes)	Double	-1
VDF_SPEC	Volume delay function index	Double	64
MODE_1	Code for network mode alternative 1	Double	a
MODE_2	Code for network mode alternative 2	Double	-
FUNC_FILE	Set of applied travel time functions applied	Double	V102
EMPTY_V	Dummy variable to take in account empty vehicles during the creation of OD vehicle matrices. (0=no 1=yes)	Integer	1
COORFACT	Coordination factor. Factor used to catch the fact that the available volume for each vehicle movement will be lower than the calculated annual consolidation volumes on the OD-legs	Single	10

Table 49 – Format of "Vehicles_parameters_Base2006" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Defines all the parameters related to cost and time coefficients for the logistics model (files under LogMod\Input\COST). Also controls the codes used in opening/closing links in the network (parameters SPEED, VDF_SPEC, MODE_1, MODE_2, FUNC_FILE) during LOS and assignment steps

Editing Options: No edits; Changes will be allowed only in the export phase merging the scenario specific tables and the base tables

Used by: "Editable tables" subgroup under "Create the editable files", "Create the tables specific of scenario" subgroup under "Edit the data" and its subgroup "Compare other tables", "Data Preparation" subgroup under "Samgods Model\LOS calculation" and "Create the new base" subgroup under "Handling scenario\Scenario Export"

2.1.3. Scenario specific data

In term of structure all the tables listed above have a copy in the scenario specific data. So we will have the following relation:

Base Table	**Scenario Specific Table**
Base2006	Sc_{SCENARIO_SHORTNAME}_Link
	Sc_{SCENARIO_SHORTNAME}_Node
Cargo_Base2006	Sc_{SCENARIO_SHORTNAME}_Cargo
Frequency_Data_Base2006	Sc_{SCENARIO_SHORTNAME}_Frequency_Data_Link
	Sc_{SCENARIO_SHORTNAME}_Frequency_Data_Node
Node_terminals_Base2006	Sc_{SCENARIO_SHORTNAME}_Node_Terminals
Nodes_Base2006	Sc_{SCENARIO_SHORTNAME}_nodes
Nodes_Commodities_Base2006	Sc_{SCENARIO_SHORTNAME}_nodes_commodities
PropLink_Base2006	Sc_{SCENARIO_SHORTNAME}_PropLink
Rail_Capacity_Base2006	Sc_{SCENARIO_SHORTNAME}_Rail_Capacity
Tax_Category_Base2006	Sc_{SCENARIO_SHORTNAME}_Tax_Category
Tax_Country_Base2006	Sc_{SCENARIO_SHORTNAME}_Tax_Country
Tax_Link_Base2006	Sc_{SCENARIO_SHORTNAME}_Tax_Link
Toll_Link_Base2006	Sc_{SCENARIO_SHORTNAME}_Toll_Link

Vehicles_exception_Base2006	Sc_{SCENARIO_SHORTNAME}_Vehicles_exc
Vehicles_parameters_Base2006	Sc_{SCENARIO_SHORTNAME}_Vehicles_Parameters

Table 50 – Relation between base and scenario specific data.

All the tables (except ProprLink and Vehicles_Parameters) have an extra attribute UP_DATE or UP_DATE_N that will record the differences between the scenario and the reference case (Base_2006).

This field, with test format, is used to store the information regarding how to update the base data. The following values are used

- U- update an existing value in the base table;
- A- append a new record to the base table;
- D – delete a record from the base table.

These values are automatically handled by the applications "Edit the data". The user must not type, modify, or change them. For ProprLink and Vehicles_Parameters all the elements present in these table will overwrite those in the reference case, e.g. a simpler substitution. There are two other tables that are handled differently from the general rule: General_{SCENARIO_SHORTNAME} and LogMod_{SCENARIO_SHORTNAME}. These are scenario specific and are not handled through the general difference record keeping. Those tables hold the definition of several control parameters used in the Logistics Module and in the GUI. They contain parameters that are controlled by catalog keys and could vary between scenarios.

2.1.3.1. General_{SCENARIO_SHORTNAME}

05_Input_data\Input_data.mdb\General_{SCENARIO_SHORTNAME}

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SC_N	Scenario number (1-default, higher if saved also in emmebank)	Integer	1
DATE_C	Date and time of creation	Text	14/08/2013 09:24
DATE_L	Date and time of last modification if the scenario is in read/write mode, otherwise the date of lock.	Text	14/08/2013 11:01
ZONES_M	Number of zones	Integer	1120
GCOST	Attribute name for extra cost on links [SEK]	Text	EC_V
GCOST_KM	Attribute name for extra cost per km on links [SEK/km]	Text	EC_KM_V
DF	Daily factor for all the modes except rail mode [number of days per year]	Integer	365
DFR	Daily factor for rail mode [number of days per year]	Integer	250
PHF	Peak Hour Factor	Single	0.119999997318
READ_WRITE	State of scenario (only read or read/write Yes=only read)	Text	Yes

Table 51 – Format of "General_{SCENARIO_SHORTNAME}" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: This table responds to a high variety of uses:

- SC_N is used to recode where the base network was saved in the emmebank (1 if not, higher is saved). Since EMME is not used in the model anymore, it will always be 1
- DATE_C and DATE_L are the time references to check when updating existing temporary data in the model. They are also used for long time storage purposes
- ZONE_M, GCOST and GCOST_KM refer to this kind of attributes saved in the network
- DF,DFR and PHF: general parameters for assignment
- READ_WRITE: state variable to prevent further edits of a scenario, or to leave the scenario as editable.

Editing Options: The first two points listed above and the last one are handled by the system, never changed by the user. The others (ZONE_M, GCPST, GCOST_KM,DF,DFR,PHF) are modifiable from catalog keys present in the interface under "Edit the data" application

Used by: “Prepare temporary data” subgroup under “Create the editable files”, “Create the tables specific of scenario” subgroup under “Edit the data”, “PWC_Matrices” and several subgroups under “Samgods Model”

2.1.3.2. LogMod_{SCENARIO_SHORTNAME}

05_Input_data\Input_data.mdb\LogMod_{SCENARIO_SHORTNAME}

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
INTER_RATE	Parameter INTEREST, interest rate used in cost calculations [%/year]	Double	0.1
STUFF	Parameter STUFF, costs for stuffing and stripping of containers at the origin and destination of a chain [SEK per tonne]	Double	18
CONTYPE	Parameter ALL_LORRY_TYPE_CONSOL, 0/1 switch that determines whether or not consolidation is allowed for all lorry types	Integer	1
INODLO	Parameter INDIVIDUAL_OD_LEG_OPTIMIZE, 0/1 switch that determines whether or not the optimization is done for each chain leg individually in a transport chain	Integer	1
MATD4FO	Parameter MINIMUM_ANNUAL_TONNE_DEMAND_4_FREQ_OPTIMIZE [Tonnes], Minimum demand for frequency optimization	Double	100
CONSOL_L	First value of parameter CONSOL. It sets the default lower bound for	Double	0.05

	the consolidation levels.		
CONSOL_U	Second value of parameter CONSOL. It sets the default upper bound for the consolidation levels.	Double	0.95
TONNES	Parameter TONNES. If this parameter is a number, then this number will be used as the typical shipment size for this commodity. If this the parameter is set to DYNAMIC_MAX, DYNAMIC_AVERAGE or DYNAMIC_GEOMEAN the typical shipment size will be different for different zones and calculated as the maximum, average or geometric mean of the PWC matrix-values for the current origin zone	Text	DYNAMIC_AVERAGE
LSTCNT	Parameter LSTCNT It sets the number of output files: 1= only best chains; 2 = best chains and second best chains. The current default is 5. It is recommended to not change this default since it is related to Rail Capacity Management module.	Integer	5
DATA	Parameter DATA, Output variables for the optimal transport solution [e.g.1,2,3, max is 8]. Nbrs 6 and 7 are useful when analysing detail in the results, since all relevant data for the transport chains are available. The cost data are split into different components, mainly operational costs, loading/unloading costs and infrastructure costs.	Text	6,7
FACTOR	Parameter FACTOR, Initial consolidation factor in BuildChain (1st iteration)	Single	0.75
LOGCTL	Parameter LOGCTL, indicator (0/1) that determines whether or not CTL file settings will be logged in the common log file for the commodity	Integer	1
LOGFLS	Parameter LOGFLS, indicator (0/1) that determines whether or not input file information will be logged in the common log file for the commodity	Integer	1
LOGCST	Parameter LOGCST, indicator (0/1) that determines whether or not cost parameters will be in the common log file for the commodity	Integer	1

Table 52 – Format of "LogMod_{SCENARIO_SHORTNAME}" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: This table is used in setting up the control files for BuildChain, ChainChoi, BuildChainRCM and ChainChoiRCM. All the parameters listed above control the process both in terms of reporting (log files) and settings for some variables. See document listed under Section 0 for their meaning and usage

Editing Options: In "Edit the data" application each parameter is present as catalog key. Its value could be revised directly from the Scenario Window. For further details refer to "User Manual"

Used by: "Prepare temporary data" subgroup under "Create the editable files", "Prepare Data third part" subgroup under "Samgods Model\Logistics Module", "Delete" and "Scenario Import" subgroups under "Handling scenario" and "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.4. Manage database

2.1.4.1. History

05_Input_data\Input_data.mdb\History

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
MODEL	Folder of the original model used to create the new model (father model)	Text	C:\SamGods_V10
BASE_SCE	Name of Base scenario in original model	Text	Base2006

Table 53 – Format of "History" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: For setting up new models (new model is created each time it is required to modify some general settings or input file such as chain types, etc). The table maintains the information of the father model used to setup the new model. This table traces the relationships among different versions.

Editing Options: Always changed by the system

Used by: "General tables" subgroup under "Handling scenario\Scenario Export"

2.1.4.2. Model_description

05_Input_data\Input_data.mdb\Model_description

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
NAME	Software name	Text	Cube
REALISE	Software version	Text	6.1.0

Table 54 – Format of "Model_description" table.

Software Required: Microsoft Access, Cube Database Window (see Help -> Cube Base -> Database window), ArcMap, ArcCatalog

Description of Use: Defines the release version for each program used in the model. This information could allows be checked regarding whether the correct version of a particular model is used.

Editing Options: No edits; always changed by the system

Used by: "Installation", "General tables" subgroup under "Handling scenario\Scenario Export" and "Scenario Import" subgroup under "Handling scenario"

Calibration data

The calibration data is used when running the model in calibration mode. When using the model as standard or advance user this data is not modified along the run. It represents the calibrated conditions that should be used in evaluating different scenarios. and represents an input an invariant input for the model.

Port Area parameters per STAN group

05_Input_data\Calibration\PortAreasParameters_22092014_L1_Base2006R21_RUN20.txt

Table Format:

Field	Description	Data Format	Example
PortArea	Port Area number	Integer	1.0
STAN1	Scaling factor for STAN group 1	Long	0.02
STAN2	Scaling factor for STAN group 2	Long	0.02
STAN3	Scaling factor for STAN group 3	Long	1.00
STAN4	Scaling factor for STAN group 4	Long	0.02
STAN5	Scaling factor for STAN group 5	Long	0.02
STAN6	Scaling factor for STAN group 6	Long	2.22
STAN7	Scaling factor for STAN group 7	Long	0.90
STAN8	Scaling factor for STAN group 8	Long	1.40
STAN9	Scaling factor for STAN group 9	Long	0.88
STAN10	Scaling factor for STAN group 10	Long	0.02
STAN11	Scaling factor for STAN group 11	Long	1.00
STAN12	Scaling factor for STAN group 12	Long	1.00

Table 55 – Format of "*PortAreasParameters.txt*" table.

Software Required: Notepad

Description of Use: Defines the scaling factor applied in TIME skim per port area and STAN group.

Editing Options: No edits. Results from the calibration procedure

Used by: "Samgods\Logistics Module\Port Area LOS inputs (STAN)", "General tables" subgroup under "Handling scenario\Scenario Export" and "Scenario Import" subgroup under "Handling scenario"

Parameters_Kielcalibration.dbf

05_Input_data\Calibration\ Parameters_Kielcalibration.dbf

Table Format:

Field	Description	Data Format	Example
ID	ID record	Integer	1.0
STEP	Step length in increasing or decreasing the Kiel canal scaling factor (starting from 1, 1+STEP or 1-STEP depending the search function)	Double	0.02
MINVAL	Minimum boundary for scaling factor. If after adjustment the	Double	0.01

	value is below this value, it will be reset to VAL0		
DIFMAX	Maximum difference between modelled and surveyed value. In this case the distribution of tons between Kiel Canal and Jylland. 2 is in percentage	Double	2
VAL0	Reset value used in case the procedure gives negative or less than MINVAL results.	Double	0.02

Table 56 – Format of "*Parameters_Kielcalibration.dbf*" table.

Software Required: Cube Data Window, Dbf editor

Description of Use: Defines the control parameters in the adjust procedure for the Kiel Canal scaling factor.

Editing Options: No edits. Inputs for the calibration procedure

Used by: "Samgods\Parameters calculation", "General tables" subgroup under "Handling scenario\Scenario Export" and "Scenario Import" subgroup under "Handling scenario"

Parameters_portcalibration.dbf

*05_Input_data**Calibration*\ *Parameters_portcalibration.dbf*

Table Format:

Field	Description	Data Format	Example
ID	ID record	Integer	1.0
STEP	Step length in increasing or decreasing the Kiel canal scaling factor (starting from 1, 1+STEP or 1-STEP depending on the search direction)	Double	0.02
MINVAL	Minimum boundary for scaling factor. If after adjustment the value is below this value, it will be reset to VAL0	Double	0.01
DIFMAX	Maximum difference between modelled and surveyed value. In this case it is the total tonne throughput per port area and STAN group. (100*1000 tons since statistics are tons/1000)	Double	100
VAL0	Reset value used in case the procedure gives negative or less than MINVAL results.	Double	0.02

Table 57 – Format of "*Parameters_portcalibration.dbf*" table.

Software Required: Cube Data Window, Dbf editor

Description of Use: Defines the control parameters in the scaling factor adjustment procedure for Port Areas per STAN.

Editing Options: No edits. Inputs for the calibration procedure

Used by: "Samgods\Parameters calculation", "General tables" subgroup under "Handling scenario\Scenario Export" and "Scenario Import" subgroup under "Handling scenario"

Output data

The output data can be split into two classes; the first one is the output that always will be generated by the model independently of the user choices, the second one is only generated on user request.

General tables in geodatabase

Model description

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Model_description

Table format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
NAME	Software name	Text	Cube
REALISE	Software version	Text	6.1.0

Table 58 – Format of "Model_description" table.

Description of Use: Defines the release version for each program involved in the model. It is used for long-term management of the model.

Visualizing Options: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Used by: Not used

Made by user choice: No

Outputs

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\outputs

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	2
OUTPUTR	Description of the output	Text	LOS_ROAD
ID_COM	Commodity code (0 for all)	Doublet	0
DATE_E	Date of last modification for the input data used to create this output	Text	03-Feb-15 07:17 PM

Table 59 – Format of "outputs" table.

Description of Use: Storage of some input data settings and some output data for the last model runs.
Visualizing Options: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Created by: "Prepare temporary data" subgroup under "Create the editable files"

Updated by: several subgroups in the "Samgods Model" application

Made by user choice: No

Node_labels

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Node_labels

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
N	Node number	Integer	1
NORIG	Emme number (from numbering system)	Integer	711400
SCBSTANN	SCBSTANN code	Integer	114
ID_COUNTRY	Country code	Integer	1
ID_REGION	Region code	Integer	114
MODE_N	Mode for access to terminals	Integer	0
UI4	User field (not used in the current model)	Integer	0
X	X coordinate	Integer	1620000
Y	Y coordinate	Integer	6601000
CENTRALL	Description	Text	Upplands-Väsby

Table 60 – Format of "Node_labels" table.

Description of Use: Table with labels for each zone in the network. For reporting purposes and inside the model for mapping of data between Voyager and EMME node numbering systems.

Visualizing Options: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Created by: "Data Preparation" subgroup under "Samgods Model\LOS calculation"

Used by: "Change matrix format" application

Made by user choice: No.

ProprLink_{SCENARIO_SHORTNAME}

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\ProprLink_{SCENARIO_SHORTNAME}

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	696
A	Start node	Integer	2793

B	End node	Integer	2794
REGION	SCBSTANN code	Integer	136
WEIGHT	Weight	Double	100

Table 61 – Format of "ProprLink_{SCENARIO_SHORTNAME}" table.

Software Required: Microsoft Access or Cube Database window (see Help>Cube Base>Database window), ArcCatalog

Description of Use: Table with all road and rail links that are distributed in different SCBSTANN areas. Each record gives the proportion of the length of the link part residing in each SCBSTANN area. The proportion is used to estimate tonne km and vehicle km statistics per region and county. (See reports #9, 16, 17 ,18).

Used by: "Reports per geographical aggregation" subgroup under "Samgods Model\Results\Results 1.0", "Samgods Model\Results RCM\Results 1.0", "Create the new base" subgroup under "Handling scenario\Scenario Export" and "Create the scenario tables" subgroup under "Handling scenario\Scenario Export\Create the new scenario specific tables"

Network format in geodatabase

Load_net_Road_0 and Load_net_Road_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Road_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Road_RCM

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Integer	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Mode for access to terminals	Double	0
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 62 – Format of "Load_net_Road_0_Node" and "Load_net_Road_RCM_Node" tables.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	16
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	16
B	End node	Integer	2813

Field	Description	Data Format	Example
Shape_Length	Length based on distance between coordinates in meters	Double	1359.1541
SPEED_1	Vehicle speed in kms per hour for all vehicles except v102-105 in Sweden	Double	50
SPEED_2	Vehicle speed in kms per hour for vehiclesv102-105 in Sweden	Double	50
CATEGORY	Link category	Double	201
FUNCTION	Index for the travel time function	Double	61
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in kms. User link data in emme used to enable holding link lengths longer than 999 kms.	Double	1.36
ID_COUNTRY	Country code	Double	1
MODE_L	Code for allowed vehicles on the link	Double	1
DISTANCE	Distance in kms	Double	1.36
SWEDEN	Flag to classify a link as residing in Sweden or not	Integer	1
LGV3	Yearly vehicle 101 flow (loaded)	Double	3
LGV3T	Yearly vehicle 101 tonne flow	Double	2.1511
LGV3E	Yearly vehicle 101 flow (empty)	Double	0.5
MGV16	Yearly vehicle 102 flow (loaded)	Double	0
MGV16T	Yearly vehicle 102 tonne flow	Double	0
MGV16E	Yearly vehicle 102 flow (empty)	Double	0
MGV24	Yearly vehicle 103 flow (loaded)	Double	0
MGV24T	Yearly vehicle 103 tonne flow	Double	0
MGV24E	Yearly vehicle 103 flow (empty)	Double	0
HGV40	Yearly vehicle 104 flow (loaded)	Double	0
HGV40T	Yearly vehicle 104 tonne flow	Double	0
HGV40E	Yearly vehicle 104 flow (empty)	Double	0

Field	Description	Data Format	Example
HGV60	Yearly vehicle 105 flow (loaded)	Double	0.17989
HGV60T	Yearly vehicle 105 tonne flow	Double	0.8457
HGV60E	Yearly vehicle 105 flow (empty)	Double	0.50340
TOT_ROAD	Yearly vehicle flow (loaded vehicles) for the road mode	Double	3.17989
TOT_ROAD_T	Yearly tonne flow for the road mode	Double	2.9968
TOT_ROAD_E	Yearly vehicle flow (empty vehicles) for the road mode	Double	1.00340

Table 63 – Format of "Load_net_Road_0_Link" and "Load_net_Road_RCM_Link" tables.

Description of Use: Assigned network for the road mode for standard logistic module and Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Road Assignment" subgroup under "Samgods Model\Assignment" and "Road Assignment RCM" subgroup under "Samgods Model\RCM Assignment"

Used by: "Results" and "Results RCM" subgroups under "Samgods Model".

Made by user choice: No

Load_net_Rail_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Rail_0

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114

Field	Description	Data Format	Example
ID_REGION	Region code	Double	114
MODE_N	Mode for access to terminals	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 64 – Format of "Load_net_Rail_0_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	21
B	End node	Integer	2097
Shape_Length	Length based on distance between coordinates in meters	Double	206.1552
ORIGINALCAP	Capacity on rail links from Capacity table (trains per day)	Double	0
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	50
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	0
CATEGORY	Link category	Double	211
FUNCTION	Index for travel time function	Double	66
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.2
ID_COUNTRY	Country code	Double	1
MODE_L	Code for allowed vehicles on the link	Double	2
SWEDEN	Flag for classify a link in Sweden or outside	Integer	1
CLOSE_	Flag 0/1 to treat vehicles 202/203	Double	0
KOMBI	Yearly vehicle 201 flow (loaded)	Double	0

Field	Description	Data Format	Example
KOMBIT	Yearly vehicle 201 tonne flow	Double	0
KOMBIE	Yearly vehicle 201 flow (empty)	Double	0
FS_TRAIN	Yearly vehicle 202 flow (loaded)	Double	325.2379
FS_TRAINT	Yearly vehicle 202 tonne flow	Double	166064.421875
FS_TRAINE	Yearly vehicle 202 flow (empty)	Double	325.23791
SYS22	Yearly vehicle 204 flow (loaded)	Double	0
SYS22T	Yearly vehicle 204 tonne flow	Double	0
SYS22E	Yearly vehicle 204 flow (empty)	Double	0
SYS25	Yearly vehicle 205 flow (loaded)	Double	0
SYS25T	Yearly vehicle 205 tonne flow	Double	0
SYS25E	Yearly vehicle 205 flow (empty)	Double	0
SYS30	Yearly vehicle 206 flow (loaded)	Double	0
SYS30T	Yearly vehicle 206 tonne flow	Double	0
SYS30E	Yearly vehicle 206 flow (empty)	Double	0
WG550	Yearly vehicle 207 flow (loaded)	Double	33.71220
WG550T	Yearly vehicle 207 tonne flow	Double	14486.2989
WG550E	Yearly vehicle 207 flow (empty)	Double	130.22619
WG750	Yearly vehicle 208 flow (loaded)	Double	0
WG750T	Yearly vehicle 208 tonne flow	Double	0
WG750E	Yearly vehicle 208 flow (empty)	Double	0.956900
WG950	Yearly vehicle 209 flow (loaded)	Double	0
WG950T	Yearly vehicle 209 tonne flow	Double	0
WG950E	Yearly vehicle 209 flow (empty)	Double	36.2745
FEEDW	Yearly vehicle 203 flow (loaded)	Double	383.6697
FEEDWT	Yearly vehicle 203 tonne flow	Double	166064.421875
FEEDWE	Yearly vehicle 203 flow (empty)	Double	325.23791

Field	Description	Data Format	Example
FEEDV	Yearly vehicle 202 flow (loaded)	Double	0
FEEDVT	Yearly vehicle 202 tonne flow	Double	0
FEEDVE	Yearly vehicle 202 flow (empty)	Double	0
TOT_RAIL	Yearly vehicle flow (loaded vehicles) for the rail mode	Double	417.3819391
TOT_RAIL_T	Yearly tonne flow for the rail mode	Double	180550.71875
TOT_RAIL_E	Yearly vehicle flow (empty vehicles) for the rail mode	Double	492.695495

Table 65 – Format of "Load_net_Rail_0_Link" table.

Description of Use: Assigned network for the rail mode

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Rail Assignment" subgroup under "Samgods Model\Assignment"

Used by: "Results" subgroup under "Samgods Model"

Made by user choice: No

Load_net_Sea_0 and Load_net_Sea_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Sea_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Sea_RCM

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114

Field	Description	Data Format	Example
ID_REGION	Region code	Double	114
MODE_N	Code for allowed modes	Double	3
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 66 – Format of "Load_net_Sea_0_Node" and "Load_net_Sea_RCM_Node" tables.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	41
B	End node	Integer	2407
Shape_Length	Length based on distance between coordinates in meters	Double	231.94827
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	9.3
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	0
CATEGORY	Link category	Double	221
FUNCTION	Index for travel time function	Double	72
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.23
ID_COUNTRY	Country code	Double	1
MODE_L	Mode	Double	3
CLOSED	0/1 flag to close links based on capacity saved in UL3 (applied to Kiel Canal)	Double	0
SWEDEN	Flag for classify a link in Sweden or outside	Double	1
CV5	Yearly vehicle flow (loaded vehicles) for vehicle 301	Double	1328.875
CV5T	Yearly tonne flows for vehicle 301	Double	1576008.0087

Field	Description	Data Format	Example
CV5E	Yearly flow (empty vehicles) for vehicle 301	Double	494.89151
CV16	Yearly vehicle flow (loaded vehicles) for vehicle 302	Double	0.48649999499321
CV16T	Yearly tonne flows for vehicle 302	Double	1897.0262
CV16E	Yearly flow (empty vehicles) for vehicle 302	Double	92.2993011474609
CV27	Yearly vehicle flow (loaded vehicles) for vehicle 303	Double	0.0064
CV27T	Yearly tonne flows for vehicle 303	Double	41.8562
CV27E	Yearly flow (empty vehicles) for vehicle 303	Double	59.1561012268066
CV100	Yearly vehicle flow (loaded vehicles) for vehicle 304	Double	0
CV100T	Yearly tonne flows for vehicle 304	Double	0
CV100E	Yearly flow (empty vehicles) for vehicle 304	Double	0.04459999874234
OV1	Yearly vehicle flow (loaded vehicles) for vehicle 305	Double	0.522
OV1T	Yearly tonne flows for vehicle 305	Double	104.5458
OV1E	Yearly flow (empty vehicles) for vehicle 305	Double	0.07059999555349
OV2	Yearly vehicle flow (loaded vehicles) for vehicle 306	Double	0
OV2T	Yearly tonne flows for vehicle 306	Double	0
OV2E	Yearly flow (empty vehicles) for vehicle 306	Double	0
OV3	Yearly vehicle flow (loaded vehicles) for vehicle 307	Double	53.4078979492188
OV3T	Yearly tonne flows for vehicle 307	Double	46724.79296875
OV3E	Yearly flow (empty vehicles) for vehicle 307	Double	849.075988769531
OV5	Yearly vehicle flow (loaded vehicles) for vehicle 308	Double	13.5809001922607
OV5T	Yearly tonne flows for vehicle 308	Double	16943.318359375
OV5E	Yearly flow (empty vehicles) for vehicle 308	Double	170.317993164062
OV10	Yearly vehicle flow (loaded vehicles) for vehicle 309	Double	48.0507011413574
OV10T	Yearly tonne flows for vehicle 309	Double	120034.1796875
OV10E	Yearly flow (empty vehicles) for vehicle 309	Double	98.8669967651367
OV20	Yearly vehicle flow (loaded vehicles) for vehicle 310	Double	8.51709938049316

Field	Description	Data Format	Example
OV20T	Yearly tonne flows for vehicle 310	Double	42584.1484375
OV20E	Yearly flow (empty vehicles) for vehicle 310	Double	34.8592987060547
OV40	Yearly vehicle flow (loaded vehicles) for vehicle 311	Double	2.90810012817383
OV40T	Yearly tonne flows for vehicle 311	Double	29069.599609375
OV40E	Yearly flow (empty vehicles) for vehicle 311	Double	0.25110000371933
OV80	Yearly vehicle flow (loaded vehicles) for vehicle 312	Double	
OV80T	Yearly tonne flows for vehicle 312	Double	17966.50390625
OV80E	Yearly flow (empty vehicles) for vehicle 312	Double	0.11080000549555
OV100	Yearly vehicle flow (loaded vehicles) for vehicle 313	Double	0
OV100T	Yearly tonne flows for vehicle 313	Double	0
OV100E	Yearly flow (empty vehicles) for vehicle 313	Double	0
OV250	Yearly vehicle flow (loaded vehicles) for vehicle 314	Double	0
OV250T	Yearly tonne flows for vehicle 314	Double	0
OV250E	Yearly flow (empty vehicles) for vehicle 314	Double	0
RO3	Yearly vehicle flow (loaded vehicles) for vehicle 315	Double	71.7285995483398
RO3T	Yearly tonne flows for vehicle 315	Double	57497.5703125
RO3E	Yearly flow (empty vehicles) for vehicle 315	Double	54.013599395752
RO6	Yearly vehicle flow (loaded vehicles) for vehicle 316	Double	0
RO6T	Yearly tonne flows for vehicle 316	Double	0
RO6E	Yearly flow (empty vehicles) for vehicle 316	Double	0
RO10	Yearly vehicle flow (loaded vehicles) for vehicle 317	Double	0
RO10T	Yearly tonne flows for vehicle 317	Double	0
RO10E	Yearly flow (empty vehicles) for vehicle 317	Double	0
ROF2	Yearly vehicle flow (loaded vehicles) for vehicle 318	Double	0
ROF2T	Yearly tonne flows for vehicle 318	Double	0
ROF2E	Yearly flow (empty vehicles) for vehicle 318	Double	0

Field	Description	Data Format	Example
ROF5	Yearly vehicle flow (loaded vehicles) for vehicle 319	Double	0
ROF5T	Yearly tonne flows for vehicle 319	Double	0
ROF5E	Yearly flow (empty vehicles) for vehicle 319	Double	0
ROF7	Yearly vehicle flow (loaded vehicles) for vehicle 320	Double	0
ROF7T	Yearly tonne flows for vehicle 320	Double	0
ROF7E	Yearly flow (empty vehicles) for vehicle 320	Double	0
RAF5	Yearly vehicle flow (loaded vehicles) for vehicle 321	Double	0
RAF5T	Yearly tonne flows for vehicle 321	Double	0
RAF5E	Yearly flow (empty vehicles) for vehicle 321	Double	0
TOT_SEA	Yearly vehicle flow (loaded vehicles) for mode sea	Double	1528.982421875
TOT_SEA_T	Yearly tonne flows for mode sea	Double	1908871.55018125
TOT_SEA_E	Yearly flow (empty vehicles) for mode sea	Double	1853.9580078125

Table 67 – Format of "Load_net_Sea_0_Link" and "Load_net_Sea_RCM_Link" table.

Description of Use: Assigned network for the sea mode

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Sea Assignment" subgroup under "Samgods Model\Assignment" and "Sea Assignment RCM" subgroup under "Samgods Model\RCM Assignment"

Used by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: No

Load_net_Air_0 and Load_net_Air_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Air_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Air_RCM

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data

Field	Description	Data Format	Example
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Code for allowed vehicles accessing the node	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 68 – Format of "Load_net_Air_0_Node" and "Load_net_Air_RCM_Node" tables.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	62
B	End node	Integer	2356
Shape_Length	Length based on distance between coordinates in meters	Double	476.759897627719
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	600
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	0
CATEGORY	Link category	Double	241
FUNCTION	Index for travel time function	Double	93
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.48
ID_COUNTRY	Country code	Double	1

Field	Description	Data Format	Example
MODE_L	Mode	Double	5
SWEDEN	Flag for classify a link in Sweden or outside	Double	1
FKYG	Yearly vehicle flow (loaded vehicles) for vehicle 401	Double	0
FKYGT	Yearly tonne flows for vehicle 401	Double	0
FKYGE	Yearly flow (empty vehicles) for vehicle 401	Double	0

Table 69 – Format of "Load_net_Air_0_Link" and "Load_net_Air_RCM_Link" tables.

Description of Use: Assigned network for the air mode

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Sea Assignment" subgroup under "Samgods Model\Assignment" and "Sea Assignment RCM" subgroup under "Samgods Model\RCM Assignment"

Used by: "Results" and "Results RCM" subgroup under "Samgods Model"

Made by user choice: No

Loaded_net_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Loaded_net_0

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114

Field	Description	Data Format	Example
MODE_N	Code for allowed vehicles on the link	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 70 – Format of "Loaded_net_0_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	40
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	36
B	End node	Integer	2791
Shape_Length	Length based on distance between coordinates in meters	Double	325.729949474258
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	50
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	50
CATEGORY	Link category	Double	201
FUNCTION	Index for travel time function	Double	31
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.33
DISTANCE	Distance in km	Double	0.33
ID_COUNTRY	Country code	Double	1
MODE_L	Mode	Double	1
CLOSED	Flag 0/1 for Sea assignment to close links where Vessel capacity > UL3	Double	0
CLOSED_	Flag 0/1 for rail assignment to close links where train type 202 or 203 is not allowed	Double	0
SWEDEN	Flag for classify a link in Sweden or outside	Double	1

Field	Description	Data Format	Example
ORIGINALCAP	Original capacity in bidirectional trains per day (from Capacity table)	Double	0
LGV3	Yearly vehicle flow (loaded vehicles) for vehicle 101	Double	0
LGV3T	Yearly tonne flows for vehicle 101	Double	0
LGV3E	Yearly flow (empty vehicles) for vehicle 101	Double	0
MGV16	Yearly vehicle flow (loaded vehicles) for vehicle 102	Double	0
MGV16T	Yearly tonne flows for vehicle 102	Double	0
MGV16E	Yearly flow (empty vehicles) for vehicle 102	Double	0
MGV24	Yearly vehicle flow (loaded vehicles) for vehicle 103	Double	0
MGV24T	Yearly tonne flows for vehicle 103	Double	0
MGV24E	Yearly flow (empty vehicles) for vehicle 103	Double	0
HGV40	Yearly vehicle flow (loaded vehicles) for vehicle 104	Double	0
HGV40T	Yearly tonne flows for vehicle 104	Double	0
HGV40E	Yearly flow (empty vehicles) for vehicle 104	Double	0
HGV60	Yearly vehicle flow (loaded vehicles) for vehicle 105	Double	0
HGV60T	Yearly tonne flows for vehicle 105	Double	0
HGV60E	Yearly flow (empty vehicles) for vehicle 105	Double	0
TOT_ROAD	Yearly vehicle flow (loaded vehicles) for mode road	Double	0
TOT_ROAD_T	Yearly tonne flows for mode road	Double	0
TOT_ROAD_E	Yearly flow (empty vehicles) for mode road	Double	0
KOMBI	Yearly vehicle flow (loaded vehicles) for vehicle 201	Double	0
KOMBIT	Yearly tonne flows for vehicle 201	Double	0
KOMBIE	Yearly flow (empty vehicles) for vehicle 201	Double	0
FS_TRAIN	Yearly vehicle flow (loaded vehicles) for vehicle 202	Double	0
FS_TRAINT	Yearly tonne flows for vehicle 202	Double	0
FS_TRAINE	Yearly flow (empty vehicles) for vehicle 202	Double	0

Field	Description	Data Format	Example
SYS22	Yearly vehicle flow (loaded vehicles) for vehicle 204	Double	0
SYS22T	Yearly tonne flows for vehicle 204	Double	0
SYS22E	Yearly flow (empty vehicles) for vehicle 204	Double	0
SYS25	Yearly vehicle flow (loaded vehicles) for vehicle 205	Double	0
SYS25T	Yearly tonne flows for vehicle 205	Double	0
SYS25E	Yearly flow (empty vehicles) for vehicle 205	Double	0
SYS30	Yearly vehicle flow (loaded vehicles) for vehicle 206	Double	0
SYS30T	Yearly tonne flows for vehicle 206	Double	0
SYS30E	Yearly flow (empty vehicles) for vehicle 206	Double	0
WG550	Yearly vehicle flow (loaded vehicles) for vehicle 207	Double	0
WG550T	Yearly tonne flows for vehicle 207	Double	0
WG550E	Yearly flow (empty vehicles) for vehicle 207	Double	0
WG750	Yearly vehicle flow (loaded vehicles) for vehicle 208	Double	0
WG750T	Yearly tonne flows for vehicle 208	Double	0
WG750E	Yearly flow (empty vehicles) for vehicle 208	Double	0
WG950	Yearly vehicle flow (loaded vehicles) for vehicle 209	Double	0
WG950T	Yearly tonne flows for vehicle 209	Double	0
WG950E	Yearly flow (empty vehicles) for vehicle 209	Double	0
FEEDW	Yearly vehicle flow (loaded vehicles) for vehicle 203	Double	0
FEEDWT	Yearly tonne flows for vehicle 203	Double	0
FEEDWE	Yearly flow (empty vehicles) for vehicle 203	Double	0
FEEDV	Yearly vehicle flow (loaded vehicles) for vehicle 202	Double	0
FEEDVT	Yearly tonne flows for vehicle 202	Double	0
FEEDVE	Yearly flow (empty vehicles) for vehicle 202	Double	0
TOT_RAIL	Yearly vehicle flow (loaded vehicles) for mode rail	Double	0
TOT_RAIL_T	Yearly tonne flows for mode rail	Double	0

Field	Description	Data Format	Example
TOT_RAIL_E	Yearly flow (empty vehicles) for mode rail	Double	0
CV5	Yearly vehicle flow (loaded vehicles) for vehicle 301	Double	0
CV5T	Yearly tonne flows for vehicle 301	Double	0
CV5E	Yearly flow (empty vehicles) for vehicle 301	Double	0
CV16	Yearly vehicle flow (loaded vehicles) for vehicle 302	Double	0
CV16T	Yearly tonne flows for vehicle 302	Double	0
CV16E	Yearly flow (empty vehicles) for vehicle 302	Double	0
CV27	Yearly vehicle flow (loaded vehicles) for vehicle 303	Double	0
CV27T	Yearly tonne flows for vehicle 303	Double	0
CV27E	Yearly flow (empty vehicles) for vehicle 303	Double	0
CV100	Yearly vehicle flow (loaded vehicles) for vehicle 304	Double	0
CV100T	Yearly tonne flows for vehicle 304	Double	0
CV100E	Yearly flow (empty vehicles) for vehicle 304	Double	0
OV1	Yearly vehicle flow (loaded vehicles) for vehicle 305	Double	0
OV1T	Yearly tonne flows for vehicle 305	Double	0
OV1E	Yearly flow (empty vehicles) for vehicle 305	Double	0
OV2	Yearly vehicle flow (loaded vehicles) for vehicle 306	Double	0
OV2T	Yearly tonne flows for vehicle 306	Double	0
OV2E	Yearly flow (empty vehicles) for vehicle 306	Double	0
OV3	Yearly vehicle flow (loaded vehicles) for vehicle 307	Double	0
OV3T	Yearly tonne flows for vehicle 307	Double	0
OV3E	Yearly flow (empty vehicles) for vehicle 307	Double	0
OV5	Yearly vehicle flow (loaded vehicles) for vehicle 308	Double	0
OV5T	Yearly tonne flows for vehicle 308	Double	0
OV5E	Yearly flow (empty vehicles) for vehicle 308	Double	0
OV10	Yearly vehicle flow (loaded vehicles) for vehicle 309	Double	0

Field	Description	Data Format	Example
OV10T	Yearly tonne flows for vehicle 309	Double	0
OV10E	Yearly flow (empty vehicles) for vehicle 309	Double	0
OV20	Yearly vehicle flow (loaded vehicles) for vehicle 310	Double	0
OV20T	Yearly tonne flows for vehicle 310	Double	0
OV20E	Yearly flow (empty vehicles) for vehicle 310	Double	0
OV40	Yearly vehicle flow (loaded vehicles) for vehicle 311	Double	0
OV40T	Yearly tonne flows for vehicle 311	Double	0
OV40E	Yearly flow (empty vehicles) for vehicle 311	Double	0
OV80	Yearly vehicle flow (loaded vehicles) for vehicle 312	Double	0
OV80T	Yearly tonne flows for vehicle 312	Double	0
OV80E	Yearly flow (empty vehicles) for vehicle 312	Double	0
OV100	Yearly vehicle flow (loaded vehicles) for vehicle 313	Double	0
OV100T	Yearly tonne flows for vehicle 313	Double	0
OV100E	Yearly flow (empty vehicles) for vehicle 313	Double	0
OV250	Yearly vehicle flow (loaded vehicles) for vehicle 314	Double	0
OV250T	Yearly tonne flows for vehicle 314	Double	0
OV250E	Yearly flow (empty vehicles) for vehicle 314	Double	0
RO3	Yearly vehicle flow (loaded vehicles) for vehicle 315	Double	0
RO3T	Yearly tonne flows for vehicle 315	Double	0
RO3E	Yearly flow (empty vehicles) for vehicle 315	Double	0
RO6	Yearly vehicle flow (loaded vehicles) for vehicle 316	Double	0
RO6T	Yearly tonne flows for vehicle 316	Double	0
RO6E	Yearly flow (empty vehicles) for vehicle 316	Double	0
RO10	Yearly vehicle flow (loaded vehicles) for vehicle 317	Double	0
RO10T	Yearly tonne flows for vehicle 317	Double	0
RO10E	Yearly flow (empty vehicles) for vehicle 317	Double	0

Field	Description	Data Format	Example
ROF2	Yearly vehicle flow (loaded vehicles) for vehicle 318	Double	0
ROF2T	Yearly tonne flows for vehicle 318	Double	0
ROF2E	Yearly flow (empty vehicles) for vehicle 318	Double	0
ROF5	Yearly vehicle flow (loaded vehicles) for vehicle 319	Double	0
ROF5T	Yearly tonne flows for vehicle 319	Double	0
ROF5E	Yearly flow (empty vehicles) for vehicle 319	Double	0
ROF7	Yearly vehicle flow (loaded vehicles) for vehicle 320	Double	0
ROF7T	Yearly tonne flows for vehicle 320	Double	0
ROF7E	Yearly flow (empty vehicles) for vehicle 320	Double	0
RAF5	Yearly vehicle flow (loaded vehicles) for vehicle 321	Double	0
RAF5T	Yearly tonne flows for vehicle 321	Double	0
RAF5E	Yearly flow (empty vehicles) for vehicle 321	Double	0
TOT_SEA	Yearly vehicle flow (loaded vehicles) for mode sea	Double	0
TOT_SEA_T	Yearly tonne flows for mode sea	Double	0
TOT_SEA_E	Yearly flow (empty vehicles) for mode sea	Double	0
FKYG	Yearly vehicle flow (loaded vehicles) for vehicle 401	Double	0
FKYGT	Yearly tonne flows for vehicle 401	Double	0
FKYGE	Yearly flow (empty vehicles) for vehicle 401	Double	0
VOLTO	Yearly vehicle flow (loaded vehicles) for all modes	Double	0
TONTO	Yearly tonne flows for all modes	Double	0
EMPTOE	Yearly flow (empty vehicles) for all modes	Double	0

Table 71 – Format of "Loaded_net_0_Link" table.

Description of Use: Assigned network for all modes. It is the merge of each mode network (Loaded_net_road_0, Loaded_net_rail_0, Loaded_net_sea_0, Loaded_net_air)_0

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: "Results" subgroup under "Samgods Model"

Used by: "Data Preparation LP" subgroup under "Samgods Model\Rail Capacity Management"

Made by user choice: no

Loaded_Bid_0 and Loaded_Bid_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Loaded_Bid_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Loaded_Bid_RCM

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Mode code	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 72 – Format of "Loaded_Bid_0_Node" and "Loaded_Bid_RCM_Node" tables.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	1
B	End node	Integer	2659

Shape_Length	Length based on distance between coordinates in meters	Double	230.867927616822
CATEGORY	Link category	Double	110
FUNCTION	Index for travel time function	Double	61
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.43
ID_COUNTRY	Country code	Double	1
MODE_L	Code for allowed vehicles on the link	Double	1
CLOSED	Flag 0/1 for vehicles 202/203	Double	0
TOT_ROAD_TM_BD	Bidirectional yearly flow for the mode road (in millions)	Double	1.35
TOT_RAIL_TM_BD	Bidirectional yearly flow for the mode rail (in millions)	Double	0
TOT_SEA_TM_BD	Bidirectional yearly flow for the mode sea (in millions)	Double	0

Table 73 – Format of "Loaded_Bid_0_Link" and "Loaded_Bid_RCM_Link" table.

Description of Use: Assigned network with bidirectional flows for road, rail and sea

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Results 1.0" subgroup under "Samgods Model\Results" and : "Results 1.0" subgroup under "Samgods Model\Results RCM"

Used by: Not use in the model, final report (see Scenario Outputs\Samgods Reports\ Reports\Bidirectional tons per mode and Scenario Outputs\RCM Reports\Bidirectional tons per mode).

Made by user choice: No

COMPARE_LOAD0_{Scenario_short_name}_{Scenario_for_comparison) and *COMPARE_LOADRCM*

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\
COMPARE_LOAD0_*{Scenario_short_name}*_{Scenario_for_comparison)

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\
COMPARE_LOADRCM_*{Scenario_short_name}*_{Scenario_for_comparison)

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Mode code	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 74 – Format of "COMPARE_LOAD0_{Scenario_short_name}_{Scenario_for_comparios)_Node" and " COMPARE_LOADRCM_{Scenario_short_name}_{Scenario_for_comparios)_Node" node tables.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	1
B	End node	Integer	2659
Shape_Length	Length based on distance between coordinates in meters	Double	230.867927616822
ID_COUNTRY	Country code	Double	1
MODE_L	Mode	Double	1
FUNCTION	Index for travel time function	Double	61
DIF_ROAD_T	Differences between bidirectional tonne flows /1000000 for the road mode	Double	0
DIF_RAIL_T	Differences between bidirectional tonne flows	Double	0

Field	Description	Data Format	Example
	/1000000 for the rail mode		
DIF_SEA_T	Differences between bidirectional tonne flows /1000000 for the sea mode	Double	0

Table 75 – Format of "COMPARE_LOAD0_{Scenario_short_name}_{Scenario_for_comparios)_Link" and " COMPARE_LOADRCM_{Scenario_short_name}_{Scenario_for_comparios)_Link" node tables.

Description of Use: calculate differences in flows between scenarios for Standard Logistics Module and Rail Capacity Management results.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Assignment" subgroup under "Compare scenarios"

Made by user choice: Yes - this output is present only if Compare scenarios has been run.

Load_net_Rail_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_Rail_RCM

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Mode code	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 76 – Format of "Load_net_Rail_RCM_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Start node	Integer	21
B	End node	Integer	2097
Shape_Length	Length based on distance between coordinates in meters	Double	206.155281301845
ID_LINK	Link identified for rail link. Two links in opposite direction share the same ID_LINK.	Long	25
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	50
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	0
CATEGORY	Link category	Double	211
FUNCTION	Index for travel time function	Double	66
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.2
ID_COUNTRY	Country code	Double	1
MODE_L	Mode	Double	2
SWEDEN	Flag for classify a link in Sweden or outside	Integer	1
CAP*	Original capacity in bidirectional trains per day on rail links. From Capacity table	Double	78
PREVCAP*	Capacity in previous Adjust Capacity Loop [bidirectional trains per day] - if not used PREVCAP=CAP	Double	78
FREMME2*	Emme node number for starting node	Double	1002
TOEMME2*	Emme node number for End node	Double	1003
CURRENTCAP*	Capacity in current Adjust Capacity Loop [bidirectional trains per day] - if not used CURRENTCAP=CAP. * values are present only for domestic rail links.	Double	78

Field	Description	Data Format	Example
KOMBI	Yearly vehicle flow (loaded vehicles) for vehicle 201	Double	0
KOMBIT	Yearly tonne flows for vehicle 201	Double	0
KOMBIE	Yearly flow (empty vehicles) for vehicle 201	Double	0
FS_TRAIN	Yearly vehicle flow (loaded vehicles) for vehicle 202	Double	325.237915039063
FS_TRAINT	Yearly tonne flows for vehicle 202	Double	166064.421875
FS_TRAINE	Yearly flow (empty vehicles) for vehicle 202	Double	325.237915039063
SYS22	Yearly vehicle flow (loaded vehicles) for vehicle 204	Double	0
SYS22T	Yearly tonne flows for vehicle 204	Double	0
SYS22E	Yearly flow (empty vehicles) for vehicle 204	Double	0
SYS25	Yearly vehicle flow (loaded vehicles) for vehicle 205	Double	0
SYS25T	Yearly tonne flows for vehicle 205	Double	0
SYS25E	Yearly flow (empty vehicles) for vehicle 205	Double	0
SYS30	Yearly vehicle flow (loaded vehicles) for vehicle 206	Double	0
SYS30T	Yearly tonne flows for vehicle 206	Double	0
SYS30E	Yearly flow (empty vehicles) for vehicle 206	Double	0
WG550	Yearly vehicle flow (loaded vehicles) for vehicle 207	Double	33.7122001647949
WG550T	Yearly tonne flows for vehicle 207	Double	14486.2989
WG550E	Yearly flow (empty vehicles) for vehicle 207	Double	130.226196289062
WG750	Yearly vehicle flow (loaded vehicles) for vehicle 208	Double	0
WG750T	Yearly tonne flows for vehicle 208	Double	0
WG750E	Yearly flow (empty vehicles) for vehicle 208	Double	0.95690000057221
WG950	Yearly vehicle flow (loaded vehicles) for vehicle 209	Double	0
WG950T	Yearly tonne flows for vehicle 209	Double	0
WG950E	Yearly flow (empty vehicles) for vehicle 209	Double	36.2745018005371
FEEDW	Yearly vehicle flow (loaded vehicles) for vehicle 203	Double	383.669799804687
FEEDWT	Yearly tonne flows for vehicle 203	Double	166064.421875

Field	Description	Data Format	Example
FEEDWE	Yearly flow (empty vehicles) for vehicle 203	Double	325.237915039063
FEEDV	Yearly vehicle flow (loaded vehicles) for vehicle 202	Double	0
FEEDVT	Yearly tonne flows for vehicle 202	Double	0
FEEDVE	Yearly flow (empty vehicles) for vehicle 202	Double	0
TOT_RAIL	Yearly vehicle flow (loaded vehicles) for mode rail	Double	417.381988525391
TOT_RAIL_T	Yearly tonne flows for mode rail	Double	180550.71875
TOT_RAIL_E	Yearly flow (empty vehicles) for mode rail	Double	492.695495605469

Table 77 – Format of "Load_net_Rail_RCM_Link" table.

Description of Use: Assigned network for the rail mode from Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Rail Assignment RCM" subgroup under "Samgods Model\RCM Assignment"

Used by: "Results RCM" subgroup under "Samgods Model"

Made by user choice: No

Load_net_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Load_net_RCM

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114

Field	Description	Data Format	Example
MODE_N	Mode code	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 78 – Format of "Load_net_RCM_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	40
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Starting node	Integer	36
B	End node	Integer	2791
Shape_Length	Length based on distance between coordinates in meters	Double	325.729949474258
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	50
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	50
CATEGORY	Link category	Double	201
FUNCTION	Index for travel time function	Double	31
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.33
ID_COUNTRY	Country code	Double	1
MODE_L	Mode	Double	1
CLOSED	Flag 0/1 for Sea assignment to close links where Vessel capacity > UL3	Double	0
SWEDEN	Flag for classify a link in Sweden or outside	Double	1
LGV3	Yearly vehicle flow (loaded vehicles) for vehicle 101	Double	0
LGV3T	Yearly tonne flows for vehicle 101	Double	0
LGV3E	Yearly flow (empty vehicles) for vehicle 101	Double	0

Field	Description	Data Format	Example
MGV16	Yearly vehicle flow (loaded vehicles) for vehicle 102	Double	0
MGV16T	Yearly tonne flows for vehicle 102	Double	0
MGV16E	Yearly flow (empty vehicles) for vehicle 102	Double	0
MGV24	Yearly vehicle flow (loaded vehicles) for vehicle 103	Double	0
MGV24T	Yearly tonne flows for vehicle 103	Double	0
MGV24E	Yearly flow (empty vehicles) for vehicle 103	Double	0
HGV40	Yearly vehicle flow (loaded vehicles) for vehicle 104	Double	0
HGV40T	Yearly tonne flows for vehicle 104	Double	0
HGV40E	Yearly flow (empty vehicles) for vehicle 104	Double	0
HGV60	Yearly vehicle flow (loaded vehicles) for vehicle 105	Double	0
HGV60T	Yearly tonne flows for vehicle 105	Double	0
HGV60E	Yearly flow (empty vehicles) for vehicle 105	Double	0
TOT_ROAD	Yearly vehicle flow (loaded vehicles) for mode road	Double	0
TOT_ROAD_T	Yearly tonne flows for mode road	Double	0
TOT_ROAD_E	Yearly flow (empty vehicles) for mode road	Double	0
KOMBI	Yearly vehicle flow (loaded vehicles) for vehicle 201	Double	0
KOMBIT	Yearly tonne flows for vehicle 201	Double	0
KOMBIE	Yearly flow (empty vehicles) for vehicle 201	Double	0
FS_TRAIN	Yearly vehicle flow (loaded vehicles) for vehicle 202	Double	0
FS_TRAINT	Yearly tonne flows for vehicle 202	Double	0
FS_TRAINE	Yearly flow (empty vehicles) for vehicle 202	Double	0
SYS22	Yearly vehicle flow (loaded vehicles) for vehicle 204	Double	0
SYS22T	Yearly tonne flows for vehicle 204	Double	0
SYS22E	Yearly flow (empty vehicles) for vehicle 204	Double	0
SYS25	Yearly vehicle flow (loaded vehicles) for vehicle 205	Double	0
SYS25T	Yearly tonne flows for vehicle 205	Double	0

Field	Description	Data Format	Example
SYS25E	Yearly flow (empty vehicles) for vehicle 205	Double	0
SYS30	Yearly vehicle flow (loaded vehicles) for vehicle 206	Double	0
SYS30T	Yearly tonne flows for vehicle 206	Double	0
SYS30E	Yearly flow (empty vehicles) for vehicle 206	Double	0
WG550	Yearly vehicle flow (loaded vehicles) for vehicle 207	Double	0
WG550T	Yearly tonne flows for vehicle 207	Double	0
WG550E	Yearly flow (empty vehicles) for vehicle 207	Double	0
WG750	Yearly vehicle flow (loaded vehicles) for vehicle 208	Double	0
WG750T	Yearly tonne flows for vehicle 208	Double	0
WG750E	Yearly flow (empty vehicles) for vehicle 208	Double	0
WG950	Yearly vehicle flow (loaded vehicles) for vehicle 209	Double	0
WG950T	Yearly tonne flows for vehicle 209	Double	0
WG950E	Yearly flow (empty vehicles) for vehicle 209	Double	0
FEEDW	Yearly vehicle flow (loaded vehicles) for vehicle 203	Double	0
FEEDWT	Yearly tonne flows for vehicle 203	Double	0
FEEDWE	Yearly flow (empty vehicles) for vehicle 203	Double	0
FEEDV	Yearly vehicle flow (loaded vehicles) for vehicle 202	Double	0
FEEDVT	Yearly tonne flows for vehicle 202	Double	0
FEEDVE	Yearly flow (empty vehicles) for vehicle 202	Double	0
TOT_RAIL	Yearly vehicle flow (loaded vehicles) for mode rail	Double	0
TOT_RAIL_T	Yearly tonne flows for mode rail	Double	0
TOT_RAIL_E	Yearly flow (empty vehicles) for mode rail	Double	0
CV5	Yearly vehicle flow (loaded vehicles) for vehicle 301	Double	0
CV5T	Yearly tonne flows for vehicle 301	Double	0
CV5E	Yearly flow (empty vehicles) for vehicle 301	Double	0
CV16	Yearly vehicle flow (loaded vehicles) for vehicle 302	Double	0

Field	Description	Data Format	Example
CV16T	Yearly tonne flows for vehicle 302	Double	0
CV16E	Yearly flow (empty vehicles) for vehicle 302	Double	0
CV27	Yearly vehicle flow (loaded vehicles) for vehicle 303	Double	0
CV27T	Yearly tonne flows for vehicle 303	Double	0
CV27E	Yearly flow (empty vehicles) for vehicle 303	Double	0
CV100	Yearly vehicle flow (loaded vehicles) for vehicle 304	Double	0
CV100T	Yearly tonne flows for vehicle 304	Double	0
CV100E	Yearly flow (empty vehicles) for vehicle 304	Double	0
OV1	Yearly vehicle flow (loaded vehicles) for vehicle 305	Double	0
OV1T	Yearly tonne flows for vehicle 305	Double	0
OV1E	Yearly flow (empty vehicles) for vehicle 305	Double	0
OV2	Yearly vehicle flow (loaded vehicles) for vehicle 306	Double	0
OV2T	Yearly tonne flows for vehicle 306	Double	0
OV2E	Yearly flow (empty vehicles) for vehicle 306	Double	0
OV3	Yearly vehicle flow (loaded vehicles) for vehicle 307	Double	0
OV3T	Yearly tonne flows for vehicle 307	Double	0
OV3E	Yearly flow (empty vehicles) for vehicle 307	Double	0
OV5	Yearly vehicle flow (loaded vehicles) for vehicle 308	Double	0
OV5T	Yearly tonne flows for vehicle 308	Double	0
OV5E	Yearly flow (empty vehicles) for vehicle 308	Double	0
OV10	Yearly vehicle flow (loaded vehicles) for vehicle 309	Double	0
OV10T	Yearly tonne flows for vehicle 309	Double	0
OV10E	Yearly flow (empty vehicles) for vehicle 309	Double	0
OV20	Yearly vehicle flow (loaded vehicles) for vehicle 310	Double	0
OV20T	Yearly tonne flows for vehicle 310	Double	0
OV20E	Yearly flow (empty vehicles) for vehicle 310	Double	0

Field	Description	Data Format	Example
OV40	Yearly vehicle flow (loaded vehicles) for vehicle 311	Double	0
OV40T	Yearly tonne flows for vehicle 311	Double	0
OV40E	Yearly flow (empty vehicles) for vehicle 311	Double	0
OV80	Yearly vehicle flow (loaded vehicles) for vehicle 312	Double	0
OV80T	Yearly tonne flows for vehicle 312	Double	0
OV80E	Yearly flow (empty vehicles) for vehicle 312	Double	0
OV100	Yearly vehicle flow (loaded vehicles) for vehicle 313	Double	0
OV100T	Yearly tonne flows for vehicle 313	Double	0
OV100E	Yearly flow (empty vehicles) for vehicle 313	Double	0
OV250	Yearly vehicle flow (loaded vehicles) for vehicle 314	Double	0
OV250T	Yearly tonne flows for vehicle 314	Double	0
OV250E	Yearly flow (empty vehicles) for vehicle 314	Double	0
RO3	Yearly vehicle flow (loaded vehicles) for vehicle 315	Double	0
RO3T	Yearly tonne flows for vehicle 315	Double	0
RO3E	Yearly flow (empty vehicles) for vehicle 315	Double	0
RO6	Yearly vehicle flow (loaded vehicles) for vehicle 316	Double	0
RO6T	Yearly tonne flows for vehicle 316	Double	0
RO6E	Yearly flow (empty vehicles) for vehicle 316	Double	0
RO10	Yearly vehicle flow (loaded vehicles) for vehicle 317	Double	0
RO10T	Yearly tonne flows for vehicle 317	Double	0
RO10E	Yearly flow (empty vehicles) for vehicle 317	Double	0
ROF2	Yearly vehicle flow (loaded vehicles) for vehicle 318	Double	0
ROF2T	Yearly tonne flows for vehicle 318	Double	0
ROF2E	Yearly flow (empty vehicles) for vehicle 318	Double	0
ROF5	Yearly vehicle flow (loaded vehicles) for vehicle 319	Double	0
ROF5T	Yearly tonne flows for vehicle 319	Double	0

Field	Description	Data Format	Example
ROF5E	Yearly flow (empty vehicles) for vehicle 319	Double	0
ROF7	Yearly vehicle flow (loaded vehicles) for vehicle 320	Double	0
ROF7T	Yearly tonne flows for vehicle 320	Double	0
ROF7E	Yearly flow (empty vehicles) for vehicle 320	Double	0
RAF5	Yearly vehicle flow (loaded vehicles) for vehicle 321	Double	0
RAF5T	Yearly tonne flows for vehicle 321	Double	0
RAF5E	Yearly flow (empty vehicles) for vehicle 321	Double	0
TOT_SEA	Yearly vehicle flow (loaded vehicles) for mode sea	Double	0
TOT_SEA_T	Yearly tonne flows for mode sea	Double	0
TOT_SEA_E	Yearly flow (empty vehicles) for mode sea	Double	0
FKYG	Yearly vehicle flow (loaded vehicles) for vehicle 401	Double	0
FKYGT	Yearly tonne flows for vehicle 401	Double	0
FKYGE	Yearly flow (empty vehicles) for vehicle 401	Double	0
VOLTO	Yearly vehicle flow (loaded vehicles) for all modes	Double	0
TONTO	Yearly tonne flows for all modes	Double	0
EMPTOE	Yearly flow (empty vehicles) for all modes	Double	0
ID_LINK	Link identified for rail link. Two links in opposite direction share the same ID_LINK.	Long	25
CAP*	Original capacity in bidirectional trains per day on rail links. From Capacity table	Double	78
PREVCAP*	Capacity in previous Adjust Capacity Loop [bidirectional trains per day] - if not used PREVCAP=CAP	Double	78
FREMME2*	Emme node number for starting node	Double	1002
TOEMME2*	Emme node number for End node	Double	1003
CURRENTCAP	Capacity in current Adjust Capacity Loop [bidirectional trains per day] - if not used CURRENTCAP=CAP.	Double	78

Field	Description	Data Format	Example
	* values are present only for domestic rail links.		

Table 79 – Format of "Load_net_RCM_Link" table.

Description of Use: Assigned network for the all modes from Rail Capacity Management. It is the merge of all mode networks (Load_net_road_RCM, Load_net_rail_RCM, Load_net_sea_RCM, Load_net_air_RCM)

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Results RCM" subgroup under "Samgods Model"

Used by: "Results RCM" subgroup under "Samgods Model" to produce related reports (Report_1, Report_4, etc).

Made by user choice: No

Rail_BiDir

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Rail_BiDir

Node Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Mode code	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Double	Upplands-Väsby

Table 80 – Format of "Rail_BiDir_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Starting node	Integer	21
B	End node	Integer	2097
Shape_Length	Length based on distance between coordinates in meters	Double	206.155281301845
ID_LINK	ID_LINK from capacity table	Integer	1
CAP*	Capacity in bidirectional trains per day (from Capacity table)	Text	418
TOT_LOAD	Number of loaded bidirectional trains per day	Double	71.24434
TOT_EMPT	Number of unloaded bidirectional trains per day	Double	31.21692
TOT_ASS	Number of total bidirectional trains per day (loaded+unloaded)	Double	102.4613
DIFF_VC	Difference between TOT_ASS and CAP	Double	-315.539
FLAG_OVCAP	Flag 0/1. 1 means the link is overcapacity by at least 1 train	Integer	0
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	50
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	0
CATEGORY	Link category	Double	211
FUNCTION	Index for travel time function	Double	66
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	0.2
ID_COUNTRY	Country code	Double	1
MODE_L	Mode	Double	2

Field	Description	Data Format	Example
SWEDEN	Flag for classify a link in Sweden or outside	Integer	1
PREVCAP*	Capacity in previous Adjust Capacity Loop [bidirectional trains per day] - if not used PREVCAP=CAP	Double	78
FREMME2	Starting Emme node	Integer	1002
TOEMME2	Ending Emme node	Integer	1003
CURRENTCAP*	Capacity in current Adjust Capacity Loop [bidirectional trains per day] - if not used CURRENTCAP=CAP. * values are present only for domestic rail links.	Double	78
KOMBI	Yearly vehicle flow (loaded vehicles) for vehicle 201	Double	0
KOMBIT	Yearly tonne flows for vehicle 201	Double	0
KOMBIE	Yearly flow (empty vehicles) for vehicle 201	Double	0
FS_TRAIN	Yearly vehicle flow (loaded vehicles) for vehicle 202	Double	325.237915039063
FS_TRAINT	Yearly tonne flows for vehicle 202	Double	166064.421875
FS_TRAINE	Yearly flow (empty vehicles) for vehicle 202	Double	325.237915039063
SYS22	Yearly vehicle flow (loaded vehicles) for vehicle 204	Double	0
SYS22T	Yearly tonne flows for vehicle 204	Double	0
SYS22E	Yearly flow (empty vehicles) for vehicle 204	Double	0
SYS25	Yearly vehicle flow (loaded vehicles) for vehicle 205	Double	0
SYS25T	Yearly tonne flows for vehicle 205	Double	0
SYS25E	Yearly flow (empty vehicles) for vehicle 205	Double	0
SYS30	Yearly vehicle flow (loaded vehicles) for vehicle 206	Double	0
SYS30T	Yearly tonne flows for vehicle 206	Double	0
SYS30E	Yearly flow (empty vehicles) for vehicle 206	Double	0
WG550	Yearly vehicle flow (loaded vehicles) for vehicle 207	Double	33.7122001647949
WG550T	Yearly tonne flows for vehicle 207	Double	14486.2989

Field	Description	Data Format	Example
WG550E	Yearly flow (empty vehicles) for vehicle 207	Double	130.226196289062
WG750	Yearly vehicle flow (loaded vehicles) for vehicle 208	Double	0
WG750T	Yearly tonne flows for vehicle 208	Double	0
WG750E	Yearly flow (empty vehicle) for vehicle class 208	Double	0.95690000057221
WG950	Yearly flow (loaded vehicle) for vehicle class 209	Double	0
WG950T	Yearly tonnes for vehicle class 209	Double	0
WG950E	Yearly flow (empty vehicle) for vehicle class 209	Double	36.2745018005371
FEEDW	Yearly flow (loaded vehicle) for vehicle class 203	Double	383.669799804687
FEEDWT	Yearly tonnes for vehicle class 203	Double	166064.421875
FEEDWE	Yearly flow (empty vehicle) for vehicle class 203	Double	325.237915039063
FEEDV	Yearly flow (loaded vehicle) for vehicle class 202	Double	0
FEEDVT	Yearly tonnes for vehicle class 202	Double	0
FEEDVE	Yearly flow (empty vehicle) for vehicle class 202	Double	0
TOT_RAIL	Yearly flow (loaded vehicle) for mode rail	Double	417.381988525391
TOT_RAIL_T	Yearly tonnes for mode rail	Double	180550.71875
TOT_RAIL_E	for mode rail	Double	492.695495605469

Table 81 – Format of "Rail_BiDir_Link" table.

Description of Use: Assigned network for the rail mode from Rail Capacity Management with daily and yearly flows.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Reports 1.0" subgroup under "Samgods Model\Results RCM"

Made by user choice: No.

Compare_Bid

Scenario_tree\Base2006\Output0_Base2006.mdb\Compare_Bid

Node Table Format:

Field	Description	Data Format	Example

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (point)	OLE Object	Long Binary Data
N	Node number	Double	1
X	Coordinate X	Double	1620000
Y	Coordinate Y	Double	6601000
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	Scbstann code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Mode code	Double	MODE_N
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in term of location	Double	Upplands-Väsby

Table 82 – Format of "Compare_Bid_Node" table.

Link Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SHAPE	Feature class type (lane)	OLE Object	Long Binary Data
A	Starting node	Integer	1
B	Ending node	Integer	2659
Shape_Length	Length based on distance between coordinates in meters	Double	230.867927616822
CATEGORY	Link category	Double	110
FUNCTION	Delay function code	Double	61
NLANES	Number of lanes (could be with digits)	Double	1
UL2	Distance in km	Double	0.43
ID_COUNTRY	Country code	Double	1
MODE_L	Mode	Double	1

Field	Description	Data Format	Example
CLOSED	Flag 0/1 for vehicles 202/203	Double	0
RAIL_STD	Bidirectional yearly flow for mode Rail (in millions) from Standard Logistics Module	Double	0
SEA_STD	Bidirectional yearly flow for mode Sea (in millions) from Standard Logistics Module	Double	0
ROAD_STD	Bidirectional yearly flow for mode road (in millions) from Standard Logistics Module	Double	1.35
GEOMETRYSOURCE	Flag 1/2 1 from RCM network, 2 from Standard Logistics Module	Double	1
RAIL_RCM	Bidirectional yearly flow for mode Rail (in millions) from Rail Capacity Management Procedure	Double	0
SEA_RCM	Bidirectional yearly flow for mode Sea (in millions) from Rail Capacity Management Procedure	Double	0
ROAD_RCM	Bidirectional yearly flow for mode road (in millions) from Rail Capacity Management Procedure	Double	1.35
DIF_A_ROAD	ROAD_RCM-ROAD_STD	Double	0
DIF_A_RAIL	RAIL_RCM-RAIL_STD	Double	0
DIF_A_SEA	SEA_RCM-SEA_STD	Double	0
DIF_P_ROAD	DIF_A_ROAD/ROAD_STD	Double	0
DIF_P_RAIL	DIF_A_RAIL/RAIL_STD	Double	0
DIF_P_SEA	DIF_A_SEA/SEA_STD	Double	0

Table 83 – Format of "Compare_Bid_Link" table.

Description of Use: Differences of bidirectional flows (in millions of tons) for Road, Sea and Rail mode on two results, one from standard logistics module and the other from rail capacity management procedure.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcMap

Created by: "Results 1.0" subgroup under "Samgods Model\Results" and : "Results 1.0" subgroup under "Samgods Model\Results RCM"

Used by: Not use in the model, final report (see Scenario Outputs\RCM Reports\ Comparison Ml Tons RCM vs. Standard per mode).

Made by user choice: No

Table format in geodatabase

CHAIN_OD_COV_Base2006{SCENARIO_SHORTNAME}_0 and CHAIN_OD_COV_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\CHAIN_OD_COV_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\CHAIN_OD_COV_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
CHAIN_TY	Chain Type	Text	A
NSHIP_D	Number of Shipments Domestic	Double	9187.49
NSHIP_I	Number of Shipments International	Double	239.81
NSHIP_T	Number of Shipments Total Domestic	Double	9187.49
KM_DOM	Km Domestic	Double	48992.5
KM_INT	Km International	Double	2498.34
KM_TOT	Km Total Domestic	Double	52656.48
TON_DOM	Tonnes Domestic	Double	2492.91
TON_INT	Tonnes International	Double	184.62
TON_TOT	Tonnes Total Domestic	Double	2492.91
TKM_DOM	TonneKm Domestic	Double	429749.55
TKM_INT	TonneKm International	Double	37927.9
TKM_TOT	TonneKm Total Domestic	Double	483111.92
COST_DOM	Costs (SEK) Domestic	Double	8039900.44
COST_INT	Costs (SEK) International	Double	383296.88
COST_TOT	Costs (SEK) Total Domestic	Double	8039900.44
AC_TKM_DOM	Average Logistic Cost Domestic	Double	18.71
AC_TKM_INT	Average Logistic Cost International	Double	10.11
AC_TKM_TOT	Average Logistic Cost Total Domestic	Double	16.64

Table 84 – Format of "CHAIN_OD_COV_{SCENARIO_SHORTNAME}_0" and "CHAIN_OD_COV_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: Translation of ChainChoi0_{SCENARIO_SHORTNAME}.rep and ChainChoi0FIN_{SCENARIO_SHORTNAME}.rep in geodatabase tables for the part related to chain types. The first is the merge of all the reports ChainChoiXX.rep from Standard Logistics Module and the second is the merge of all the reports ChainChoiXXFIN.rep from Rail Capacity Management Module. These files reside under LogMod_Y\ChainChoi\Output folder and are merged in ChainChoi0_{SCENARIO_SHORTNAME}.rep and ChainChoi0DIN_{SCENARIO_SHORTNAME}.rep via MERGE.exe program.

The two geodatabase tables are the source for " Report_2_LM_CHAINS" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube Database window (see Help>Cube Base>Database window)

Created by: “Save Reports” subgroup under “Samgods Model\Logistics Module”

Made by user choice: No.

COM_L_D_{SCENARIO_SHORTNAME}_0 and COM_L_D_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\COM_L_D_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\COM_L_D_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
COMM	Commodity group	Double	1
TRANVOL	Actual transported tons (tonnes)	Double	5212689
DEMANDVOL	PWC values (tonnes)	Double	5212689
SUCCESRATE	Success Rate (100% means all the transport chains got a solution)	Double	100

Table 85 – Format of “COM_L_D_{SCENARIO_SHORTNAME}_0” and “COM_L_D_{SCENARIO_SHORTNAME}_RCM” tables.

Description of Use: Translation of ChainChoi0_{SCENARIO_SHORTNAME}.rep and ChainChoi0DIN_{SCENARIO_SHORTNAME}.rep in geodatabase tables for the part related to success rate. The first is the merge of all the reports ChainChoiXX.rep from Standard Logistics Module and the second is the merge of all the reports ChainChoiXXFIN.rep from Rail Capacity Management Module. These files reside under LogMod_Y\ChainChoi\Output folder and are merged in ChainChoi0_{SCENARIO_SHORTNAME}.rep and ChainChoi0DIN_{SCENARIO_SHORTNAME}.rep via MERGE.exe program.

The two geodatabase tables are the source for " Report_2_LM_DEMAND" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube Database window (see Help>Cube Base>Database window)

Created by: “Save Reports” subgroup under “Samgods Model\Logistics Module” and “Save Reports” subgroup under “Samgods Model\Rail Capacity Management\Run Final Process”

Made by user choice: No.

VHCL_OD_COV_{SCENARIO_SHORTNAME}_0 and *VHCL_OD_COV_{SCENARIO_SHORTNAME}_RCM*

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\VHCL_OD_COV_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\VHCL_OD_COV_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
VEH_TY	Vehicle Type	Double	101
DESCR	Description of vehicle type	Text	Lorry light LGV.< 3.5 ton
NSHIP_D	Number of Shipments Domestic	Double	59.56
NSHIP_I	Number of Shipments International	Double	59.65
NSHIP_T	Number of Shipments Total Domestic	Double	0.75
NV_DOM	Number of Vehicles Domestic	Double	58.91
NV_INT	Number of Vehicles International	Double	58.99
NV_TOT	Number of Vehicles Total Domestic	Double	0.09
KM_DOM	Km Domestic	Double	4608.5
KM_INT	Km International	Double	4612.13
KM_TOT	Km Total Domestic	Double	6.72
TON_DOM	Tonnes Domestic	Double	92.61
TON_INT	Tonnes International	Double	92.73
TON_TOT	Tonnes Total Domestic	Double	0.12
TKM_DOM	TonneKm Domestic	Double	7529.42

Field	Description	Data Format	Example
TKM_INT	TonneKm International	Double	7535.17
TKM_TOT	TonneKm Total Domestic	Double	10.27
ALF_DOM	Average Load Factor Domestic	Double	0.79
ALF_INT	Average Load Factor International	Double	0.79
ALF_TOT	Average Load Factor Total Domestic	Double	0.68
AVDIST_DOM	Average Distance Domestic	Double	78.2
AVDIST_INT	Average Distance International	Double	78.2
AVDIST_TOT	Average Distance Total Domestic	Double	73.9

Table 86 – Format of "VHCL_OD_COV_{SCENARIO_SHORTNAME}_0" and "VHCL_OD_COV_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: Translation of ChainChoi0_{SCENARIO_SHORTNAME}.rep and ChainChoi0DIN_{SCENARIO_SHORTNAME}.rep in geodatabase tables for the part related to vehicle types. The first is the merge of all the reports ChainChoiXX.rep from Standard Logistics Module and the second is the merge of all the reports ChainChoiXXFIN.rep from Rail Capacity Management Module. These files reside under LogMod_Y\ChainChoi\Output folder and are merged in ChainChoi0_{SCENARIO_SHORTNAME}.rep and ChainChoi0DIN_{SCENARIO_SHORTNAME}.rep via MERGE.exe program.

The two geodatabase tables are the source for " Report_2_Logistics Module" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube Database window (see Help>Cube Base>Database window)

Created by: "Save Reports" subgroup under "Samgods Model\Logistics Module" and "Save Reports" subgroup under "Samgods Model\Rail Capacity Management\Run Final Process"

Made by user choice: No.

Report_1_{SCENARIO_SHORTNAME}_0 and Report_1_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_1_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_1_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
VEH_NR	Vehicle type	Text	101
VEH_CLASS	Description of vehicle	Text	LGV3
NV_A_TOT	Total vehicles	Double	3322.09
NV_A_DTOT	Total vehicles (Dtotal)	Double	3321.29
VKM_A_TOT	Total vehicle kilometres	Double	9277.03
VKM_A_DTOT	Total vehicle kilometres (Dtotal)	Double	9258.29
NV_L_TOT	Loaded vehicles	Double	1658.93
NV_L_DTOT	Loaded vehicles (Dtotal)	Double	1658.54
VKM_L_TOT	Loaded vehicle kilometres	Double	4600.46
VKM_L_DTOT	Loaded vehicle kilometres (Dtotal)	Double	4591.29
NV_E_TOT	Unloaded vehicles	Double	1663.16
NV_E_DTOT	Unloaded vehicles (Dtotal)	Double	1662.75
VKM_E_TOT	Unloaded vehicle kilometres	Double	4676.57
VKM_E_DTOT	Unloaded vehicles (Dtotal)	Double	4666.99

Table 87 – Format of "Report_1_{SCENARIO_SHORTNAME}_0" and "Report_1_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for "Report_1 Tot VHC and VHCKM by VHC Type" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: no

Report_3_TonKM_perMode_0 and Report_3_TonKM_perMode_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_3_TonKM_perMode_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_3_TonKM_perMode_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
SCENARIO	STANDARD/RCM (method that produces the results)	Text	STANDARD
ROAD	Total domestic tonnekm from model (in million) for road mode	Double	42.04
RAIL	Total domestic tonnekm from model (in million) for rail mode	Double	33.62
SEA	Total domestic tonnekm from model (in million) for sea mode	Double	38.79
AIR	Total international tonnekm from model (in million) for air mode	Double	2.34
ROADS	Total domestic tonnekm from statistics (in million) for road mode - value hardcoded in script 02_APPLICATIONS\2_RUN\RESULTS\NEMAT01A.S	Double	39.9
RAILS	Total domestic tonnekm from model (in million) for rail mode - value hardcoded in script 02_APPLICATIONS\2_RUN\RESULTS\NEMAT01A.S	Double	22.3
SEAS	Total domestic tonnekm from model (in million) for sea mode- value hardcoded in script 02_APPLICATIONS\2_RUN\RESULTS\NEMAT01A.S	Double	36.9
ROADD	Absolute Differences between modelled and statistics for road mode	Double	2.14
RAILD	Absolute Differences between modelled and statistics for rail mode	Double	11.32
SEAD	Absolute Differences between modelled and statistics for sea mode	Double	1.89
RMSE	Root-mean-square error	Double	8.26

Table 88 – Format of "Report_3_TonKM_perMode_0" and "Report_3_TonKM_perMode_RCM" tables.

Description of Use: The two geodatabase tables are the source for " Report_3_TonKm_per_Mode_with_2006Statistics" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: no

Report_4_{SCENARIO_SHORTNAME}_0 and Report_4_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_4_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_4_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
VEH_NR	Vehicle class	Text	101
VEH_CLASS	Description of vehicle class	Text	LGV3
TON_TOT	Total tonnes	Double	2712.78
TON_DTOT	Total domestic tonnes	Double	2712.2
TON_INT	International tonnes	Double	0.58
TKM_TOT	Total tonnekm	Double	7520.17
TKM_DTOT	Total domestic tonnekm	Double	7505.61
TKM_INT	International tonnekm	Double	14.56

Table 89 – Format of "Report_4_{SCENARIO_SHORTNAME}_0" and "Report_4_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for " Report_4_Total tonnes and tonnekm by VHC Type" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: no

Report_5_{SCENARIO_SHORTNAME}_0 and Report_5_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_5_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_5_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ZONE_N	Zone number	Double	1
ZONE_ORIG	Original zone number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
CENTRALL	Node description in terms of location	Text	Upplands-Väsby
P01_IMPORT	Total logistic cost in import for Commodity 1	Double	832.67
P01_EXPORT	Total logistic cost in export for Commodity 1	Double	859.21
…	…	…	…
P35_IMPORT	Total logistic cost in import for Commodity 35	Double	161690.81
P35_EXPORT	Total logistic cost in export for Commodity 1	Double	67590.22

Table 90 – Format of "Report_5_{SCENARIO_SHORTNAME}_0" and "Report_5_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for "Report_5_Total logistic cost at zone-level " under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcCatalog

Created by: "Save Reports" subgroup under "Samgods Model\Logistics Module" and "Save Reports" subgroup under "Samgods Model\Rail Capacity Management\Run Final Process"

Made by user choice: No.

Report_6_{SCENARIO_SHORTNAME}_0 and Report_6_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_6_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_6_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	93
ZONE_N	Zone number	Double	173
ZONE_ORIG	Original zone number based on the combination of ID_Country, ID_region and MODE_N values	Double	786011
PORTAREANR	Port area number	Double	2
CENTRALL	Node description in terms of location	Text	Hultsfred
P01_DAIMPORN	Imported tons for commodity 1	Double	0.06
P01_DAEXPORN	Exported tons for commodity 1	Double	0.07
P01_REGULARN	Regular tons for commodity 1	Double	0.54
…	…	…	…
P35_DAIMPORN	Imported tons for commodity 35	Double	0
P35_DAEXPORN	Exported tons for commodity 35	Double	0
P35_REGULARN	Regular tons for commodity 35	Double	0

Table 91 – Format of "Report_6_{SCENARIO_SHORTNAME}_0" and "Report_6_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for " Report_6_Goods flow through terminals (number of tonnes in and out per year)" under data panel Reports\Logistics Module and Reports\Rail Capacity Management. They are also input for calculation of " Report_13_Tons_per_PortArea_and_STAN_Group".

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: "Save Reports" subgroup under "Samgods Model\Logistics Module" and "Save Reports" subgroup under "Samgods Model\Rail Capacity Management\Run Final Process"

Made by user choice: No.

Report_7_{SCENARIO_SHORTNAME}_0 and Report_7_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_7_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_7_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
COMMODITY	Commodity type	Text	1
ROAD	Domestic tonnekm container on road	Double	1003.78
RAIL	Domestic tonnekm container on rail	Double	2440.79
SEA	Domestic tonnekm container on sea	Double	0.84
AIR	Domestic tonnekm container on air	Double	0
V101	Domestic tonnekm container Vehicle class 101	Double	1.2
....	Domestic tonnekm container	Double	...
V401	Domestic tonnekm container Vehicle class 401	Double	0

Table 92 – Format of "Report_7_{SCENARIO_SHORTNAME}_0" and "Report_7_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for " Report_7_Domestic tonnekm with container per mode (road, rail, sea, air) and vehicle cl" under data panel Reports\Logistics Module and Reports\Rail Capacity Management. They are also input for calculation of " Report_13_Tons_per_PortArea_and_STAN_Group".

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcCatalog

Created by: “Save Reports” subgroup under “Samgods Model\Logistics Module” and “Save Reports” subgroup under “Samgods Model\Rail Capacity Management\Run Final Process”

Made by user choice: No.

Report_8_{SCENARIO_SHORTNAME}_0 and Report_8_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_8_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_8_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
COMMODITY	Commodity type	Text	1
ROAD	Domestic vehiclekm container on road	Double	98.28
RAIL	Domestic vehiclekm container on rail	Double	4.82
SEA	Domestic vehiclekm container on sea	Double	0.09
AIR	Domestic vehiclekm container on air	Double	0
V101	Domestic vehiclekm container Vehicle class 101	Double	1.2
....	Domestic vehiclekm container	Double	...
V401	Domestic vehiclekm container Vehicle class 401	Double	0

Table 93 – Format of “Report_8_{SCENARIO_SHORTNAME}_0” and “Report_8_{SCENARIO_SHORTNAME}_RCM” tables.

Description of Use: The two geodatabase tables are the source for " Report_8_Domestic vehicle kms with container per mode (road, rail, sea, air) and vehicle cl" under data panel Reports\Logistics Module and Reports\Rail Capacity Management. They are also input for calculation of " Report_13_Tons_per_PortArea_and_STAN_Group".

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcCatalog

Created by: "Save Reports" subgroup under "Samgods Model\Logistics Module" and "Save Reports" subgroup under "Samgods Model\Rail Capacity Management\Run Final Process"

Made by user choice: No.

Report_9_{SCENARIO_SHORTNAME}_0 and Report_9_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_9_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_9_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
REGION	Region code	Double	114
ID_COUNTRY	Country code	Double	1
DESCRIP	Description of region	Text	Upplands Väsby
VKM_L_ROAD	Road loaded vehiclekm	Double	2134.9469
TKM_ROAD	Road tonnekm	Double	46437.5225
VKM_E_ROAD	Road empty vehiclekm	Double	1744.6717
VKM_L_RAIL	Rail loaded vehiclekm	Double	38.3121
TKM_RAIL	Rail tonnekm	Double	20523.7748
VKM_E_RAIKL	Rail empty vehiclekm	Double	11.7594
VKM_A_ROAD	Road total vehiclekm	Double	3879.6186
VKM_A_RAIL	Rail total vehiclekm	Double	50.0715

Table 94 – Format of "Report_9_{SCENARIO_SHORTNAME}_0" and "Report_9_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for " Report_9_Vehicle kms and Tonne kms per geographic region" under data panel Samgods Reports\Reports and RCM Reports\Reports.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcCatalog

Created by: “Results” and “Results RCM” subgroups under “Samgods Model”

Made by user choice: No.

Report_10_{SCENARIO_SHORTNAME}_0 and Report_10_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_10_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_10_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
COMMODITY	Commodity type	Text	1
TYPE	Domestic, International or Total Domestic	Text	Domestic
ROAD	Tonka on road	Double	397113.25
RAIL	Tonnekm on rail	Double	39074.77
SEA	Tonnekm on sea	Double	32762.63
AIR	Tonnekm on air	Double	0
V101	Tonnekm Vehicle class 101	Double	1.2
....	Tonnekm....	Double	...
V401	Tonnekm Vehicle class 401	Double	0

Table 95 – Format of “Report_10_{SCENARIO_SHORTNAME}_0” table.

Description of Use: The two geodatabase tables are the source for " Report_10_Transport work (tonne kms) per mode and vehicle cl, total and split per commodity, domestic, tdomestic and international" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: “Save Reports” subgroup under “Samgods Model\Logistics Module” and “Save Reports” subgroup under “Samgods Model\Rail Capacity Management\Run Final Process”

Made by user choice: No.

Report_11_{SCENARIO_SHORTNAME}_0 and Report_11_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_11_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_11_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
COMMODITY	Commodity type	Text	1
TYPE	Domestic, International or Total Domestic	Text	Domestic
ROAD	Tons on road	Double	4805.92
RAIL	Tons on rail	Double	421.46
SEA	Tons on sea	Double	158.2
AIR	Tons on air	Double	0
V101	Tons Vehicle class 101	Double	1.2
....	Tons....	Double	...
V401	Tons Vehicle class 401	Double	0

Table 96 – Format of “Report_11_{SCENARIO_SHORTNAME}_0” and “Report_11_{SCENARIO_SHORTNAME}_RCM” tables.

Description of Use: The two geodatabase tables are the source for " Report_11_ Transported goods volume per mode and vehicle cl, total and split per commodity, domestic, tdomestic and international" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: “Save Reports” subgroup under “Samgods Model\Logistics Module” and “Save Reports” subgroup under “Samgods Model\Rail Capacity Management\Run Final Process”

Made by user choice: No.

Report_12_{SCENARIO_SHORTNAME}_0 and Report_12_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_12_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_12_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
COMMODITY	Commodity type	Text	1
VH_CL	Vehicle class	Text	101
NCOSTDTOT	Node cost total domestic	Double	269.36
NCOSTINT	Node cost international	Double	0
LCOSTDTOT	Link cost total domestic	Double	345.17
LCOSTINT	Link cost international	Double	0

Table 97 – Format of “Report_12_{SCENARIO_SHORTNAME}_0” and “Report_12_{SCENARIO_SHORTNAME}_RCM” tables.

Description of Use: The two geodatabase tables are the source for " Report_12_node and link costs per vehicle and product group" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: “Save Reports” subgroup under “Samgods Model\Logistics Module” and “Save Reports” subgroup under “Samgods Model\Rail Capacity Management\Run Final Process”

Made by user choice: No.

Report_13_Portarea_0_STD and Report_13_Portarea_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_13_Portarea_0_STD

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_13_Portarea_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
PORTAREA	Port area number	Double	1
DES	3 tables are saved, each of them has a different DES (Model results, Statistics, Differences)	Text	Model results
PDES	Port area description	Text	Haparanda-Skellefteå
STAN1	Tons/1000 per port area and STAN group 1	Double	0
…	…	…	…
STAN12	Tons/1000 per port area and STAN group 12	Double	21.52
TOT	Total Tons/1000 per port area	Double	9170.14

Table 98 – Format of "Report_13_Portarea_0_STD" and "Report_13_Portarea_RCM" tables.

Description of Use: The two geodatabase tables are the source for "Report_13_Tons_per_PortArea_and_STAN_Group" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: Created by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: No.

Report_14_Oresund_Kiel_0_STD and Report_14_Oresund_Kiel_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_14_Oresund_Kiel_0_STD

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_14_Oresund_Kiel_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
A	Starting node	Integer	2566
B	End node	Integer	24604
ID	Identification number in the system	Integer	1
DES	Description of type of statistics and infrastructure (Oresund road, Oresund rail, Kiel Canal, Jylland)	Text	OresundRoad
STATISTIC	Unit of statistics (vehicles or tons)	Text	Vehicles
ID_S	ID of statistics (Vehicles=1 Tons=2)	Integer	1

Field	Description	Data Format	Example
VALUE_	Value of statistics (tons or vehicles depending on ID_S)	Integer	251589
TOTVH	Total vehicles from model	Double	495658
TOTTON	Total tons from model	Double	5418635.63197
DIF	Absolute differences between MODEL and VALUE_	Double	244069
DIF_PER	Relative differences between MODEL and VALUE_	Double	0.97
MODEL	Total vehicles or tons from model depending on ID_S value	Double	495658

Table 99 – Format of "Report_14_Oresund_Kiel_0_STD" and "Report_14_Oresund_Kiel_RCM" tables.

Description of Use: The two geodatabase tables are the source for " Report_14_Oresund Bridge_Kiel Canal_Jylland" under data panel Reports\Logistics Module and Reports\Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: No.

Report_15_RailLinksRCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_15_RailLinksRCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_LINK	ID_LINK from capacity table	Integer	1
A	Starting Voyager node	Integer	2000
B	Ending Voyager node	Integer	2001
EMME_A	Starting Emme node	Integer	1002

Field	Description	Data Format	Example
EMME_B	Ending Emme node	Integer	1003
CAP	Capacity in bidirectional trains per day (from Capacity table)	Text	418
TOT_LOAD	Number of loaded bidirectional trains per day	Double	71.24434
TOT_EMPT	Number of unloaded bidirectional trains per day	Double	31.21692
TOT_ASS	Number of total bidirectional trains per day (loaded+unloaded)	Double	102.4613
DIFF_VC	Difference between TOT_ASS and CAP	Double	-315.539
FLAG_OVCAP	Flag 0/1. 1 means the link is overcapacity by at least 1 train	Integer	0
FLAG_PER	Flag 0/1. 1 means the link is overcapacity by at least 1%	Integer	0
FLAG_TOT	Flag 0/1. 1 means that the FLAG_OVCAP=1 or FLAG_PER=1. Represent all the links with overcapacity conditions.	Integer	0

Table 100 – Format of "Report_15_RailLinksRCM" table.

Description of Use: The table is the source for " Report_15_Trains per day (tot, empty, loaded) (RCM) " under data panel Reports\Rail Capacity Management. This specific output is produced only for Rail Capacity Management.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog

Created by: "Results RCM" subgroup under "Samgods Model"

Made by user choice: No.

Report_16_{SCENARIO_SHORTNAME}_0 and Report_16_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_16_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_16_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1

Field	Description	Data Format	Example
ID_COUNTY	County code	Double	1
NAME	County name	Text	Stockholm
VKM_LGV3	VehicleKm Vehicle class 101	Double	773.22
VKM_MGV16	VehicleKm Vehicle class 102	Double	15643.87
VKM_MGV24	VehicleKm Vehicle class 103	Double	25213.88
VKM_HGV40	VehicleKm Vehicle class 104	Double	30262.45
VKM_HGV60	VehicleKm Vehicle class 105	Double	106313.66
VKM_ROAD	VehicleKm total road	Double	178207.09
VKM_LGV3P	Percentage Vehicle class 101 (VHM_LGV3/VHM_ROAD*100)	Double	0.43
VKM_MGV16P	Percentage Vehicle class 102 (VKM_MGV16/VHM_ROAD*100)	Double	8.78
VKM_MGV24P	Percentage Vehicle class 103 (VKM_MGV24/VHM_ROAD*100)	Double	14.15
VKM_HGV40P	Percentage Vehicle class 104 (VKM_HGV40/VHM_ROAD*100)	Double	16.98
VKM_HGV60P	Percentage Vehicle class 105 (VKM_HGV60/VHM_ROAD*100)	Double	59.66

Table 101 – Format of "Report_16_{SCENARIO_SHORTNAME}_0" and "Report_16_{SCENARIO_SHORTNAME}_RCM" table.

Description of Use: The two geodatabase tables are the source for " Report_16_VHCLKM and distribution by county - totals" under data panel Samgods Reports\Reports and RCM Reports\Reports.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcCatalog

Created by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: No.

Report_17_{SCENARIO_SHORTNAME}_0 and Report_17_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_17_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_17_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_COUNTY	County code	Double	1
NAME	County name	Text	Stockholm
VK_LGV3_11	VehicleKm Vehicle class 101	Double	442.74
VK_MGV16_11	VehicleKm Vehicle class 102	Double	10749.62
VK_MGV24_11	VehicleKm Vehicle class 103	Double	16649.97
VK_HGV40_11	VehicleKm Vehicle class 104	Double	22261.32
VK_HGV60_11	VehicleKm Vehicle class 105	Double	67905.24
VK_ROAD_11	VehicleKm total road	Double	118008.9
VK_LGV3P11	Percentage Vehicle class 101 (VHM_LGV3_11/VHM_ROAD_11*100)	Double	0.38
VK_MGV16P11N	Percentage Vehicle class 102 (VKM_MGV16_11/VHM_ROAD_11*100)	Double	9.11
VK_MGV24P11N	Percentage Vehicle class 103 (VKM_MGV24_11/VHM_ROAD_11*100)	Double	14.11
VK_HGV40P11N	Percentage Vehicle class 104 (VKM_HGV40_11/VHM_ROAD_11*100)	Double	18.86
VK_HGV60P11N	Percentage Vehicle class 105 (VKM_HGV60_11/VHM_ROAD_11*100)	Double	57.54

Table 102 – Format of "Report_17_{SCENARIO_SHORTNAME}_0" and "Report_17_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for Report_17_VHCLKM and distribution by county - E10 roads" under data panel Samgods Reports\Reports and RCM Reports\Reports. It contains the same results as per Report_16 but filtered on roads with CATEGORY=11 (E10 roads).

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcCatalog

Created by: "Results" and "Results RCM" subgroups under "Samgods Model"

Made by user choice: No.

Report_18_{SCENARIO_SHORTNAME}_0 and Report_18_{SCENARIO_SHORTNAME}_RCM

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_18_{SCENARIO_SHORTNAME}_0

Scenario_tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb\Report_18_{SCENARIO_SHORTNAME}_RCM

Table Format:

Field	Description	Data Format	Example
OBJECTID	Counter	Integer	1
ID_COUNTY	County code	Double	1
NAME	County name	Text	Stockholm
VK_LGV3_OR	VehicleKm Vehicle class 101	Double	330.48
VK_MGV16_ORN	VehicleKm Vehicle class 102	Double	4894.25
VK_MGV24_ORN	VehicleKm Vehicle class 103	Double	8563.91
VK_HGV40_ORN	VehicleKm Vehicle class 104	Double	8001.13
VK_HGV60_ORN	VehicleKm Vehicle class 105	Double	38408.42
VK_ROAD_OR	VehicleKm total road	Double	60198.19
VK_LGV3POR	Percentage Vehicle class 101 (VHM_LGV3_OR/VHM_ROAD_OR*100)	Double	0.55
VK_MGV16PORN	Percentage Vehicle class 102 (VKM_MGV16_ORN/VHM_ROAD_OR*100)	Double	8.13
VK_MGV24PORN	Percentage Vehicle class 103 (VKM_MGV24_ORN/VHM_ROAD_OR*100)	Double	14.23
VK_HGV40PORN	Percentage Vehicle class 104 (VKM_HGV40_ORN/VHM_ROAD_OR*100)	Double	13.29
VK_HGV60PORN	Percentage Vehicle class 105 (VKM_HGV60_ORN/VHM_ROAD_OR*100)	Double	63.8

Table 103 – Format of "Report_18_{SCENARIO_SHORTNAME}_0" and "Report_18_{SCENARIO_SHORTNAME}_RCM" tables.

Description of Use: The two geodatabase tables are the source for "Report_18_VHCLKM and distribution by country - Other roads" under data panel Samgods Reports\Reports and RCM Reports\Reports. It contains the same results as per Report_16 but filtered on roads with CATEGORY<>11.

Visualizing Options: Microsoft Access or Cube GIS window (see Help>Cube Base>GIS window), ArcCatalog, ArcCatalog

Created by: “Results” and “Results RCM” subgroups under “Samgods Model”

Made by user choice: No.

Results in matrix format

In the following table there is the list of matrices created by the model.

LOS matrices

Folder	Name of matrix file	Name of matrix	Description	User choice:
{Scenario_Dir}	COST_ROAD_{SCENARIO_SHORTNAME}.MAT	D101_{SCENARIO_SHORTNAME}, D102_{SCENARIO_SHORTNAME}, D103_{SCENARIO_SHORTNAME}, D104_{SCENARIO_SHORTNAME}, D105_{SCENARIO_SHORTNAME}	Distance (km) for vehicle type 101, …	Yes
		T101_{SCENARIO_SHORTNAME}, T102_{SCENARIO_SHORTNAME}, T103_{SCENARIO_SHORTNAME}, T104_{SCENARIO_SHORTNAME}, T105_{SCENARIO_SHORTNAME}	Time (hours) for vehicle type 101, …	
		X101_{SCENARIO_SHORTNAME}, X102_{SCENARIO_SHORTNAME}, X103_{SCENARIO_SHORTNAME}, X104_{SCENARIO_SHORTNAME}, X105_{SCENARIO_SHORTNAME}	Extra costs for vehicle type 101, …	
		DD101_{SCENARIO_SHORTNAME}, DD02_{SCENARIO_SHORTNAME}, DD103_{SCENARIO_SHORTNAME}, DD104_{SCENARIO_SHORTNAME}, DD105_{SCENARIO_SHORTNAME}	Domestic distance (km) for vehicle type 101, …	

Folder	Name of matrix file	Name of matrix	Description	User choice:
{Scenario_Dir}	COST_RAIL_{SCENARIO_SHORTNAME}.MAT	D201_{SCENARIO_SHORTNAME} D202_{SCENARIO_SHORTNAME} D204_{SCENARIO_SHORTNAME}, D205_{SCENARIO_SHORTNAME} D206_{SCENARIO_SHORTNAME}, D207_{SCENARIO_SHORTNAME}, D208_{SCENARIO_SHORTNAME},	Distance (km) for vehicle type 201, …	Yes
		T201_{SCENARIO_SHORTNAME}, T202_{SCENARIO_SHORTNAME}, T204_{SCENARIO_SHORTNAME}, T205_{SCENARIO_SHORTNAME}, T206_{SCENARIO_SHORTNAME} T207_{SCENARIO_SHORTNAME}, T208_{SCENARIO_SHORTNAME}	Time (hours) for vehicle type 201, …	
		X201_{SCENARIO_SHORTNAME} X202_{SCENARIO_SHORTNAME} X204_{SCENARIO_SHORTNAME} X205_{SCENARIO_SHORTNAME} X206_{SCENARIO_SHORTNAME} X207_{SCENARIO_SHORTNAME} X208_{SCENARIO_SHORTNAME}	Extra costs for vehicle type 201, …	
		DD201_{SCENARIO_SHORTNAME} DD202_{SCENARIO_SHORTNAME} DD204_{SCENARIO_SHORTNAME DD205_{SCENARIO_SHORTNAME} DD206_{SCENARIO_SHORTNAME} DD207_{SCENARIO_SHORTNAME DD208_{SCENARIO_SHORTNAME}	Domestic distance (km) for vehicle type 201, …	

Folder	Name of matrix file	Name of matrix	Description	User choice:
{Scenario_Dir}	COST_SEA_{SCENARIO_SHORTNAME}.MAT	D301D302_{SCENARIO_SHORTNAME} D303_{SCENARIO_SHORTNAME} D304_{SCENARIO_SHORTNAME} D305_{SCENARIO_SHORTNAME} D306_{SCENARIO_SHORTNAME} D307_{SCENARIO_SHORTNAME} D308_{SCENARIO_SHORTNAME} D309_{SCENARIO_SHORTNAME} D310_{SCENARIO_SHORTNAME} D311_{SCENARIO_SHORTNAME} D312_{SCENARIO_SHORTNAME} D313_{SCENARIO_SHORTNAME} D314_{SCENARIO_SHORTNAME} D315_{SCENARIO_SHORTNAME} D316_{SCENARIO_SHORTNAME} D317_{SCENARIO_SHORTNAME} D318_{SCENARIO_SHORTNAME}, D319_{SCENARIO_SHORTNAME} D320_{SCENARIO_SHORTNAME} D321_{SCENARIO_SHORTNAME},	Distance (km) for vehicle type 301, …	Yes

Folder	Name of matrix file	Name of matrix	Description	User choice:
		T301_{SCENARIO_SHORTNAME} T302_{SCENARIO_SHORTNAME} T303_{SCENARIO_SHORTNAME}, T304_{SCENARIO_SHORTNAME} T305_{SCENARIO_SHORTNAME} T306_{SCENARIO_SHORTNAME} T307_{SCENARIO_SHORTNAME} T308_{SCENARIO_SHORTNAME} T309_{SCENARIO_SHORTNAME} T310_{SCENARIO_SHORTNAME} T311_{SCENARIO_SHORTNAME} T312_{SCENARIO_SHORTNAME} T313_{SCENARIO_SHORTNAME} T314_{SCENARIO_SHORTNAME} T315_{SCENARIO_SHORTNAME} T316_{SCENARIO_SHORTNAME} T317_{SCENARIO_SHORTNAME} T318_{SCENARIO_SHORTNAME}	Time (hours) for vehicle type 301, …	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		X301_{SCENARIO_SHORTNAME} X302_{SCENARIO_SHORTNAME} X303_{SCENARIO_SHORTNAME}, X304_{SCENARIO_SHORTNAME} X305_{SCENARIO_SHORTNAME} X306_{SCENARIO_SHORTNAME}, X307_{SCENARIO_SHORTNAME} X308_{SCENARIO_SHORTNAME} X309_{SCENARIO_SHORTNAME}, X310_{SCENARIO_SHORTNAME} X311_{SCENARIO_SHORTNAME} X312_{SCENARIO_SHORTNAME} X313_{SCENARIO_SHORTNAME} X314_{SCENARIO_SHORTNAME} X315_{SCENARIO_SHORTNAME} X316_{SCENARIO_SHORTNAME} X317_{SCENARIO_SHORTNAME} X318_{SCENARIO_SHORTNAME}	Extra costs for vehicle type 301, …	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		DD301_{SCENARIO_SHORTNAME}, DD302_{SCENARIO_SHORTNAME}, DD303_{SCENARIO_SHORTNAME} DD304_{SCENARIO_SHORTNAME}, DD305_{SCENARIO_SHORTNAME}, DD306_{SCENARIO_SHORTNAME} DD307_{SCENARIO_SHORTNAME}, DD308_{SCENARIO_SHORTNAME}, DD309_{SCENARIO_SHORTNAME} DD310_{SCENARIO_SHORTNAME}, DD311_{SCENARIO_SHORTNAME}, DD312_{SCENARIO_SHORTNAME} DD313_{SCENARIO_SHORTNAME}, DD314_{SCENARIO_SHORTNAME}, DD315_{SCENARIO_SHORTNAME} DD316_{SCENARIO_SHORTNAME}, DD317_{SCENARIO_SHORTNAME}, DD318_{SCENARIO_SHORTNAME}	Domestic distance (km) for vehicle type 301, …	
{Scenario_Dir}	COST_AIR_{SCENARIO_SHORTNAME}.MAT	D401_{SCENARIO_SHORTNAME}	Distance (km) for vehicle type 401	Yes
		T401_{SCENARIO_SHORTNAME}	Time (hours) for vehicle type 401	
		X401_{SCENARIO_SHORTNAME}	Extra costs for vehicle type 401	
		DD01_{SCENARIO_SHORTNAME}	Domestic distance (km) for vehicle type 401	

Table 104 – LOS matrices.

Structure: Origin (on rows), Destination (on coulombs), Value (18.6 double) on matrix cell

Description of Use: to store the LOS matrices at the end of cost calculation phase

Visualizing Options: Cube Matrix window (see Help>Cube Base>Matrix window)

Created by: “LOS calculation” subgroup under “Samgods Model”

Made by user choice: Yes

Trip Matrices from Standard Logistic Module

Folder	Name of matrix file	Name of matrix	Description	User choice:
{Scenario_Dir}	ROAD_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT	101_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 101 per year	No
		102_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 102 per year	
		103_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 103 per year	
		104_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 104 per year	
		105_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 105 per year	
		Road_{SCENARIO_SHORTNAME}	OD Loaded Vehicles total road mode per year	
{Scenario_Dir}	RAIL_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT	201_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 201 per year	No
		202_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 202 per year	
		204_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 204 per year	
		205_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 205 per year	
		206_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 206 per year	
		207_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 207 per year	
		208_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 208 per year	
		209_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 209 per year	
		Rail_{SCENARIO_SHORTNAME}	OD Loaded Vehicles total rail mode per year	
{Scenario_Dir}	SEA_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT	301_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 301 per year	No
		302_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 302 per year	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		303_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 303 per year	
		304_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 304 per year	
		305_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 305 per year	
		306_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 306 per year	
		307_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 307 per year	
		308_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 308 per year	
		309_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 309 per year	
		310_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 310 per year	
		311_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 311 per year	
		312_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 312 per year	
		313_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 313 per year	
		314_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 314 per year	
		315_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 315 per year	
		316_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 316 per year	
		317_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 317 per year	
		318_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 318 per year	
		319_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 319 per year	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		320_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 320 per year	
		321_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 321 per year	
		Sea_{SCENARIO_SHORTNAME}	OD Loaded Vehicles total Sea mode per year	
{Scenario_Dir}	AIR_VHCLFLOW0_{SCENARIO_SHORTNAME}.MAT	Air_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 401 per year. (mode air)	No
{Scenario_Dir}	ROAD_TON0_{SCENARIO_SHORTNAME}.MAT	101_{SCENARIO_SHORTNAME}	OD Tons 101 per year	No
		102_{SCENARIO_SHORTNAME}	OD Tons 102 per year	
		103_{SCENARIO_SHORTNAME}	OD Tons 103 per year	
		104_{SCENARIO_SHORTNAME}	OD Tons 104 per year	
		105_{SCENARIO_SHORTNAME}	OD Tons 105 per year	
		Road_{SCENARIO_SHORTNAME}	OD Tons total road mode per year	
{Scenario_Dir}	RAIL_TON0_{SCENARIO_SHORTNAME}.MAT	201_{SCENARIO_SHORTNAME}	OD Tons 201 per year	No
		202_{SCENARIO_SHORTNAME}	OD Tons 202 per year	
		204_{SCENARIO_SHORTNAME}	OD Tons 204 per year	
		205_{SCENARIO_SHORTNAME}	OD Tons 205 per year	
		206_{SCENARIO_SHORTNAME}	OD Tons 206 per year	
		207_{SCENARIO_SHORTNAME}	OD Tons 207 per year	
		208_{SCENARIO_SHORTNAME}	OD Tons 208 per year	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		209_{SCENARIO_SHORTNAME}	OD Tons 209 per year	
		Rail_{SCENARIO_SHORTNAME}	OD Tons total rail mode per year	
{Scenario_Dir}	SEA_TON0_{SCENARIO_SHORTNAME}.MAT	301_{SCENARIO_SHORTNAME}	OD Tons 301 per year	No
		302_{SCENARIO_SHORTNAME}	OD Tons 302 per year	
		303_{SCENARIO_SHORTNAME}	OD Tons 303 per year	
		304_{SCENARIO_SHORTNAME}	OD Tons 304 per year	
		305_{SCENARIO_SHORTNAME}	OD Tons 305 per year	
		306_{SCENARIO_SHORTNAME}	OD Tons 306 per year	
		307_{SCENARIO_SHORTNAME}	OD Tons 307 per year	
		308_{SCENARIO_SHORTNAME}	OD Tons 308 per year	
		309_{SCENARIO_SHORTNAME}	OD Tons 309 per year	
		310_{SCENARIO_SHORTNAME}	OD Tons 310 per year	
		311_{SCENARIO_SHORTNAME}	OD Tons 311 per year	
		312_{SCENARIO_SHORTNAME}	OD Tons 312 per year	
		313_{SCENARIO_SHORTNAME}	OD Tons 313 per year	
		314_{SCENARIO_SHORTNAME}	OD Tons 314 per year	
		315_{SCENARIO_SHORTNAME}	OD Tons 315 per year	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		316_{SCENARIO_SHORTNAME}	OD Tons 316 per year	
		317_{SCENARIO_SHORTNAME}	OD Tons 317 per year	
		318_{SCENARIO_SHORTNAME}	OD Tons 318 per year	
		319_{SCENARIO_SHORTNAME}	OD Tons 319 per year	
		320_{SCENARIO_SHORTNAME}	OD Tons 320 per year	
		321_{SCENARIO_SHORTNAME}	OD Tons 321 per year	
		Sea_{SCENARIO_SHORTNAME}	OD Tons total Sea mode per year	
{Scenario_Dir}	AIR_TON0_{SCENARIO_SHORTNAME}.MAT	Air_{SCENARIO_SHORTNAME}	OD Tons 401 per year. (mode air)	No
{Scenario_Dir}	ROAD_EMP0_{SCENARIO_SHORTNAME}.MAT	101_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 101 per year	No
		102_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 102 per year	
		103_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 103 per year	
		104_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 104 per year	
		105_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 105 per year	
		Road_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles total road mode per year	
{Scenario_Dir}	RAIL_EMP0_{SCENARIO_SHORTNAME}.MAT	201_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 201 per year	No
		202_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 202 per year	
		204_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 204 per year	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		205_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 205 per year	
		206_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 206 per year	
		207_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 207 per year	
		208_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 208 per year	
		209_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 209 per year	
		Rail_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles total rail mode per year	
{Scenario_Dir}	SEA_EMP0_{SCENARIO_SHORTNAME}.MAT	301_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 301 per year	No
		302_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 302 per year	
		303_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 303 per year	
		304_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 304 per year	
		305_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 305 per year	
		306_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 306 per year	
		307_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 307 per year	
		308_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 308 per year	
		309_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 309 per year	
		310_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 310 per year	
		311_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 311 per year	

Folder	Name of matrix file	Name of matrix	Description	User choice:
		312_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 312 per year	
		313_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 313 per year	
		314_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 314 per year	
		315_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 315 per year	
		316_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 316 per year	
		317_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 317 per year	
		318_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 318 per year	
		319_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 319 per year	
		320_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 320 per year	
		321_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 321 per year	
		Sea_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles total Sea mode per year	
{Scenario_Dir}	AIR_EMP0_{SCENARIO_SHORTNAME}.MAT	Air_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 401 per year. (mode air)	No

Table 105 – Trip Matrices from Standard Logistic Module.

Structure: Origin (on rows), Destination (on coulombs), Value (18.6 double) on matrix cell

Description of Use: to store the trip matrices calculated by standard logistic module

Visualizing Options: Cube Matrix window (see Help>Cube Base>Matrix window)

Created by: “Conversion from LogMod to VY” subgroup under “Samgods Model\Assignment”

Made by user choice: No

Trip Matrices from Rail Capacity Management

Folder	Name of matrix file	Name of matrix	Description	Made by user:
{Scenario_Dir}	ROAD_VHCLFLOW_FIN_{SCENARIO_SHORTNAME}.MAT	101_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 101 per year	No
		102_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 102 per year	
		103_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 103 per year	
		104_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 104 per year	
		105_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 105 per year	
		Road_{SCENARIO_SHORTNAME}	OD Loaded Vehicles total road mode per year	
{Scenario_Dir}	RAIL_VHCLFLOW_FIN_{SCENARIO_SHORTNAME}.MAT	201_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 201 per year	No
		202_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 202 per year	
		204_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 204 per year	
		205_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 205 per year	
		206_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 206 per year	
		207_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 207 per year	
		208_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 208 per year	
		209_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 209 per year	
		Rail_{SCENARIO_SHORTNAME}	OD Loaded Vehicles total rail mode per year	
{Scenario_Dir}	SEA_VHCLFLOW_FIN_{SCENARIO_SHORTNA	301_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 301 per year	No

Folder	Name of matrix file	Name of matrix	Description	Made by user:
	ME}.MAT	302_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 302 per year	
		303_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 303 per year	
		304_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 304 per year	
		305_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 305 per year	
		306_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 306 per year	
		307_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 307 per year	
		308_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 308 per year	
		309_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 309 per year	
		310_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 310 per year	
		311_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 311 per year	
		312_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 312 per year	
		313_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 313 per year	
		314_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 314 per year	
		315_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 315 per year	
		316_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 316 per year	
		317_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 317 per year	
		318_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 318 per year	
		319_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 319 per year	

Folder	Name of matrix file	Name of matrix	Description	Made by user:
		320_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 320 per year	
		321_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 321 per year	
		Sea_{SCENARIO_SHORTNAME}	OD Loaded Vehicles total Sea mode per year	
{Scenario_Dir}	AIR_VHCLFLOW_FIN_{SCENARIO_SHORTNAME}.MAT	Air_{SCENARIO_SHORTNAME}	OD Loaded Vehicles 401 per year. (mode air)	No
{Scenario_Dir}	ROAD_TON_FIN_{SCENARIO_SHORTNAME}.MAT	101_{SCENARIO_SHORTNAME}	OD Tons 101 per year	No
		102_{SCENARIO_SHORTNAME}	OD Tons 102 per year	
		103_{SCENARIO_SHORTNAME}	OD Tons 103 per year	
		104_{SCENARIO_SHORTNAME}	OD Tons 104 per year	
		105_{SCENARIO_SHORTNAME}	OD Tons 105 per year	
		Road_{SCENARIO_SHORTNAME}	OD Tons total road mode per year	
{Scenario_Dir}	RAIL_TON_FIN_{SCENARIO_SHORTNAME}.MAT	201_{SCENARIO_SHORTNAME}	OD Tons 201 per year	No
		202_{SCENARIO_SHORTNAME}	OD Tons 202 per year	
		204_{SCENARIO_SHORTNAME}	OD Tons 204 per year	
		205_{SCENARIO_SHORTNAME}	OD Tons 205 per year	
		206_{SCENARIO_SHORTNAME}	OD Tons 206 per year	
		207_{SCENARIO_SHORTNAME}	OD Tons 207 per year	
		208_{SCENARIO_SHORTNAME}	OD Tons 208 per year	
		209_{SCENARIO_SHORTNAME}	OD Tons 209 per year	

Folder	Name of matrix file	Name of matrix	Description	Made by user:
		Rail_{SCENARIO_SHORTNAME}	OD Tons total rail mode per year	
{Scenario_Dir}	SEA_TON_FIN_{SCENARIO_SHORTNAME}.MAT	301_{SCENARIO_SHORTNAME}	OD Tons 301 per year	No
		302_{SCENARIO_SHORTNAME}	OD Tons 302 per year	
		303_{SCENARIO_SHORTNAME}	OD Tons 303 per year	
		304_{SCENARIO_SHORTNAME}	OD Tons 304 per year	
		305_{SCENARIO_SHORTNAME}	OD Tons 305 per year	
		306_{SCENARIO_SHORTNAME}	OD Tons 306 per year	
		307_{SCENARIO_SHORTNAME}	OD Tons 307 per year	
		308_{SCENARIO_SHORTNAME}	OD Tons 308 per year	
		309_{SCENARIO_SHORTNAME}	OD Tons 309 per year	
		310_{SCENARIO_SHORTNAME}	OD Tons 310 per year	
		311_{SCENARIO_SHORTNAME}	OD Tons 311 per year	
		312_{SCENARIO_SHORTNAME}	OD Tons 312 per year	
		313_{SCENARIO_SHORTNAME}	OD Tons 313 per year	
		314_{SCENARIO_SHORTNAME}	OD Tons 314 per year	
		315_{SCENARIO_SHORTNAME}	OD Tons 315 per year	
		316_{SCENARIO_SHORTNAME}	OD Tons 316 per year	
		317_{SCENARIO_SHORTNAME}	OD Tons 317 per year	

Folder	Name of matrix file	Name of matrix	Description	Made by user:
		318_{SCENARIO_SHORTNAME}	OD Tons 318 per year	
		319_{SCENARIO_SHORTNAME}	OD Tons 319 per year	
		320_{SCENARIO_SHORTNAME}	OD Tons 320 per year	
		321_{SCENARIO_SHORTNAME}	OD Tons 321 per year	
		Sea_{SCENARIO_SHORTNAME}	OD Tons total Sea mode per year	
{Scenario_Dir}	AIR_TON_FIN_{SCENARIO_SHORTNAME}.MAT	Air_{SCENARIO_SHORTNAME}	OD Tons 401 per year. (mode air)	No
{Scenario_Dir}	ROAD_EMP_FIN_{SCENARIO_SHORTNAME}.MAT	101_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 101 per year	No
		102_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 102 per year	
		103_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 103 per year	
		104_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 104 per year	
		105_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 105 per year	
		Road_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles total road mode per year	
{Scenario_Dir}	RAIL_EMP_FIN_{SCENARIO_SHORTNAME}.MAT	201_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 201 per year	No
		202_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 202 per year	
		204_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 204 per year	
		205_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 205 per year	
		206_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 206 per year	
		207_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 207 per year	

Folder	Name of matrix file	Name of matrix	Description	Made by user:
		208_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 208 per year	
		209_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 209 per year	
		Rail_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles total rail mode per year	
{Scenario_Dir}	SEA_EMP_FIN_{SCENARIO_SHORTNAME}.MAT	301_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 301 per year	No
		302_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 302 per year	
		303_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 303 per year	
		304_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 304 per year	
		305_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 305 per year	
		306_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 306 per year	
		307_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 307 per year	
		308_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 308 per year	
		309_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 309 per year	
		310_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 310 per year	
		311_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 311 per year	
		312_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 312 per year	
		313_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 313 per year	
		314_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 314 per year	
		315_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 315 per year	

Folder	Name of matrix file	Name of matrix	Description	Made by user:
		316_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 316 per year	
		317_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 317 per year	
		318_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 318 per year	
		319_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 319 per year	
		320_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 320 per year	
		321_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 321 per year	
		Sea_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles total Sea mode per year	
{Scenario_Dir}	AIR_EMP_FIN_{SCENARIO_SHORTNAME}.MAT	Air_{SCENARIO_SHORTNAME}	OD Empty Loaded Vehicles 401 per year. (mode air)	No

Table 106 – Trip Matrices from Rail Capacity Management.

Structure: Origin (on rows), Destination (on coulombs), Value (18.6 double) on matrix cell

Description of Use: to store the trip matrices calculated by Rail Capacity Management module

Visualizing Options: Cube Matrix window (see Help>Cube Base>Matrix window)

Created by: “Conversion from LogMod to VY” subgroup under “Samgods Model\RCM Assignment”

Made by user choice: No

Other matrices

All the following matrices:

COMPARE_LOS_ROAD_{SCENARIO_SHORTNAME}_Base2006

COMPARE_LOS_RAIL_{SCENARIO_SHORTNAME}_Base2006

COMPARE_LOS_SEA_{SCENARIO_SHORTNAME}_Base2006

COMPARE_LOS_AIR_{SCENARIO_SHORTNAME}_Base2006

COMPARE_OD_ROAD_X_{SCENARIO_SHORTNAME}_Base2006

COMPARE_OD_RAIL_X_{SCENARIO_SHORTNAME}_Base2006

COMPARE_OD_SEA_X_{SCENARIO_SHORTNAME}_Base2006

COMPARE_OD_AIR_X_{SCENARIO_SHORTNAME}_Base2006

COMPARE_TON_ROAD_X_{SCENARIO_SHORTNAME}_Base2006

COMPARE_TON_RAIL_X_{SCENARIO_SHORTNAME}_Base2006

COMPARE_TON_SEA_X_{SCENARIO_SHORTNAME}_Base2006

COMPARE_TON_AIR_X_{SCENARIO_SHORTNAME}_Base2006

Will be created if the user wishes to verify differences between scenarios.

Structure: Origin (on rows), Destination (on coulombs), Value (18.6 double) on matrix cell.

Description of Use: store the differences between scenarios for the same matrix. The differences are calculated with the following equation:

Dif_of_pair_value=OD_pair_value current scenario – OD_pair_value base scenario (i.e. values greater than zero mean that the OD pair value in the current scenario is greater than the base scenario).

Visualizing Options: Cube Matrix window (see Help>Cube Base>Matrix window).

Created by: ”Compare” application.

Made by user choice: Yes.

Other output related to matrices

Each matrix listed in the point 0and 0 is also exportable in csv or dbf format. The exported matrices will have the following file name:

{Scenario_DIR}\{Type of matrix}_{Type of mode}_{Matrix name}{Vhclnubmber}_VOY.CSV (or DBF).

And the file format will be:

Field	Description	Data Format
Origin	Origin zone	Integer
Destination	Destination zone	Integer
Value	Value	Double

Table 107 – Matrix format.

Visualizing Options: Microsoft excel.

Created by: ”Change matrix format” application

Made by user choice: Yes

Report files

Under scenario folder after have run all the application in the model will be three files:

- Report_SG_BS05.txt
- Report_ED_BS05.txt
- Report_HL_BS05.txt

For the description of the reports created and their meaning refer to the user manual.

Temporary data to visualize scenario data

Under scenario folder after have run all the application in the model will be four files:

- Input_data.mxr and Input_data.mxd, Scenario_Data.vpr
- Scenario_Data.mdb

The first group has the purpose to define default legends to visualize in the GIS window all the data.

The second file is the temporary geodatabase with the scenario data created by join of the scenario specific tables and base tables, in other words the real scenario data. It could be exist or not based on the user choices. It is used only by the “Edit the data application” and no other applications point to it.

LogMod folder

The structure and input/output files for LogMod folder is covered by document listed under Section 0.4. . Here will be listed the input and output files highlighting the outputs related to the Rail Capacity Management. For a description of this procedure please refer to chapter 0.

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y			Main folder with all the inputs and outputs for Logistics Module and Rail Capacity Management Module. Executable and input files for parallelization of processes and log file from RCM procedure Error.log
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	BUILDCHAIN		Folder with all the control files and executable for BuildChain.exe and BuildChain4RCM.exe The control files are two groups: • BuildChain_Common.ctl, BuildChain_STAN2.ctl, BuildChain_STAN8.ctl and BuildChain_STAN9.ctl • BuildChainXX.ctl The first group has general parameters for all the commodities or specific STAN groups. The second one specific settings per each commodity (XX).
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	BUILDCHAIN	OUTPUT	BuildChainxx.log (only if Do you want to produce BuildChainxx.log files? catalog key set to 'yes' or 'all') Chainsxx.dat Connectionxx.lst (related to Select.dat)
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	ChainChoi		Folder with all the control files and executable for ChainChoi.exe, ChainChoi4RCM.exe and LP2CC.exe The control files are two groups: • ChainChoi_Common.ctl, ChainChoi_STAN2.ctl, ChainChoi_STAN8.ctl and ChainChoi_STAN9.ctl • ChainChoiXX.ctl The first group has general parameters for all the commodities or specific STAN groups. The second one specific settings per each commodity (XX).
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	ChainChoi	OUTPUT	From ChainChoi.exe: ChainChoixx.fac ChainChoixx.rep ChainChoixx_01.out ChainChoixx_02.out ChainChoixx_03.out ChainChoixx_04.out ChainChoixx_05.out ChainChoixx_data_06.out ChainChoixx_data_07.out Connectionxx.lst (related to Select.dat)

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					From ChainChoi4RCM.exe ChainChoi01_xxLPX.out (as ChainChoixx_01.out with solutions from LP) From LP2CC.exe ChainChoixx_01FIN.out ChainChoixx_data_06FIN.out ChainChoixx_data_07FIN.out
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	ChainChoi	OUTPUT\CoVo	CONSOLXX_CHAINTYPE.314 VOLUMEXX_ CHAINTYPE.314
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	EXTRACT		Folder with all the control files and executable for extract. The control files are three groups: • EXTRACTYYY.ctl - Standard Log Mod • EXTRACTYYY_LP0.ctl - LP0 • EXTRACTYYYFIN.ctl - Final • EXTRACTYYY_LP1.ctl - LP1 • EXTRACTYY_LP1AdjC1.CTL - LP1Adj1 Per each process a different set of control files is produced per vehicle type (YYY). For LP0, LP1, LP1AdjC1 etc only vehicle classes 201-209 are used.
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	EXTRACT	OUTPUT	The first set of files represent the OD matrices for empties, tonnes and total vehicles and has been produced by extract procedure. OD_EmpXXX_STD.314 OD_EmpXXX_FIN.314 OD_EmpXXX_LPX.314 OD_TonnesXXX_STD.314 OD_TonnesXXX_LPX.314 OD_TonnesXXX_FIN.314 OD_VhclXXX_STD.314 OD_VhclXXX_FIN.314 OD_VhclXXX_LPX.314 The second set: ODEmpAddOn_STD.300 ODEmpAddOn_LPX.300 Is the OD add on factors for Rail Capacity Management procedure and are produced by MPS.jar file (under RCM folder).
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	INPUT		Chaintype.lis is used by Standard Logistics Module and Rail Capacity Management Module. CalibrationParameters.txt and PortAreas_def.txt are used to rescale LOS matrices using the scaling factors in the first file and the port area definition from second file. They are input for MPS.jar file (process controlled by " Port

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					Area LOS inputs (STAN)" subgroup in Samgods\Logistics Module.
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	INPUT	COST	Cargo.txt PilotFees.txt Vhcls_dry_bulk.txt Vhcls_dry_bulk_iron.txt Vhcls_dry_bulk_S10_11.txt Vhcls_gen_cargo.txt Vhcls_gen_cargo_STEEL.txt Vhcls_Liq_bulk.txt
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	INPUT	GENERAL	averageVehicleCpacity.txt
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	INPUT	LOS	FreqAir.314 FreqCombi.314 FreqContainerVessel.314 FreqLorry.314 FreqOtherVessel.314 FreqRailFerry.314 FreqRoadFerry.314 FreqRoRoVessel.314 FreqSystem.314 FreqWaggonlod.314 vXXX_ddist.314 vXXX_dist.314 vXXX_timeh.314 vXXX_xkr.314
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	INPUT	LOS\S01 LOS\S.. LOS\S12	FreqAir.314 FreqCombi.314 FreqContainerVessel.314 FreqLorry.314 FreqOtherVessel.314 FreqRailFerry.314 FreqRoadFerry.314 FreqRoRoVessel.314 FreqSystem.314 FreqWaggonlod.314 vXXX_ddist.314 vXXX_dist.314 vXXX_timeh.314 vXXX_xkr.314 Per each STAN group (denoted as S01-S12 folders) a different set of LOS is applied. In each folder the Vxxx_timeh.314 values are updated for vehicles 301-

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					321 with scaling factors in CalibrationParameters.txt
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	INPUT	NODES	ContainerHandling.txt DirectFeederTrain.txt DirectSea.txt DirectSystemTrain.txt DirectWagonLoad.txt Nodes.txt TransferCombiSea.txt transferfeedertrainwagonload.txt TransferRoadAir.txt TransferRoadCombi.txt TransferRoadRoad.txt TransferRoadRoadFerry.txt TransferRoadSea.txt TransferRoadTrain.txt TransferSeaSea.txt TransferSystemTrainSea.txt TransferWagonloadRailFerry.txt TransferWagonloadSea.txt In BuildChain will be used this set of files.
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	INPUT	NODES_CC	ContainerHandling.txt DirectFeederTrain.txt DirectSea.txt DirectSystemTrain.txt DirectWagonLoad.txt Nodes.txt TransferCombiSea.txt transferfeedertrainwagonload.txt TransferRoadAir.txt TransferRoadCombi.txt TransferRoadRoad.txt TransferRoadRoadFerry.txt TransferRoadSea.txt TransferRoadTrain.txt TransferSeaSea.txt TransferSystemTrainSea.txt TransferWagonloadRailFerry.txt TransferWagonloadSea.txt Same files as NODES folder except Nodes.txt. Here the capacity for some sea ports has revised to a limit. In ChainChoi will be used this set of files.
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	LOG		Folder with all .log files from Standard Logistics Module and Rail Capacity Management module.

Catalog Folder	First Level	Second Level	Third Level	Fourth Level	Description
					logXXFIN.log logXXSTD.log
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	MERGEREP		Control files for Mergerep.exe program.
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	MERGEREP	OUTPUT	ChainChoi.rep ChainChoiFIN.rep
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	RCM		Folder with MPS.jar and LP_solve.exe programs together: • control files for different steps: ○ MPS_LOS.CTL - to rescale costs for sea mode ○ mps_LP.ctl - control file for LP0, LP1, etc ○ mps_LP1Adj1.ctl - control file for LP1 AdjC1. • Input data : ○ Spanning tree data for all modes (PathTreeRail.txt, PathTreeSea.txt, PathTreeRoad.txt and PathTreeAir.txt) ○ Links_List.txt and Nodes_List.txt ○ LockedSTDLogMod_empty.txt ○ RailLinkCapacitiesBidirectional_STD.dat • Intermediate files ○ • log files For a complete list see chapter 0
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	RCM	ColumnMerge	Intermediate files
C:\SamGods_v10	{Scenario_Folder}	LogMod_Y	RCM	ColumnData	Intermediate files

Table 108 – LogMod folder.

Other files with different purposes

Under scenario folder a set of files is present with different purposes:

- calibration procedure files
- exported information in dbf, emme, csv format from geodatabase (to exchange the data with users without Cube Interface)
- general results from Standard Logistics Module and Rail Capacity Management Module
- warm start procedure
- debugging

Calibration procedure files

The Calibration procedure invokes a cycling process where scaling factors for Port Areas and STAN groups together the Kiel canal scaling factor are altered along the run. The process produces some txt files that keep the scaling factor information in each loop.

The scaling factors for port areas and STAN groups are saved in the following files:

- CalParameter_Loop_1.txt - parameters in the first loop - this file is a copy of the file specified in " Initial values for port calibration by Port Area and STAN group:", in the current model " PortAreasParameters_22092014_L1_Base2006R21_RUN20.txt" under 05_Inputs\Calibration
- CalParameter_Loop_2.txt - parameters estimated after completion of loop 1
- CalParameterNextLoop.txt - parameters used in next loop - identical to CalParameter_loop_"lastloop".txt, in the example lastloop=2

The scaling factor for Kiel Canal has a different treatment since is only one value:

- The initial value is provided by the interface with catalog key " Starting value for scaling factor on Kiel canal" is copied in ParKiel.txt file;
- KielNextValue.txt holds the scaling factor along all the iterations. Its structure is LoopNbr Scaling factor. The value used is obtained looking up LoopNbr.

The following files contain statistics on the calibration procedure:

- Differences_{Scenario_code}.csv - differences in tons between modelled and statistics per port area and STAN group
- PortArea_report_{Scenario_code}_loop1.csv - same information present in Report_13_Tons_per_PortArea_and_STAN_Group,
- PortArea_report_RCM_{Scenario_cose}_Loop1.csv - same information present in Report_13_Tons_per_PortArea_and_STAN_Group (RCM)

Exported files

The following files represent information saved in main geodatabase and exported to allow access to information for modellers or users without a valid Cube Licence.

- Capacity_table_{Scenario_Code}.dbf - scenario specific table with ID_LINK and CAPACITY for all rail links. This is the export of Rail_Capacity_{Scenario_short_name} (merged table between Rail_Capacity_Base2006 and SC_Rail_Capacity_{Scenario_short_name}). See 2.1.2.8 for table structure.

- Rail_capacities_STD_{Scenario_Code}.DBF - scenario specific table with initial values from Adjust Procedure. Rail_Capacity_{Scenario_short_name} (merged table between Rail_Capacity_Base2006 and SC_Rail_Capacity_{Scenario_short_name}). See 2.1.2.8 for table structure.
- EMME_NET_Base2006.211 and V101_102_LINK_SPEEDS.DAT: Scenario network exported in EMME format (211) together the speed definitions per each link in DAT format. The speed table has the following headers " Inode Jnode v101 v102" where:
  - Inode=Initial emme node number
  - Jnode=final emme node number
  - V101 = speed in km/h for vehicle class 101
  - V102 = speed in km/h for vehicle classes 102-105
- General_Table_{SCENARIO_SHORTNAME}.DBF is the exported table General_{Scenario_short_name} containing all the main parameters of the model. It is used in the Samgods application when is required to read a general variable.
- RENUM_NODES.DBF has the same information of RENUM_NODES table. This file could be link in the Matrix Window showing the emme zone number instead of the voyager node numbers. For details refer to Matrix Window > Using zone labels on the Cube Help.

Reports from STD Logistics Module and Rail Capacity Management Module

Two report files are produced using MERGE.exe program:

- ChainChoi0_Base2006.rep
- ChainChoi0_Base2006FIN.rep

For their description refer to

Each report contains statistics at vehicle type level, chain level and commodity level. These files are saved as reports (Report_2_Logistics Module, Report_2_LM_CHAINS, Report_2_LM_DEMAND) and tables (VHCL_OD_COV, CHAIN_OD_COV, COM_L_D) in the model.

Warm start procedure

The cycling process for LP could be restart after a previous run from a different point. For instance if in a previous run only LP0 has been performed, it is possible to run Rail Capacity Management module adding LP1. To recode the history of previous run, the type of stages performed is recoded in Log_file.log.

This file is maintained by the system and the user must never change its content.

Each step (Standard Logistic Module, LP0, LP1, LP1Adjust and so forth) produces a line with a different code. These codes are used to manage the right execution order.

The codes associated to each step are:

LP	LPAdjust	Description	Date	Time
99	99	STD	16-Feb-15	1:06 AM
-1	0	LP0	16-Feb-15	1:57 AM
99	99	FIN	16-Feb-15	6:52 AM
1	0	LP1	17-Feb-15	8:52 AM
1	1	LP1Adjust1	17-Feb-15	
1	2	LP2Adjust2	18-Feb-15	

Table 109 – Codes under log_file_x.log.

logfile_scenario.dat contains the last values printed out from the model, it is a temporary file. Information is later on saved in log_file_x.log Per each loop a different log file is produced and maintained.

Debugging

Some intermediate outputs are saved in scenario folder to allow checks on some complex calculations.

The calculation carried out in application Samgods\LOS Calculation\Data preparation leads to a network containing all the tolls and taxes defined with the different input tables (TAX_by_link, TAX_by_CATEGORY, TAX_by_COUNTRY, TOLL_BY_LINK and user extra attributes EC_V and EC_KM_V) together the different rules to open or close a particular link to a particular vehicle class. The intermediate output with all the information required for the next steps (LOS calculation, assignment in Standard Logistics Module and Rail Capacity Management Module) is saved under scenario folder with name FINAL_NETWORK_{Scenario_Code}.NET. This file can be accessed from Data Panel under Scenario Outputs\Other\ Network with costs for checking.

The .NET file is a binary format used in Cube Interface, therefore only Cube can open and visualize the network in the Network Window (Graphics). A part this, the structure is identical to a Geodatabase network. In the network are present the node and link tables.

Node Table Format:

Field	Description	Data Format	Example
N	Node number	Integer	1
NORIG	Original node number based on the combination of ID_Country, ID_region and MODE_N values	Double	711400
SCBSTANN	SCBSTANN code	Double	114
ID_REGION	Region code	Double	114
MODE_N	Mode code	Double	0
UI4	User field (not used in the current model)	Double	0
CENTRALL	Node description in terms of location	Text	Upplands-Väsby
X	Coordinate x	Double	1620000
Y	Coordinate y	Double	6601000

Table 110 – Format of "FINAL_NETWORK_{Scenario_Code}.NET.node" table.

Link Table Format:

Field	Description	Data Format	Example
AX/BX	Minimum X coordinate	double	10548
AY/BY	Minimum Y coordinate	double	257
A	Starting node	Integer	2231
B	End node	Integer	2353
OBJECTID	ID object	double	625
ID_LINK	Identification code for rail domestic link	Double	313
ORIGINALCAP	Original value for capacity in bidirectional trains per day	Double	23
M_101	Flag 0/1 If 1 open to 101	Integer	0
M_102	Flag 0/1 If 1 open to 102	Integer	0
M_103	Flag 0/1 If 1 open to 103	Integer	0
M_104	Flag 0/1 If 1 open to 104	Integer	0

Field	Description	Data Format	Example
M_105	Flag 0/1 If 1 open to 105	Integer	0
M_201	Flag 0/1 If 1 open to 201	Integer	1
M_202	Flag 0/1 If 1 open to 202	Integer	3
M_204	Flag 0/1 If 1 open to 204	Integer	1
M_205	……	Integer	1
M_401	Flag 0/1 If 1 open to 401	Integer	0
MODESTR	String with all the allowed modes	Text	xdwhi
SPEED_1	Speed in kms per hour for all modes except v102-105 in Sweden	Double	70
SPEED_2	Speed in kms per hour for vehicle types v102-105 in Sweden	Double	0
CATEGORY	Link category	Double	70
FUNCTION	Index for travel time function	Double	66
NLANES	Number of lanes (may be a decimal number)	Double	1
UL2	Distance in km	Double	70.2
UL3	Capacity for vessels on sea (dwell tons)	Double	0
ID_COUNTRY	Country code	Double	1
EC_V101	Extra cost on link (defined via EC_V attribute in the network) for v101		-99999
EC_KM_V101	Extra costkm on link (defined via EC_KM_V attribute in the network) for v101		-99999
EC_V102	Extra cost on link (defined via EC_V attribute in the network) for v102		-99999
EC_KM_V102	Extra costkm on link (defined via EC_KM_V attribute in the network) for v102		-99999
EC_V103	…..		-99999
EC_KM_V103	……		-99999
AORIG	Emme initial node number	Double	4246
BORIG	Emme final node number	Double	9088
KM_TAX101	Sum of TAX_C, EC_KM_V for V101		0
TOLL101	Sum of TAX_LT, TAX_L, EC_V,TOLL_L for V101		0
KM_TAX102	Sum of TAX_C, EC_KM_V for V102		0
TOLL102	Sum of TAX_LT, TAX_L, EC_V,TOLL_L for V102		0
KM_TAX103	…..		0
TOLL103	…..		0
KM_TAX401	Sum of TAX_C, EC_KM_V for V401		0
TOLL401	Sum of TAX_LT, TAX_L, EC_V,TOLL_L for V401		0
MODE_L	Mode		2

Table 111 – Format of "FINAL_NETWORK_{Scenario_Code}.NET.Link" table.

The variables listed above has the following meaning:

- TAX_C: country tax based on Tax_Country_{Scenario_short_name} table - SEK/km
- TAX_LT: tax by category on Tax_Category_{Scenario_short_name} table - SEK
- TAX_L: tax by link on Tax_Link_{Scenario_short_name} table - SEK
- TOLL_L: toll on link from Toll_Link_{Scenario_short_name} table - SEK
- EC_V: extra cost applied using EC_V attribute in the network (see user manual for details) - SEK

- EC_KM_V: extra cost km applied using EC_KM_V attribute in the network (see user manual for details) - SEK

Those taxes and tolls are recombined into two attributes:

- KM_TAX: includes TAX_C and EC_KM_V, in other words values based on distance
- TOLL: includes TAX_LT, TAX_L , TOLL_L and EC_V, therefore values based on a link

The final results of KM_TAX and TOLL are not merely the sum of the individual values since TAX_C, TAX_LT and TAX_L mutually exclude each other. We have the following cases:

- if TAX_L is specified, TAX_C and TAX_LT will be not applied
- if TAX_L is not specified and TAX_LT has a value, only TAX_LT will be applied
- if TAX_L and TAX_LT are not specified, only TAX_C will be applied.

Also values equal to zero have a meaning, since a zero-valued tax could be specified by TAX_L=0, and that will give a tax level at 0. To treat this case all the values are initialized to -99999, before applying any specific values from tables in geodatabase..

The model structure

The Samgods model could be described as a set of different software integrated in a standalone interface. The technical documentation refers to the GUI interface implemented in Cube software and in this section the different physical elements constituting the model will be explored. The naming and syntax used is derived from the Cube Software.

The model can be viewed in Cube interface opening the catalog file, always placed under the model folder, having extension .cat. The purpose with this file is to coordinate all the different parts of the model in terms of:

- Applications - files with .app extension
- Scenario keys - elements saved in .cat file
- Location of input and output data - based on program steps present in each application
- Give access to all the input and output data by scenario
- Define the rules on how to use the model (developer or user class)

The catalog file (named SamGods.cat) stores all the connections, maps and definition values for each entity listed above.

The catalog keys, listed in the catalog files, have a variety of purposes. They can define:

- The execution order for the applications
- Enable or disable the creation of specific outputs
- Point to an input file or parameter

The Cube-applications represent the programming parts, using the input data and applying the set of rules defined via the scripting language, they produce specific outputs and manage the data. (see the description of each application for more details).

The list of catalog keys defined in the model, classified by user rights and application usage are:

Catalog key name		Example of value	User	Application
Catalog name	**Interface name**			
Cube_software	Cube Software	6.1.0	Developer	Installation
EMME_software	EMME Software	3.3.4	Developer	Installation
GIS_software	ArcGis Software	10.1 Sp1	Developer	Installation
Python_Software	Python software	27	Developer	Installation
LogMod_software	Logistics Model software	(20140317)	Developer	Installation
Cube_Folder	Location of Cube Program	C:\Program Files (x86)\	Developer	Installation
Emme_folder	Location of EMME Program	C:\Program Files (x86)\	Developer	Installation
ArcGis	Location of ArcGis Program	C:/Program Files (x86)/	Developer	Installation
Python	Location of Python Program	C:\Python27\ArcGIS10.1	Developer	Installation Create the editable files Edit the data Samgods Model Handling scenario
JavaVersion	Location of Java folder:	C:\Windows\System32	Developer	Installation Samgods Model
Base_Scenario	Scenario name for the BASE scenario	Base2006	Developer	Installation Create the editable files Edit the data Samgods Model Handling scenario PWC_Matrices
Scenario_copy	Which scenario do you want to copy as new scenario (Default is Base2006)?	Base2006	All	Create the editable files
Waittime	Wait time for prompt messages during execution (default 10 minutes=600 secs):	600	Developer and Advanced User	Create the editable files Edit the data Samgods Model
read_write	Do you want to lock the scenario from further editing?	Yes	All	Edit the data
Edit_soft	Select from which interface you want to import your edits:	GIS Window (Use input files constructed by the Cube GUI)	All	Edit the data
EMME_data	Use emme network file:	C:\Temp\Emme_Net.211	All	Edit the data

Catalog key name		Example of value	User	Application
Catalog name	**Interface name**			
EMME_speed	Use emme speed table:	C:\Temp\SpeedV101_V102.DAT	All	Edit the data
Up_parameters	General parameters [Table]	C:\SamGods_v10\Scenario_Data.mdb\General_Base2006	Developer	Edit the data
GCOST	Attribute name for the extra cost on specific link[SEK]	EC_V	All	Edit the data Handling scenario
GCOST_D	Attribute name for extra cost on specific link [SEK/km]	EC_KM_V	All	Edit the data Handling scenario
DF	Daily Factor for all modes (except for Rail Mode) [days per year]	365	Developer and Advanced User	Edit the data Handling scenario
DFR	Daily Factor for Rail Mode [days per year]	250	Developer and Advanced User	Edit the data Handling scenario
PHF	Peak Hour Factor	0.12	Developer and Advanced User	Edit the data Handling scenario
Zones	Number of zones in the network (see value in scenario network)	1120	All	Edit the data Samgods Model
LogParam	General parameters for Logistic Module [Table]	C:\SamGods_v10\Scenario_Tree\Base2006\Scenario_Data.mdb\LogMod_Base2006	Developer	Edit the data Handling scenario
INTER_RATE	Interest rate used in cost calculations [$year^{-1}$]	0.1	All	Edit the data Handling scenario
STUFF	Stuffing cost [SEK per tonne]	18	All	Edit the data Handling scenario
LSTCNT	Number of transport chain solutions from the Logistics Module	5	All	Edit the data Handling scenario
DATA	Output variables for the optimal transport solution [e.g.1,2,3, max is 8]. Nbrs 6 and 7 are the useful ones with the first solution details.	6,7	All	Edit the data Handling scenario
FACTOR	Initial consolidation factor in BuildChain (1^{st} iteration)	0.75	All	Edit the data Handling scenario
CONTYPE	ALL_LORRY_TYPE_CONSOL	Yes	All	Edit the data Handling scenario
INODLO	INDIVIDUAL_OD_LEG_OPTIMIZE	Yes	All	Edit the data Handling scenario
MATD4FO	MINIMUM_ANNUAL_TONNE_DEMAND_4_FREQ_OPTIMIZE [Tonnes]	100	All	Edit the data Handling scenario
CONSOL_L	Lower bound for consolidation factor [Tonnes] (applied to all commodities). May be applied to individual commodities using consolidation table	0.05	All	Edit the data Handling scenario

Catalog key name		Example of value	User	Application
Catalog name	**Interface name**			
CONSOL_U	Upper bound for consolidation factor [Tonnes] (applied to all commodities) May be applied to individual commodities using consolidation table	0.95	All	Edit the data Handling scenario
TONNES	TONNES	DYNAMIC_AVERAGE	All	Edit the data Handling scenario
LOGCTL	Do you want to log parameters from control files?	Yes	All	Edit the data Handling scenario
LOGFLS	Do you want to log reading of input data?	Yes	All	Edit the data Handling scenario
LOGCST	Do you want to log cost values from vehicle files?	Yes	All	Edit the data Handling scenario
Del_tmp	Do you want to delete the temporary geodatabase?(Tick=yes)	Tick	All	Edit the data
Mxd_file	ArcMap GIS Project File	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Input_Data.mxd	All	Edit the data
Up_cargo	Commodity data [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Cargo_{SCENARIO_SHORTNAME}	All	Edit the data
Up_Vehicles_par	General parameters of Vehicle Classes [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Vehicles_Parameters_{SCENARIO_SHORTNAME}	All	Edit the data
Up_Vehicles_exc	Specific parameters of Vehicle Classes in some commodities [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Vehicles_exception_{SCENARIO_SHORTNAME}	All	Edit the data
Up_Tax_C	Table with tax by country (SEK/km) [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Tax_Country_{SCENARIO_SHORTNAME}	All	Edit the data
Up_Tax_L	Tax by Category (SEK) [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Tax_Category_{SCENARIO_SHORTNAME}	All	Edit the data
Up_tax_link	Tax by Link (SEK) [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Tax_Link_{SCENARIO_SHORTNAME}	All	Edit the data
Up_toll	Toll for bridges (SEK) [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Toll_Link_{SCENARIO_SHORTNAME}	All	Edit the data
Up_net	Network specific of scenario [Geodata]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Network	All	Edit the data

Catalog key name		Example of value	User	Application
Catalog name	**Interface name**			
Up_node_terminals	Port terminals (pilot fees values) [Geodata]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Ports_Swe	All	Edit the data
Up_nodes	Zones and terminals attributes [Geodata]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Nodes	All	Edit the data
Up_no_comm	Nodes commodities (interchanges allow) [Geodata]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Nodes_commodities	All	Edit the data
Up_freq	Frequency data [Geodata]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Frequency_network	All	Edit the data
Up_Rail_Capacity	Table with Rail Capacity [Table]	C:\SamGods_v10\Scenario_Tree\{SCENARIO_SHORTNAME}\Scenario_Data.mdb\Rail_Capacity_{SCENARIO_SHORTNAME}	All	Edit the data
Step_by_step	Run only one step, or run all steps	All	All	Samgods Model
Year	Year of PWC matrices	2006	All	Samgods Model PWC_Matrices
Select_cal_cos_mode	Select the mode for the LOS Calculation and the Assignment steps	All	All	Samgods Model
Select_commodity	Select commodities for the Logistics Module (for all commodities, select 0)	0	All	Samgods Model Compare Scenarios Handling scenario Change matrix format
Save_LOS	Do you want to save the LOS matrices?	Tick	All	Samgods Model
log_out	Do you want to produce BuildChainxx.log files?	No	All	Samgods Model
Cluster_p	Do you want to run the Logistics Module on several processors?	Yes	All	Samgods Model
Cluster_nodes	How many processors would you like to run simultaneously?	4	All	Samgods Model
scal_factor	Scaling factor for outputs	Thousands	All	Samgods Model
Prompt	Do you want to pause the run waiting for your action on steps LOS, LogMod,LP0,LP1,L1b?	No	All	Samgods Model
Sound	Step completion:	Email	All	Samgods Model
MediaP_Folder	Folder where the Media Player is installed	C:\Program Files (x86)\Windows Media Player\wmplayer.exe	Developer	Samgods Model
Song	Location of song to Play upon completion of task	C:\Users\Public\Music\Sample Music\Kalimba.mp3	Developer	Samgods Model
Email	Insert the email address for status on run:	You@youremail.com	All	Samgods Model

Catalog key name		Example of value	User	Application
Catalog name	**Interface name**			
UTIL_PERCENT	Cut off criteria for RCM process (will consider links with V/C > XX% where XX is the value defined)	50	D/A-user	Samgods Model
maxiters_EA	Maximum number of iterations in congested empty rail assignment	500	D/A-user	Samgods Model
SpeedFlowCurve	Delay function for Empty rail assignment (Derivative):	2*10000*(MAX(0,VOLC[LI.ID_LINK]-C))	D/A-user	Samgods Model
Save_Output_RCM	Options to save output files from RCM process	Yes, for all iterations	All	Samgods Model
RCM_Loop	Maximum number of loops for Linear Programming process (LP)	1	All	Samgods Model
Loop1b	Maximum number of loops for the Adjust Capacity Procedure	0	All	Samgods Model
ocf	Percentage of Overcapacity to be removed (over_capacity_factor):	0.5	D/A-user	Samgods Model
Lock_file	File with list of locked solutions for RCM:	LockedSTDLogMod_empty.txt	D/A-user	Samgods Model
Debug_checklist	Link number to use in debug mode CheckLinkN_itXX.dat Max 10 links (comma s, ex. 1,25,182,164)	51,82,84	D/A-user	Samgods Model
PortCalibration	Initial values for port calibration by Port Area and STAN group:	C:\SamGods_v10\05_Input_Data\Calibration\PortAreasParameters_12082014.txt	Developer	Samgods Model
StartKiel	Starting value for scaling factor on Kiel canal	1.55	Developer	Samgods Model
OverallLoopMax	Maximum number of loops in calibration process:	1	Developer	Samgods Model
ParametersPortCal	Parameters for port calibration procedure (step length, minimum value, threshold, default):	C:\SamGods_v10\05_Input_Data\Calibration\Parameters_portcalibration.dbf	Developer	Samgods Model
ParametersKiel	Parameters for Kiel calibration procedure (step length, minimum value, threshold, default):	C:\SamGods_v10\05_Input_Data\Calibration\Parameters_Kielcalibration.dbf	Developer	Samgods Model
Resume_option	Restart process from loop:	1	Developer	Samgods Model
ConsolTable	Consolidation factors table	C:\SamGods_v10\05_Input_Data\Input_Data.mdb\BuildChain_CONSOL	D/A-user	Samgods Model Handling scenario
EMAILSERVER	Email server address:	company-com.mail.eo.outlook.com	Developer	Samgods Model
PASSWORD	Email password:	**xZGlUUYEJQUGMlzUNRLfCLFmQZNeVkLaMAUL**	Developer	Samgods Model
USERNAME	Email username:	yourusername	Developer	Samgods Model
FROM	Sending account:	You@youremail.com	Developer	Samgods Model
Select_compare	What do you want to compare?	All	All	Compare Scenarios
ScenarioC_name	Scenario's name to use in comparison:	Base2006	All	Compare Scenarios
ScenarioC_dir	Scenario folder for comparison:	C:\SamGods_v10\Scenario_Tree\Base2006	All	Compare Scenarios

Catalog key name		Example of value	User	Application
Catalog name	**Interface name**			
Delete	Do you want to delete the current scenario? (mandatory if you want to delete the scenario)	Tick	All	Handling scenario
Compact	Do you want to compact the geodatabase files?	Tick	All	Handling scenario
Exp_selection	Select how you want to use the export function	3	All	Handling scenario
	(1) Do you want to export the model to a new folder using the current scenario as base?			
	(2) Do you want to export the current scenario to another model?			
	(3) Do nothing			
New_folder	Where do you want to export model(1) or scenario (2)?	C:\SamGods_v2	All	Handling scenario
New_Base	Name of the base scenario in the new model (1), (2)	Base2006	All	Handling scenario
Imp_Selection	Do you want to control the imported scenario?	No	All	Handling scenario
Format	Do you want to export in csv, DBF or GDB format?	CSV	All	Change matrix format
GDB_c_u	Do you want create a new geodatabase or use an existing one?	Existing	All	Change matrix format
GDB_folder	Define the gdb where export matrix:	C:\Samgods_V10\Scenario_Tree\{SCENARIO_SHORTNAME}\Output0_{SCENARIO_SHORTNAME}.mdb	All	Change matrix format
Type_mat	What kind of matrix do you want to export?	COST	All	Change matrix format
Type_mode	What is the mode that you want for the matrices?	ROAD	All	Change matrix format
Type_VHCL	What vehicle class do you want for the LOS matrices?	101	All	Change matrix format
Type_COST	Do you want the distance, time, domestic distance or extra cost LOS matrix?	Extracost	All	Change matrix format
Type_VHCL_1	For which vehicle (or total) do you want the VEHICLES and TONNES matrices?	202	All	Change matrix format

Table 112 Catalog keys.

The list of applications and subgroups defined in the model is shown in the following table. Numbers between parenthesis represent the execution order within a subgroup.

Application name	Subgroup level 1	Subgroup level 2	Subgroup level 3	Subgroup level 4	Subgroup level 5	File name
Installation						C:\Samgods_V10\02_Applicaions\5_Installation\Installation_st.app
Create the editable files						C:\Samgods_V10\02_Applications\1_Editing\Setup.app

Application name	Subgroup level 1	Subgroup level 2	Subgroup level 3	Subgroup level 4	Subgroup level 5	File name
	Prepare temporary data (1)					C:\Samgods_V10\02_Applications\1_Editing\EDIT_PREPAR.app
	Editable tables (2)					C:\Samgods_V10\02_Applications\1_Editing\EDIT_TABLES.app
	Editable map data (3)					C:\Samgods_V10\02_Applications\1_Editing\EDIT_MAP.app
Edit the data						C:\Samgods_V10\02_Applications\1_Editing\Edit.app
	Create the tables specific of scenario (1)					C:\Samgods_V10\02_Applications\1_Editing\CREATE_TABLES.app
		Numbering System Voyager				C:\Samgods_V10\02_Applications\1_Editing\Numbering\NUMBER01.app
		Emme Edits				C:\Samgods_V10\02_Applications\1_Editing\EMMEEDITS.app
		Compare Nodes and Node_terminals				C:\Samgods_V10\02_Applications\1_Editing\Comparison\COMPAR01.app
		Compare other tables				C:\Samgods_V10\02_Applications\1_Editing\EDIT_OTHERS.app
		Rail Capacity Checks				C:\Samgods_V10\02_Applications\1_Editing\Capacity\RAILCA01.app
		Frequency matrices				C:\Samgods_V10\02_Applications\1_Editing\Frequency.app
	Save changes in the main gdb (2)					C:\Samgods_V10\02_Applications\1_Editing\CREATE_SAVE.app
		Regions for links				C:\Samgods_V10\02_Applications\1_Editing\Regions\REGION01.app
Samgods Model						C:\Samgods_V10\02_Applications\2_Run\SamGod_VY.app
	Initialization (1)					C:\Samgods_V10\02_Applications\2_Run\Ini\INITIA00.app
	LOS calculation (4)					C:\Samgods_V10\02_Applications\2_Run\Costs_VY\COST_VY.app
		Data Preparation				C:\Samgods_V10\02_Applications\2_Run\Costs_VY\DATAPREPARATION_VY.app
(Samgods Model)			Tax calculation			C:\Samgods_V10\02_Applications\2_Run\Costs_VY\DATAPREPARATION_VY_01.app
		Road				C:\Samgods_V10\02_Applications\2_Run\Costs_VY\ROAD.app
		Rail				C:\Samgods_V10\02_Applications\2_Run\Costs_VY\RAIL.app
		Sea				C:\Samgods_V10\02_Applications\2_Run\Costs_VY\SEA.app
		Air				C:\Samgods_V10\02_Applications\2_Run\Costs_VY\AIR.app
		Conversion from				C:\Samgods_V10\02_Applications\2_Run\Costs_VY\CONVERALL.app

Application name	Subgroup level 1	Subgroup level 2	Subgroup level 3	Subgroup level 4	Subgroup level 5	File name
		VY to LogMod				
			From VY to LogMod Road			C:\Samgods_V10\02_Applications\2_Run\Costs_VY\CONCOSTVYE2.app
			From VY to LogMod Rail			C:\Samgods_V10\02_Applications\2_Run\Costs_VY\CONRAILVYE2.app
			From VY to LogMod Sea			C:\Samgods_V10\02_Applications\2_Run\Costs_VY\CONSEAVYE2.app
			From VY to LogMod Air			C:\Samgods_V10\02_Applications\2_Run\Costs_VY\CONAIRVYE2.app
	Logistics Module (5)					C:\Samgods_V10\02_Applications\2_Run\LogisticModVY\LOGMOD.app
		Prepare data				C:\Samgods_V10\02_Applications\2_Run\LogisticModVY\PREPARE.app
		Prepare Data second part				C:\Samgods_V10\02_Applications\2_Run\LogisticModVY\PREPARE_01.app
		Prepare Data third part				C:\Samgods_V10\02_Applications\2_Run\LogisticModVY\PREPARE_02.app
		Port Area LOS inputs (STAN)				C:\Samgods_V10\02_Applications\2_Run\LogisticModVY\PORTAREAS.app
		Run the logistic model				C:\Samgods_V10\02_Applications\2_Run\LogisticModVY\RUNTMODEL.app
		Save Reports				C:\Samgods_V10\02_Applications\2_Run\LogisticModVY\SAVEREPORTS.app
	Assignment (6)					C:\Samgods_V10\02_Applications\2_Run\Ass_VY\ASS_VY.app
		Conversion from LogMod to VY				C:\Samgods_V10\02_Applications\2_Run\Ass_VY\CONVERSION OF EMME MATRICES.app
		Road Assignment				C:\Samgods_V10\02_Applications\2_Run\Ass_VY\ASS_ROAD_VY.app
		Rail Assignment				C:\Samgods_V10\02_Applications\2_Run\Ass_VY\ASS_RAIL_VY.app
		Sea Assignment				C:\Samgods_V10\02_Applications\2_Run\Ass_VY\ASS_SEA_VY.app
		Air Assignment				C:\Samgods_V10\02_Applications\2_Run\Ass_VY\ASS_AIR_VY.app
(Samgods Model)	Results (7)					C:\Samgods_V10\02_Applications\2_Run\Results\RESULTS.app
		Results 1.0				C:\Samgods_V10\02_Applications\2_Run\Results\RESULTS_NEW.app
			Port Areas			C:\Samgods_V10\02_Applications\2_Run\Results\PORTAR01.app
			Oresund Kiel and Jylland			C:\Samgods_V10\02_Applications\2_Run\Results\ORESUN01.app
			Reports per geographical aggregation			C:\Samgods_V10\02_Applications\2_Run\Results\REPORT01.app
	Rail Capacity					C:\Samgods_V10\02_Applications\9_RCM1\RAILCA01.app

Application name	Subgroup level 1	Subgroup level 2	Subgroup level 3	Subgroup level 4	Subgroup level 5	File name
	Management (8)					
		Data Preparation LP				C:\Samgods_V10\02_Applications\9_RCM1\Dataprep_LP\DATAPR01.app
		Warm start				C:\Samgods_V10\02_Applications\9_RCM1\WARMST01.app
		LP Loop				C:\Samgods_V10\02_Applications\9_RCM1\LP\LPLOOP01.app
			LP0 step			C:\Samgods_V10\02_Applications\9_RCM1\LP\LP0\LP0STEP.app
				Prepare CTL files – Extract procedure		C:\Samgods_V10\02_Applications\9_RCM1\LP\LP0\PREPARE_EXTRACT_CTL.app
			LP1+ step			C:\Samgods_V10\02_Applications\9_RCM1\LP\LP1\LP1ST01.app
				Prepare CTL files		C:\Samgods_V10\02_Applications\9_RCM1\LP\LP1\PREPARE_02.app
				Run BuildChainRCM and ChainChoiRCM		C:\Samgods_V10\02_Applications\9_RCM1\LP\LP1\RUNBUILDCHAINRCM.app
				Run MPS LP Extract		C:\Samgods_V10\02_Applications\9_RCM1\LP\LP1\RUNMPS_LP_EXTRACT.app
				Check convergence		C:\Samgods_V10\02_Applications\9_RCM1\LP\LP1\CHECKC01.app
		LP1 Adjust Capacities				C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\LP1ADJ01.app
			Data Preparation			C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\DATAPR01.app
				Conversion from LogMod to VY		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\CONVERSION_EMME_MAT.app
				Empty Assignment		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\EMPTYASSIGNMENT1B1.app
				Setup Capacity Values		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\SETUPC01.app
(Samgods Model)			LP Adjust Capacity Loop			C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\LPADJUL01.app
				LP1b step		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\LP1B.app
					Prepare CTL files – Extract procedure	C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\PREPARE_EXTRACT_CTL.app
				Conversion from LogMod to VY		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\CONVERSION_EMME_MATLOOP.app
				Rail Assignment		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\EMPTYASSIGNMENT.app
				Setup Capacity Values		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\SETUPCAPACITIES.app

Application name	Subgroup level 1	Subgroup level 2	Subgroup level 3	Subgroup level 4	Subgroup level 5	File name
				Convergence Criteria		C:\Samgods_V10\02_Applications\9_RCM1\LPAdjCap\CONVCRITERIA.app
		Run Final Process				C:\Samgods_V10\02_Applications\9_RCM1\Final\RUNFIN01.app
			Prepare control files			C:\Samgods_V10\02_Applications\9_RCM1\Final\PREPARE_02.app
			Run the final process			C:\Samgods_V10\02_Applications\9_RCM1\Final\RUNTMODEL.app
			Save Reports			C:\Samgods_V10\02_Applications\9_RCM1\Final\SAVEREPORTS.app
	RCM Assignment (9)					C:\Samgods_V10\02_Applications\9_RCM1\Ass_VY\ASS_VY.app
		Conversion from LogMod to VY				C:\Samgods_V10\02_Applications\9_RCM1\Ass_VY\CONVERSION OF EMME MATRICES.app
		Road Assignment RCM				C:\Samgods_V10\02_Applications\9_RCM1\Ass_VY\ASS_ROAD_VY.app
		Rail Assignment RCM				C:\Samgods_V10\02_Applications\9_RCM1\Ass_VY\ASS_RAIL_VY.app
		Sea Assignment RCM				C:\Samgods_V10\02_Applications\9_RCM1\Ass_VY\ASS_SEA_VY.app
		Air Assignment RCM				C:\Samgods_V10\02_Applications\9_RCM1\Ass_VY\ASS_AIR_VY.app
	Results RCM (10)					C:\Samgods_V10\02_Applications\9_RCM1\Results\RESULTS.app
		Reports 1.0				C:\Samgods_V10\02_Applications\9_RCM1\Results\REPORTNEW.app
			PORT AREAS			C:\Samgods_V10\02_Applications\9_RCM1\Results\PORTAR01.app
			Oresund Kiel and Jylland			C:\Samgods_V10\02_Applications\9_RCM1\Results\ORESUN01.app
(Samgods Model)			Rail Capacity			C:\Samgods_V10\02_Applications\9_RCM1\Results\RAILCA01.app
			Reports per geographical aggregation			C:\Samgods_V10\02_Applications\9_RCM1\Results\REPORT01.app
	Parameters calculation (11)					C:\Samgods_V10\02_Applications\2_Run\Calibr\PARAMETERS.app
Compare Scenarios						C:\Samgods_V10\02_Applications\3_Compare\Compare.app
	LOS Matrices (1)					C:\Samgods_V10\02_Applications\3_Compare\LOSMAT.app
	OD Matrices (2)					C:\Samgods_V10\02_Applications\3_Compare\OD_matrices\ODMATR01.app
	Assignment (3)					C:\Samgods_V10\02_Applications\3_Compare\ASSIGNMENT.app
Handling scenario						C:\Samgods_V10\02_Applications\4_Handling\Handling.app
	Delete (1)					C:\Samgods_V10\02_Applications\4_Handling\DELETE.app
	Compact (2)					C:\Samgods_V10\02_Applications\4_Handling\COMPACT.app

Application name	Subgroup level 1	Subgroup level 2	Subgroup level 3	Subgroup level 4	Subgroup level 5	File name
	Scenario Export (3)					C:\Samgods_V10\02_Applications\4_Handling\SCENARIOEXPORT.app
		Model Export				C:\Samgods_V10\02_Applications\4_Handling\MODELEXPORT.app
		General tables				C:\Samgods_V10\02_Applications\4_Handling\GENERAL_TABLES.app
		Create the new base				C:\Samgods_V10\02_Applications\4_Handling\CREATE_NEW_BSE.app
		Create the new scenario specific tables				C:\Samgods_V10\02_Applications\4_Handling\CREATE_SS_TABLE_T.app
			Create the scenario tables			C:\Samgods_V10\02_Applications\4_Handling\CREATE_SS_TABLE.app
			Create the scenario specific tables			C:\Samgods_V10\02_Applications\4_Handling\CREATE_SS_TABLE_2.app
	Scenario Import (4)					C:\Samgods_V10\02_Applications\4_Handling\SCENARIOIMPORT.app
PWC_Matrices						C:\Samgods_V10\02_Applications\6_PWC_matrices\PWC_MA01.app
Change matrix format						C:\Samgods_V10\02_Applications\7_Matrix_form\Change01.app

Table 113 Applications and subgroups.

The application structure

"Installation" application

Input data:

- Python scripts under c:\SamGods\07_Python\Original:
    - Compact_GDB.py;
    - Copy_Feature.py;
    - Copy_Table.py;
    - Create_shape_file_terminals.py.

Output data:

- Same file names with a different final folder.
    - folder c:\SamGods\07_Python.
- Create the "Model_description" table in the Input_data.mdb and empty.mdb (under SamGods\02_Applications\Edits) files. The first one is the general geodatabase, the second one is the template used for all the output geodatabases.

Purpose

Change the map for the files used in the model structure.

Store the general definitions of the model for long-term storage purposes.

"Create the editable files" application

Input data: Scenario specific tables in the main geodatabase and base tables.

Output data: Scenario specific tables in the main geodabase and scenario data in the temporary geodatabase under scenario folder with name Scenario_data.mdb. Txt files under the work folder SamGods\02_Applications\1_Editing to manage different checks and time data.

Purpose:

Join the scenario tables and the base tables to create the editable data for the current scenario.

Application	Description	Programs
Prepare temporary data (1)	**Check if the scenario already exist**	1 to 7
	- create the check.txt file (check if general_parameters table already exist in the main geodatabase)	
	- create the date_time1.txt (last modification of Scenario_data.mdb)	
	- create the date_time.txt (current time)	
	- create the report1.txt (verify if the temporary geodatabase already exist)	
	First case: The scenario does not exist	
	- jump step 4 because the temporary geodatabase not exist	
	- copy all the scenario specific table from the scenario selected by Scenario_copy catalog key	

Application	Description	Programs
	Second case: The scenario already exists but without temporary geodatabase	
	- jump step 4 because the temporary geodatabase not exist	
	Third case: The scenario already exist and exist the temporary geodatabase	
	First sub case: the scenario is lock - no actions are required. Process goes to end	
	Second sub case: the scenario is still editable - go to next step and check if any edit was made before this run in the temporary geodatabase. If yes goes to next step. If no goes to end.	
	Copy the scenario specific table in the temporary geodatabase Scenario_data.mdb	-
	General_{SCENARIO_SHORTNAME}	8
	SC_{SCENARIO_SHORTNAME}_Cargo	9
	SC_{SCENARIO_SHORTNAME}_Vehicles_Parameters	10
	SC_{SCENARIO_SHORTNAME}_Link	11
	SC_{SCENARIO_SHORTNAME}_Node	11
	SC_{SCENARIO_SHORTNAME}_Node_Terminals	12
	SC_{SCENARIO_SHORTNAME}_Nodes	13
	SC_{SCENARIO_SHORTNAME}_Nodes_commodities	14
	SC_{SCENARIO_SHORTNAME}_Frequency_data_link	15
	SC_{SCENARIO_SHORTNAME}_Frequency_data_node	16
	SC_{SCENARIO_SHORTNAME}_Tax_Country	17
	SC_{SCENARIO_SHORTNAME}_Tax_Category	18
	SC_{SCENARIO_SHORTNAME}_Tax_Link	19
	SC_{SCENARIO_SHORTNAME}_Toll_Link	20
	SC_{SCENARIO_SHORTNAME}_PropLink	21
	LogMod_{SCENARIO_SHORTNAME}	22
	SC_{SCENARIO_SHORTNAME}_Rail_Capacity	23
	SC_{SCENARIO_SHORTNAME}_Vehicles_exc	24
Editable tables (2)	**Merge the scenario tables with the base tables in the temporary geodatabase**	
	SC_{SCENARIO_SHORTNAME}_Cargo plus Cargo_Base2006	1
	SC_{SCENARIO_SHORTNAME}_Vehicles_Parameters plus Vehicles_parameters_Base2006	2
	SC_{SCENARIO_SHORTNAME}_Tax_Country plus Tax_Country_Base2006	3
	SC_{SCENARIO_SHORTNAME}_Tax_Category plus Tax_Category_Base2006	4
	SC_{SCENARIO_SHORTNAME}_Tax_Link plus Tax_Link_Base2006	5
	SC_{SCENARIO_SHORTNAME}_Toll_Link plus Toll_Link_Base2006	6
	SC_{SCENARIO_SHORTNAME}_Rail_Capacity plus Rail_Capacity_Base2006	7
	SC_{SCENARIO_SHORTNAME}_Vehicles_exc plus Vehicles_exception_Base2006	8
Editable map data (3)	**Merge the scenario tables with the base tables in the temporary geodatabase creating networks and feature classes to use in GIS window**	-
	Sc_{SCENARIO_SHORTNAME}_Link and Sc_{SCENARIO_SHORTNAME}_Node plus Base2006_Link and Base2006_Node	-
	1. prepare the data to add general information from lookup tables (modes table)	1 to 4
	2. save in network format the tables created in the previous steps	5
	SC_ {SCENARIO_SHORTNAME}_node_Terminals plus Node_terminalsBase2006	6 to 7
	Sc_{SCENARIO_SHORTNAME}_Nodes	8 to 9
	Sc_{SCENARIO_SHORTNAME}_Nodes_commodities	10 to 11
	Sc_{SCENARIO_SHORTNAME}_Frequency_Data_Link and Sc_{SCENARIO_SHORTNAME}_Frequency_Data_Node plus Base:2006_Frequency_Data Network	12 to 13
	Copy Samgods feature class in temporary geodatabase and end the process	14

"Edit the data" application

Input data: Scenario tables in the temporary geodatabase Scenario_Data.mdb under scenario folder.

Output data: Scenario specific tables in the main geodabase. Txt files under the work folder SamGods\02_Applications\1_Editing to manage time data.

Purpose:

Compare the editable tables, networks and feature classes in the temporary geodatabase with the base data in the main geodatabase Input_data.mdb, after comparison save the updated scenario tables and general_ {SCENARIO_SHORTNAME} table.

Application		Description	Programs	Subprograms
Create the tables specific of scenario (1)		Check if the scenario is in read or write/read mode	1	
		if read mode is forbidden save the scenario specific tables in main geodatabase all the processes will be skipped	3	
		if read/write mode then go to next step		
		Delete the previous report file REPORT_ED_BS06.txt	2	
		Compare the scenario data in the temporary geodatabase Scenario_data.mdb with the base data in the main geodatabase to create the scenario specific tables	4	
		Create SC_{SCENARIO_SHORTNAME}_Cargo in temporary geodatabase	4	
		Create SC_{SCENARIO_SHORTNAME}_Vehicles_parameters in temporary geodatabase	5	
		Check if edits come from the Voyager or the Emme network	6	
	Numbering system	In case of edits from Voyager, check all the codes to create the correct original number for the nodes:	7	
		Check the number of zones, ID_REGION, ID_COUNTRY and MODE_N codes	7	1,2
		Based on the previous codes defined set the SCBSTANN code	7	3
		Code the NORIG value for the zones	7	4,5,6
		Code the NORIG value for the nodes	7	7,8
		Delete the information related to modes joined in the "Create editable files"	7	9,10
		Merge all the new ORIG codes created	7	11
		Skim network for subsequent steps	7	12
		Dump the base scenario network	7	13,16
		Dump the current scenario network	7	14
		Convert network from EMME to Voyager format	7	15
		Compare node and link tables of the base scenario with the current scenario	7	17
		Check if edits come from the Voyager or the Emme network	8	
	Emme Edits	In case of edits from EMME, check all the codes to create the correct original number for the nodes:	9	
		Check if EMME network is defined in 211 format	9	1
		Create link and node tables from EMME network	9	2
		Check the number of zones, ID_REGION, ID_COUNTRY and MODE_N codes	9	3
		Based on the previous codes defined set the SCBSTANN code	9	4
		Check the number of zones	9	5
		Code the NORIG value for the zones	9	6
		Code the NORIG value for the nodes	9	7,8
		Node table union	9	9
		Recode the link table	9	10
		Recode the extra attribute	9	11
		Merge link and node data in the current scenario network	9	12
		Dump the current scenario network	9	13
		Dump the base scenario network in dbf format for subsequent steps	9	14
		Compare node and link tables of base scenario with current scenario	9	15
		Skim network for subsequent steps	9	16

Application		Description	Programs	Subprograms
		Coming from edits from Voyager or edits from EMME, the process continues from this point	10	
	Compare Nodes and Node_terminals	Compare current scenario with base scenario and scenario specific tables for nodes and node terminals	11	
		Dump Base scenario Node terminals	11	1
		Dump current scenario Node terminals	11	2
		Compare node terminals tables of current scenario with base scenario	11	3
		Create SC_{SCENARIO_SHORTNAME}_Node_Terminals		4
		Dump Base scenario Nodes	11	5
		Dump current scenario Nodes	11	6
		Compare nodes tables of current scenario with base scenario	11	7
		Create SC_{SCENARIO_SHORTNAME}_Nodes	11	8
	Compare other tables	Compare current scenario with base scenario and create the following scenario specific tables:	12	
		SC_{SCENARIO_SHORTNAME}_Nodes_Commodities	12	1
		SC_{SCENARIO_SHORTNAME}_Tax_Country	12	2
		SC_{SCENARIO_SHORTNAME}_Tax_Category	12	3
		SC_{SCENARIO_SHORTNAME}_Tax_Link	12	4
		SC_{SCENARIO_SHORTNAME}_Toll_Link	12	5
		SC_{SCENARIO_SHORTNAME}_Vehicles_exc	12	6
	Rail Capacity Checks	Check rail management data and create a scenario specific table for rail capacity	13	
		Check consistency with network and ID_LINK numbers	13	1,2,3
		Dump Rail capacity table for base scenario	13	4
		Dump Rail capacity table for current scenario	13	5
		Create SC_{SCENARIO_SHORTNAME}_Rail_Capacity	13	6
		Verify the maximum zone number between tables	14 to 19	
		If the network does not have any new zones, create the Sc_{SCENARIO_SHORTNAME}_Frequency_Data_Link and SC_{SCENARIO_SHORTNAME}_Freqeuncy_Data_Node tables in the temporary geodatabase	20 to 24	
	Frequency matrices	If new terminals are added to the network it will create a list of frequency values based on the default frequency table	25	
		Create the LOS matrices to understand the connections between terminals	25	1,2
		Set the default values for the frequency matrices based on the A_default_frequency table	25	3
		Filter the default values only for the new zones	25	4,5,6
		Convert the frequency matrices to network format and merge with the scenario frequency network to preserve previous edits	25	7
		Dump the network	25	8
		Create the Sc_{SCENARIO_SHORTNAME}_Frequency_Data_Link and SC_{SCENARIO_SHORTNAME}_Freqeuncy_Data_Node tables in the temporary geodatabase	26	
Save changes in the main gdb (2)	Regions for links	Delete check.txt	1	1
		Check the record number	1	2,3
		Dump network	1	4
		Classify cases	1	5
		If you need the geoprocessing toll to classify links	1	6
		Create temporary network considering classified cases	1	7
		Call python for link classification	1	8
		Check if you need weight calculation	1	9,10
		Weight calculation	1	11,12

The application structure

Application		Description	Programs	Subprograms
		Merge all tables and create SC_{SCENARIO_SHORTNAME}_PropLink	1	13,14,15,16
		Save the scenario specific tables from the temporary geodatabase to the main geodatabase:	2 to 16	
		SC_{SCENARIO_SHORTNAME}_Cargo	2	
		SC_{SCENARIO_SHORTNAME}_Vehicles_Parameters	3	
		Copy SC_{SCENARIO_SHORTNAME}_Link and SC_{SCENARIO_SHORTNAME}_Node tables via python script	4	
		SC_{SCENARIO_SHORTNAME}_Node_Terminals	5	
		SC_{SCENARIO_SHORTNAME}_nodes	6	
		SC_{SCENARIO_SHORTNAME}_nodes_commodities	7	
		SC_{SCENARIO_SHORTNAME}_Frequency_Data_Link	8	
		SC_{SCENARIO_SHORTNAME}_Frequency_Data_Node	9	
		SC_{SCENARIO_SHORTNAME}_Tax_Country	10	
		SC_{SCENARIO_SHORTNAME}_Tax_Category	11	
		SC_{SCENARIO_SHORTNAME}_Tax_Link	12	
		SC_{SCENARIO_SHORTNAME}_Toll_Link	13	
		SC_{SCENARIO_SHORTNAME}_PropLink	14	
		SC_{SCENARIO_SHORTNAME}_Rail_Capacity	15	
		SC_{SCENARIO_SHORTNAME}_Vehicles_exc	16	
		Delete temporary txt file containing time and date stamps	17	
		Update information of time and date on temporary txt	18	
		Update the state of the scenario (write to only read) and ZONES_M	19	
		Update the General_Base table:		
		Save new value in the temporary geodatabase Scenario_Data.mdb	20	
		Save new value in the main geodatabase Input_Data.mdb	21	
		Update the LogMod_Base table:		
		Save new value in the temporary geodatabase Scenario_Data.mdb and in the main geodatabase Input_Data.mdb	22,23	
		Delete all the temporary files created along the run	24	

"SamGods" application

Input data: Scenario tables and base table in main geodatabase Input_data.mdb.

Output data: Refer to Output section 0.

Purpose:

Provide all the input data required by the Logistics and the Rail Capacity Management Modules and derive a set of standard outputs alongside the results under Logmod folder produced by the external programs.

The description is split into two tables, the first one mainly related to the Standard Logistics Module, the second one to the Rail Capacity Management Module.

Application			Description	Programs	Subprograms	
Initialization			Check the resume option and identified if new run or continuation from previous cycle	1		
			Set the scaling factor for Kiel Canal depending on cycle number	2		
Loop #2			Copy the input parameters for the current loop			
Pilot #3			Control the resume option of cycling process and warm start procedure within a fixed loop			
Transport Costs (4)	Data Preparation (1)		Check the choices made by user, if not right start a prompt. Create the outputX.mdb if not exist and if required.	1	1	
			Load the parameters from general_{SCENARIO_SHORTNAME}table into the system variables	1	2	
			Verify if few outputs already exist and if they have the same input data	1	3	
			Check if the scenario is locked or still editable	1	4	
			Create the Vehicle_parameter table for the scenario	1	5	
			Create the Cargo table for the scenario	1	6	
			Create the network for the scenario	1	7	
			Dump in dbf format the link and node tables	1	8	
		Tax calculation	Set the node class (see table A_nodeclass) value for all the nodes	1	9	1

The application structure

Application			Description	Programs	Subprograms	
			Classify the links based on the class node and the ID_COUNTRY in the A_Categories code - identify the country id for each link	1	9	2
			Convert all the scenario tables for taxes and tolls to a link format	1	9	3 to 12
			Create the network with all the extra attributes with undefined values	1	9	13
			Recode the mode for each vehicle class	1	9	15
			Update with scenario values all the extra attributes	1	9	14
			Add the fields for MODE	1	9	16
			Create the equations to save the final value for tax and extracost	1	9	17
			Save the final values (TAX_TOT,EXCTTOT and TOLL) for each link and by vehicle class	1	9	18
			Create the intrazonal matrix in dbf format	1	10	
			Change the format of intrazonal matrix from dbf to MAT	1	11	
			Save the node_labels table in Output0.mdb	1	12	
			Create the port area table for the scenario	1	13	
			Create the {SCENARIO_DIR}\LogMod\Input\PortAreas_def.txt file for Rescaling procedure	1	14	
			Control the execution order and skip some steps not required in cycling process	1	15	
			Create the Capacity_table for current scenario	1	16	
			Merge the network from step 9 and the Capacity table to use in next assignments	1	17	
			Control the execution order on cycling process and options to run step by step the model	1	18	
			Select which mode run based on Select_cal_cos_mode catalog key	2		
			Execution order, start the road mode	3	1	
			Delete {SCENARIO_DIR}\LogMod\RCM\PathTreeRoad.txt if exist	3	2	
			Loop between vehicle classes (from 101 to 105)	3	3	
			Set the values for VOT and KMCOST reading the Vehicle_parameters table	3	4	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	3	5	
			Define the speed values and free flow times	3	6	
			Skim the los matrices	3	7	
			Merge the 5 LOS matrices in one file	3	8	
			Updated the outputs table with new output	3	9 to 10	
	Road (3)		Skip all the other steps if Select_cal_cos_mode has "Road" value	3	11	
			Execution order, start the rail mode	4	1	
			Delete {SCENARIO_DIR}\LogMod\RCM\PathTreeRail.txt if exist	4	2	
			Loop between vehicle classes (from 201 to 209)	4	3	
	Rail (4)		Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	4	4	

Application			Description	Programs	Subprograms	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	4	5	
			Skim the los matrices	4	6	
			For all the vehicle classes except 202 there is a unique mode, so jump the subsequent steps. For 202 go to next step	4	7	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	4	8	
			Skim the los matrices	4	9	
			Label to skip the steps 7 and 8 if vehicle class different from 202	4	10	
			Merge the 9 LOS matrices in one file	4	11	
			Updated the outputs table with new output	4	12,13	
			Skip all the other steps if Select_cal_cos_mode has "Rail" value	4	14	
			Execution order, start the sea mode	5	1	
			Delete {SCENARIO_DIR}\LogMod\RCM\PathTreeSea.txt if exist	5	2	
			Read Port Area table and set up transoceanic factors in MAT format	5	3	
			Loop between vehicle classes (from 301 to 321)	5	4	
			Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	5	5	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	5	6	
			Skim the los matrices	5	7	
			Merge the 21 LOS matrices in one file	5	8 to 10	
			Updated the outputs table with new output	5	11 to 12	
	Sea (5)		Skip all the other steps if Select_cal_cos_mode has "Sea" value	5	13	
			Execution order, start the air mode	6	1	
			Delete {SCENARIO_DIR}\LogMod\RCM\PathTreeAir.txt if exist	6	2	
			Set the value of vehicle type	6	3	
			Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	6	4	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	6	5	
			Skim the los matrices	6	6	
			Updated the outputs table with new output	6	7 to 8	
	Air (6)		Skip all the other steps if Select_cal_cos_mode has "Air" value	6	9	
			Select which mode run based on Select_cal_cos_mode catalog key	7	1	
			Mode Road	7	2	1
			Loop between vehicle classes (from 1 to 5)	7	2	2
	Conversion from VY to LogMod (7)		Read the vehicle class number to rename properly the output files	7	2	3
		From VY to LogMod Road	Crate the dbf tables from the matrix file adding the NORIG code	7	2	4

The application structure

Application			Description	Programs	Subprograms	
			Sort the records and create the V10x_dist.314,V10x_timeh.314, V10xxkr.314 and V10x_ddist.314 under LogMod\Input\LOS folder	7	2	5
			Skip all the other steps if Select_cal_cos_mode has "Road" value	7	2	6
			Mode Rail	7	3	1
			Loop between vehicle classes (from 6 to 13)	7	3	2
			Read the vehicle class number to rename properly the output files	7	3	3
			Crate the dbf tables from the matrix file adding the NORIG code	7	3	4
			Sort the records and create the V20x_dist.314,V20x_timeh.314, V20x_xkr.314 and V20x_ddist.314 under LogMod\Input\LOS folder	7	3	5
		From VY to LogMod Rail	Skip all the other steps if Select_cal_cos_mode has "Rail" value	7	3	6
			Mode Sea	7	4	1
			Loop between vehicle classes (from 14 to 34)	7	4	2
			Read the vehicle class number to rename properly the output files	7	4	3
			Crate the dbf tables from the matrix file adding the NORIG code	7	4	4
			Sort the records and create the V30x_dist.314,V30x_timeh.314, V30x_xkr.314 and V30x_ddist.314 under LogMod\Input\LOS folder	7	4	5
		From VY to LogMod Sea	Skip all the other steps if Select_cal_cos_mode has "Sea" value	7	4	6
			Mode air	7	5	1
			Crate the dbf tables from the matrix file adding the NORIG code			2
			Sort the records and create the V401_dist.314,V401_timeh.314, V401_xkr.314 and V401_ddist.314 under LogMod\Input\LOS folder	7	5	3
		From VY to LogMod Air	Skip all the other steps if Select_cal_cos_mode has "Air" value	7	5	4
			Label to skip properly the unselected modes	8		
Logistic Model (5)	Prepare data (1)		Select if run the logistic model, selection based on Step_by_Step catalog key	1	1	
			Write in txt format the Cargo_Base2006 table under LogMod\Input\Cost with name Cargo.txt	1	2	
			Merge the scenario specific table and the base table for Node_terminals_Base2006	1	3	
			Write in txt format the Node_terminals_{SCENARIO_SHORTNAME} table under LogMod\Input\Cost with name Pilotfees.txt	1	4	
			Merge the scenario specific table and the base table for Vehicles_exception_{SCENARIO_SHORTNAME}	1	5	

Application			Description	Programs	Subprograms	
			Compare the Vehicle Parameters table and Vehicle_exception table to get the final values	1	6	
			Write in txt format the Vehicle_parameters table under LogMod\Input\Cost with name VHCLS_DRY_BULK.TXT, VHCLS_GEN_CARGO.TXT, VHCLS_LIQ_BULK.TXT, VHCLS_GEN_CARGOSTEEL.TXT, VHCLS_DRY_BULK_IRON.TXT and VHCLS_DRY_BULK_S10_11.TXT	1	7	
			Write in txt format the Vehicle_parameters table under LogMod\Input\General with name averagevehiclecapacity.txt	1	8	
			Move in dbf format for the scenario specific tables SC_{SCENARIO_SHORTNAME}_Frequency_data_Link and SC_{SCENARIO_SHORTNAME}_Frequency_Data_node	1	9	
			Merge the dbf table with the base frequency network	1	10	
			Change the format from network to matrix	1	11	
			Write in txt format under LOGMOD\INPUT\LOS\ the 10 files for frequency (.314)	1	12	
	Prepare data second part (2)		Merge the scenario specific table and the base table for Nodes_{SCENARIO_SHORTNAME}	2	1	
			Write in txt format the Nodes_{SCENARIO_SHORTNAME}table under LogMod\Input\Nodes with name Nodes.txt	2	2	
			Merge the scenario specific table and the base table for Nodes_commodities_{SCENARIO_SHORTNAME}	2	3	
			Write in txt format the Nodes_Commodities_{SCENARIO_SHORTNAME}table under LogMod\Input\Nodes into 17 txt files	2	4 to 5	
			Copy files from LogMod\INPUT\Nodes to LogMod\INPUT\Nodes_CC	2	6	
			Create Nodes.txt file with Dwell capacity for Värnern lake	2	7	
			Create {SCENARIO_DIR}\LogMod\Extract\emptyfrac.dat for EXTRACT procedure	2	8	
	Prepare data third part (3)		Change the format for BuildChain_MODES table for subsequent steps	3	1	
			Prepare BuildChain_Common.ctl, BuildChain_STAN2.ctl, BuildChain_STAN8.ctl and BuildChain_STAN9.ctl control files	3	2	
			Prepare ChainChain_Common.ctl, ChainChain_STAN2.ctl, ChainChain_STAN8.ctl and ChainChain_STAN9.ctl control files	3	3	

The application structure

Application			Description	Programs	Subprograms	
			Setup loop count to control list of commodities (if a specific commodity or STAN group is selected)	3	4	
			Loop from commodity 1 to commodity 35 to create the control files for BuildChain.exe, ChainChoi.exe, BuildChain4RCM.exe, ChainChoi4RCM.exe and LP2CC.exe programs	3	5	
			Create control files for BuilChain under LogMod\BUILDCHAIN	3	6	
			Create control files for ChainChoi under LogMod\CHAINCHOI	3	7	
			Skip other commodities if not selected All (0)	3	8	
			Convert the Chain_list table in txt format under LOGMOD\INPUT\CHAINTYPE.LIS	3	9	
			Loop from vehicle class 101 to vehicle class 401 to create the control files for extract.exe program	3	10	
			Read the Vehicle_parameters table to identify the vehicle class	3	11	
			Create control files for Extract program under LogMod\Extract	3	12	
			Create control files for Mergerep program under LogMod\MERGEREP\	3	13	
			Copy empty vehicle matrices under LogMod\Extract to prevent failure	3	14	
	Port Area LOS inputs (STAN) (4)		Run from here when cycling process (since changing only LOS matrices for sea)	4	1	
			Copy table with scaling factors for current cycle in LogMod\Input\CalibrationParameters.txt	4	2	
			Setup control file for MPS.jar program to rescale LOS matrices	4	3	
			Setup batch file for LogMod\RCM\Rescale_LOS.BAT to run MPS.jar program	4	4	
			Run Rescale_LOS.BAT file and check if finished properly	4	5	
	Run the logistic model (5)		Print out the batch file to call logistic model selecting the commodity based on the value in Select_commodity catalog key	5	1	
			Print out the batch file for call alternatively the BuildChain and ChainChoi programs	5	2	

Application			Description	Programs	Subprograms	
			Create the batch file for the extract procedure	5	3	
			Change the execution order if selected the normal logistic model or advanced. Call the batch files RUNALL.BAT and EXTRACTALL.BAT if normal procedure has been chosen	5	4	
			Label to jump the normal logistic model and run advanced	5	5	
			Setup the batch file Logmod\EXTRACTALL.BAT	5	6	
			Setup the batch file Logmod\EXTRACT.BAT	5	7	
			Set up files to describe then number of processors and batch file for call the Samgods_Parallelization_Module.jar	5	8	
			Run RUNALL.BAT and EXTRACTALL.BAT using Parallelization Module	5	9	
			Create the batch file for merge.exe	5	10	
			Call the external program "Merge reports"	5	11	
	Save Reports (6)		Save the report file from ChainChoi.exe (.rep)	6	1	
			Change the format of report file into Voyager	6	2 to 3	
			Merge domestic distances from LOS files for subsequent step	6	4	
			Delete temporary files used to accumulate results from all commodities	6	5 to 9	
			Loop over commodities (ID_LOOPC)	6	10	
			Filter node table and produce table with only zones	6	11	
			Control the commodities that must be used (if all, STAN group or single commodity)	6	12	
			Delete headers from ChainChoi\OUTPUT\ChainChoiXX01.out files	6	13	
			Calculate sum of variables for reports 5,6,7,8	6	14	
			Merge ChainChoiXX.rep files (input for reports 10 and 11)	6	15	
			Merge VhclRepXXrep files (input for reports 12)	6	16	
			Calculate report 5 and 6	6	17	
			Calculate report 7 and 8		18	
			Calculate report 10 and 11		19	
			Calculate report 12		20	
			Rename LogMod\LOG\logxx.log into logxxSTD.log		21	
			Updated the outputs table with new output	6	22 to 23	
			Label to skip the report phase	6	25	
Assignment (6)	Conversion from LogMod to VY (1)		Loop for all the commodities (from 1 to 35)	1	1	
			Load the vehicle number from Vehicle_parameters table	1	2	
			Covert the 314 format in matrix format	1	3 to 5	

The application structure

Application			Description	Programs	Subprograms	
			Merge by mode the matrices	1	6 to 11	
			Updated the output tables with new output	1	12 to 13	
			Select the mode based on Select_cal_cos_mode catalog key	2		
	Road Assignment (3)		Execution order, start the road mode	3	1	
			Loop between vehicle classes (from 1 to 5)	3	2	
			Set the values for VOT and KMCOST reading the Vehicle_parameters table	3	3	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	3	4	
			Define the speed values and free flow times	3	5	
			Assignment	3	6	
			Merge the 5 networks, one for each lorry type, into one file	3	7	
			Updated the outputs table with new output	3	8 to 9	
			Skip all the other steps if Select_cal_cos_mode has "Road" value	3	10	
	Rail Assignment (4)		Execution order, start the rail mode	4	1	
			Loop between vehicle classes (from 6 to 13)	4	2	
			Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	4	3	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	4	4	
			Assignment	4	5	
			For all the vehicle classes except 202 there is a unique mode, so jump the subsequent steps. For 202 go to next step	4	6	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	4	7	
			Assignment	4	8	
			Label to skip the steps 7 and 8 if vehicle class different from 202	4	9	
			Merge the 9, one for each train type (one is done for vehicle 203, but that one is integrated with 202),networks into one file	4	10	
			Updated the outputs table with new output	4	11 to 12	
			Skip all the other steps if Select_cal_cos_mode has "Rail" value	4	13	
	Sea Assignment (5)		Execution order, start the sea mode	5	1	
			Loop between vehicle classes (from 14 to 34)	5	2	
			Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	5	3	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	5	4	
			Assignment	5	5	
			Merge the 21, one for each sea vessel or ferry type, networks in one file	5	6 to 8	
			Updated the outputs table with new output	5	9 to 10	

Application			Description	Programs	Subprograms	
			Skip all the other steps if Select_cal_cos_mode has "Sea" value	5	11	
	Air Assignment (6)		Execution order, start the air mode	6	1	
			Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	6	2	
			Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	6	3	
			Assignment	6	4	
			Save the network in geodatabase format	6	5	
			Updated the outputs table with new output	6	6 to 7	
			Skip all the other steps if Select_cal_cos_mode has "Air" value	6	8	
			Set the execution order	7		
Results (7)			Label to start the report phase and check in the matrices and the networks exist under the scenario folder	1		
			Merge all the assigned networks by mode into a single, common network	2		
			Create the report for vehicle and vehicle km	3		
			Save the report in the geodatabase	4		
	Results 1.0 (5)		Create network with bidirectional flow	5	1	
			Produce report 3	5	2	
		Port Areas (3)	Read report 6	5	3	1
			Sum over port areas	5	3	2
			Produce report 13	5	3	3
		Oresund Kiel and Jylland (4)	Merge final network with other statistics	5	4	1
			Calculate total values	5	4	2
			Produce report 14	5	4	3
		Reports per geographical aggregation (5)	Merge Prop_link_Base2006 with scenario table	5	5	1
			Dump link table for assigned network	5	5	2
			Add CATEGORY ID_COUNTRY REGION ID_COUNTY attributes based on values in Prop_link	5	5	3
			Produce report 9	5	5	4
			Produce first part of report 17	5	5	5
			Produce first part of report 18	5	5	6
			Produce report 16 adding county names	5	5	7
			Produce report 17 adding county names	5	5	8
			Produce report 18 adding county names	5	5	9
			Updated the outputs table with new output	6 to 7		
			Update REPORT_GS_{SCENARIO_SHORTNAME}.prn file with the list of outputs created during the run	8		

The application structure

Application			Description	Programs	Subprograms	
			Delete all the temporary files	9		
			Update the logfile for entire run	10		

Table 114 – Samgods application - first part Standard Logistics Module

.

Application					Description	Programs	Subprograms			
Rail Capacity Management (8)	Data Preparation LP (1)				Check the execution order and skip eventually the process	1	1			
					Verify if Output0_{SCENARIO_SHORTNAME}.mdb/Loaded_Net_0_Link exists	1	2			
					Send a prompt message during the run if network not present	1	3			
					Dump assigned network in node and link tables	1	4			
					Create {SCENARIO_DIR}\LogMod\RCM\PathTreeNodeUse.txt	1	5			
					Create {SCENARIO_DIR}\LogMod\RCM\Nodes_List.txt	1	6			
					Create {SCENARIO_DIR}\LogMod\RCM\Links_List.txt	1	7			
					Read Capacity_table_{SCENARIO_CODE}.DBF and prepare data for next step	1	8			
					Create {SCENARIO_DIR}\LogMod\RCM\RailLinkCapacitiesBidirectional_STD.dat	1	9			
					Rename OD vehicles from _0 to _STD	1	10			
	Warm start (2)				Read log_file.log and set up variables that will be evaluated in next step	2	1			
					Set the execution order for LP Loop (3) and Adjust capacities (4) depending on which step has been performed in previous run	2	1			

Application					Description	Programs	Subprograms			
	LP Loop (3)	LP0 step (1)			Setup MPS.ctl control file	3	1	1		
					Prepare 1_RUN_MPS_LP0_step1.BAT	3	1	2		
					Execute 1_RUN_MPS_LP0_step1.BAT	3	1	3		
					Check if run finished successfully, otherwise prompt the user	3	1	4		
					Prepare 2_runLP.bat	3	1	5		
					Execute 2_runLP.bat	3	1	6		
					Prepare 3_RUN_MPS_LP0_step2.BAT	3	1	7		
					Execute 3_RUN_MPS_LP0_step2.BAT	3	1	8		
					Check if run finished successfully, otherwise prompt the user	3	1	9		
			Prepare CTL files - Extract procedure (10)		Loop over trains (201-209)	3	1	10	1	
					Read vehicle type variable	3	1	10	2	
					Create EXTRACTXXLP0.CTL files	3	1	10	3	
					Set up files to describe then number of processors and batch file for call the Samgods_Parallelization_Module.jar	3	1	11		
					Run 4_extractallLP0.BAT using Parallelization Module or not depending on user choice	3	1	12		
					Back up LP0 results before moving to the next iteration	3	1	13		
					Update log_file.log file	3	1	14		
					Skip next steps (user choice)	3	1	15		
		LP1+ (2)			Label for LP1+ process	3	2	1		

The application structure

Application					Description	Programs	Subprograms			
					Loop over maximum number of LP1 from catalog key "Maximum number of loops for Linear Programming process (LP)"	3	2	2		
					Set minimum loop number for a warm start	3	2	3		
			Prepare CTL files (4)		Loop over trains (201-209)	3	2	4	1	
					Read vehicle type variable	3	2	4	2	
					Create EXTRACTXX_LP1.CTL files	3	2	4	3	
			Run BuildChain RCM and ChainChoiRCM (5)		Print out the RUNALL_RCM.BAT batch file for running the column generation step for all the commodities	3	2	5	1	
					Print out the batch file for calling the BuildChain and ChainChoi programs in proper sequence	3	2	5	2	
					Select if run parallelization or normal procedure	3	2	5	3	
					Label to jump to parallelization process	3	2	5	4	
					Prepare control files for parallelization process	3	2	5	5	
					Run parallelization process	3	2	5	6	
			Run MPS LP Extract (6)		Prepare 1_RUN_MPS_LP1_step1.BAT	3	2	6	1	
					Execute 1_RUN_MPS_LP1_step1.BAT	3	2	6	2	
					Check if run finished successfully, otherwise prompt the user	3	2	6	3	
					Prepare 2_run1.bat	3	2	6	4	
					Execute 2_run1.bat	3	2	6	5	
					Prepare 3_RUN_MPS_LP1_step2.BAT	3	2	6	6	

Application					Description	Programs	Subprograms			
					Execute 3_RUN_MPS_LP1_step2.BAT	3	2	6	7	
					Check if run finished successfully, otherwise prompt the user	3	2	6	8	
					Set up files to describe the number of processors and batch file for call the Samgods_Parallelization_Module.jar	3	2	6	9	
					Run 4_extractallLP1.BAT using the Parallelization Module or not depending on user choice	3	2	6	10	
			Check convergence (7)		Setup RCM_last_loop	3	2	7	1	
					Back up current loop	3	2	7	2	
					Update log_file.log	3	2	7	3	
					Skip the process if the maximum number of loops has been reached	3	2	7	4	
	LP1 Adjust Capacities (4)	Data Preparation (1)	Conversion from LogMod to VY (1)		Label to skip Adjust process	4	1	1	1	
					Loop for all the train types (from 201 to 209)	4	1	1	2	
					Load the vehicle number from the Vehicle_parameters table	4	1	1	3	
					Covert the 314 format into the matrix format for loaded, empties and tonnes OD matrices	4	1	1	4 to 6	
					Merge the loaded, empty vehicles and tonnes matrices	4	1	1	7	
			Empty Assignment (2)		Read number of bidirectional links with capacity value (maximum ID_LINK)	4	1	2	1	
					Prepare network for assignment	4	1	2	2	
					Calculate OD-matrices for loaded , empty trains, and Tonne into matrices in daily volumes applying the number of operating days in a year	4	1	2	3	
					Assign matrices for loaded vehicles and tonne flows	4	1	2	4	

The application structure

Application					Description	Programs	Subprograms			
					Set variables in the network for empty assignment	4	1	2	5	
					Create dummy matrix with information of domestic / other trips for trains	4	1	2	6	
					Divide the OD empty matrix in two matrices, one for domestic flows, the other for all the rest	4	1	2	7	
					Perform empty assignment	4	1	2	8	
			Setup Capacity Values (3)		Compute overloads and links over capacity	4	1	3	1	
					Modify capacity based on method explained in 0	4	1	3	2	
					Produce a revised {SCENARIO_DIR}\RailLinkCapacitiesBidirectional_LP1AdjC(1).DBF	4	1	3	3	
					Produce a revised LogMod\RCM\RailLinkCapacitiesBidirectional_LP1AdjC(1).dat	4	1	3	4	
		LP Adjust Capacity Loop (2)			Label for start Adjust Capacity procedure	4	2	1		
					Loop over number of LP1b loops specified via "Maximum number of loops for Adjust Capacity Procedure"	4	2	2		
					Control variables for warm start procedure	4	2	3		
			LP1b step (4)		Setup MPS_LP1.ctl control file	4	2	4	1	
					Prepare 1_RUN_MPS_LP1Adj(1)_step1.BAT	4	2	4	2	
					Execute 1_RUN_MPS_LP1Adj(1)_step1.BAT	4	2	4	3	
					Check if run finished successfully, otherwise prompt the user	4	2	4	4	
					Prepare 2_runLPb1.bat	4	2	4	5	
					Execute 2_runLPb1.bat	4	2	4	6	
					Prepare 3_RUN_MPS_LP1Adj(1)__step2.BAT	4	2	4	7	

Application					Description	Programs	Subprograms			
					Execute 3_RUN_MPS_LP1Adj(1)__step2.BAT	4	2	4	8	
					Check if run finished successfully, otherwise prompt the user	4	2	4	9	
				Prepare CTL files - Extract procedure (10)	Loop over trains (201-209)	4	2	4	10	1
					Read vehicle type variable	4	2	4	10	2
					Create EXTRACTXX_LP1Adj(1).CTL files	4	2	4	10	3
					Set up files to describe then number of processors and batch file for call the Samgods_Parallelization_Module.jar	4	2	4	11	
					Run 4_extractallLP1Adj(1).BAT using Parallelization Module or not depending on user choice	4	2	4	12	
			Conversion from LogMod to VY(5)		Loop for all the commodities (from 201 to 209)	4	2	5	1	
					Load the vehicle number from Vehicle_parameters table	4	2	5	2	
					Covert the 314 format into the matrix format for loaded, empties and tonnes OD matrices	4	2	5	3 to 5	
					Merge the matrices	4	2	5	6	
			Rail assignment (6)		Read number of bidirectional links with capacity value (maximum ID_LINK)	4	2	6	1	
					Prepare network for assignment	4	2	6	2	
					Calculate OD-matrices for loaded , empty trains, and Tonne into matrices in daily volumes applying the number of operating days in a year	4	2	6	3	
					Assign Daily Loaded and Tons matrices	4	2	6	4	
					Set variables in the network for empty assignment	4	2	6	5	
					Create dummy matrix with information of domestic / other trips for trains	4	2	6	6	

The application structure

Application					Description	Programs	Subprograms			
					Divide the OD empty matrix in two matrices, one for domestic flows, the other for all the rest	4	2	6	7	
					Perform empty train assignment	4	2	6	8	
			Setup Capacity Values (7)		Compute overloads and links over capacity	4	2	7	1	
					Modify capacity based on method explained in 0	4	2	7	2	
					Set variable for next loop	4	2	7	3	
					Produce a revised {SCENARIO_DIR}\RailLinkCapacitiesBidirectional_LP1AdjC(2).DBF	4	2	7	4	
					Produce a revised LogMod\RCM\RailLinkCapacitiesBidirectional_LP1AdjC(2).dat	4	2	7	5	
			Convergence Criteria(8)		Calculate number of overcapacity links	4	2	8	1	
					Write Report_LP1b.csv (list of number of links overcapacity per loop)	4	2	8	2	
					Back up LP1AdjustC(1) input and output data	4	2	8	3	
					Update log_file.log	4	2	8	4	
					Stop cycling process if number of links with more than 1 train or more than 1% is zero (no links with flows over its capacity)	4	2	8	5	
	Run final process (5)	Prepare control files (1)			Label for final process	5	1	1		
					Loop over vehicle types (1-35)	5	1	2		
					Set current vehicle type	5	1	3		
					Build control file for extract procedure	5	1	4		
					Build control file for mergerep	5	1	5		

Application					Description	Programs	Subprograms			
					Print out the RUNALL_FIN.BAT batch file for calling the LP2CC program for all the commodities	5	2	1		
					Print out the batch file COMMODITY.BAT for call alternatively the BuildChain and ChainChoi programs	5	2	2		
					Create the batch file for the extract procedure	5	2	3		
					Change the execution order if selected the normal logistic model or advanced. Call the batch files RUNALL.BAT and EXTRACTALL.BAT if normal procedure has been chosen	5	2	4		
		Run the final process (2)			Label to jump the normal logistics model and run advanced	5	2	5		
					Setup the batch file Logmod\EXTRACTALL.BAT	5	2	6		
					Setup the batch file Logmod\EXTRACT.BAT	5	2	7		
					Set up files to describe the number of processors and batch file for calling Samgods_Parallelization_Module.jar	5	2	8		
					Run RUNALL.BAT and EXTRACTALL.BAT using the Parallelization Module	5	2	9		
					Create the batch file for merge.exe	5	2	10		
					Call the external program "Merge reports"	5	2	11		
					Save the report file from ChainChoi.exe (.rep)	5	3	1		
					Change the format of the report file in Voyager	5	3	2 to 3		
					Merge domestic distances from LOS files for subsequent step	5	3	4		
					Delete temporary files used to accumulate results from all commodities	5	3	5 to 9		
					Loop over commodities (ID_LOOPC)	5	3	10		
					Filter node table and produce table with only zones	5	3	11		
		Save Reports (3)			Control the commodities that must be used (if all or STAN group or single commodity)	5	3	12		
					Delete headers from ChainChoi\OUTPUT\ChainChoiXX01.out files	5	3	13		
					Calculate sum of variables for reports 5,6,7,8	5	3	14		
					Merge ChainChoiXX.rep files (input for reports 10 and 11)	5	3	15		
					Merge VhclRepXXrep files (input for reports 12)	5	3	16		
					Calculate report 5 and 6	5	3	17		
					Calculate report 7 and 8	5	3	18		
					Calculate report 10 and 11	5	3	19		
					Calculate report 12	5	3	20		

The application structure

Application					Description	Programs	Subprograms			
					Rename LogMod\LOG\logxx.log into logxxSTD.log	5	3	21		
					Updated output tables with new output	5	3	22 to 23		
					Label to skip the report phase	5	3	25		
RCM Assignment (9)	Conversion from LogMod to VY (1)	Note: Same as for STD			Loop for all the commodities (from 1 to 35)	1	1			
					Load the vehicle number from Vehicle_parameters table	1	2			
					Covert the 314 format into matrix format	1	3 to 5			
					Merge the matrices by mode in a unique MAT file	1	6 to 11			
					Updated output tables with new output	1	12 to 13			
	Road Assignment (2)	Note: Same as for STD			Loop between vehicle classes (from 1 to 5)	2	1			
					Set the values for VOT and KMCOST reading the Vehicle_parameters table	2	2			
					Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	2	3			
					Define the speed values and free flow times	2	4			
					Assignment	2	5			
					Merge the 5 network in one file	2	6			
					Updated the outputs table with new output	2	7 to 8			
	Rail Assignment (3)	Note: Different from STD			Read number of bidirectional links with capacity value (maximum ID_LINK)	3	1			
					Prepare network for assignment	3	2			
					Calculate OD Empty, Loaded and Tons matrices in daily volumes	3	3			
					Assign Loaded and Tons matrices	3	4			
					Set variables in the network for empty assignment	3	5			
					Create dummy matrix with information of domestic / other trips	3	6			
					Divide the OD empty matrix in two matrices, one for domestic flows, the other for all the rest	3	7			
					Perform empty assignment	3	8			
					Merge results from empty assignment and tons and loaded assignment	3	9			
					Updated the outputs table with new output	3	10			

Application					Description	Programs	Subprograms			
							to 11			
	Sea Assignment (4)	Note: Same as for STD			Loop between vehicle classes (from 14 to 34)	4	1			
					Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	4	2			
					Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	4	3			
					Assignment	4	4			
					Merge the 21 networks in one file	4	5 to 7			
					Updated the outputs table with new output	4	8 to 9			
					Set the values for VOT and KMCOST reading the Vehicle_parameters table for the first mode	6	1			
					Calculate the Km_TAX, TOLL, DIST_COST and VOT variables	6	2			
					Assignment	6	3			
					Save the network in geodatabase format	6	4			
					Updated the outputs table with new output	6	5 to 6			
Results RCM (10)					Merge all the assigned network by mode in a unique network	1				
					Create the report for vehicle and vehicle km	2				
					Save the report in the geodatabase	3				
	Results 1.0 (4)				Create network with bidirectional flow	4	1			
					Compare assigned networks STD and RCM	4	2			
					Produce report 3	4	3			
		Port Areas (4)			Read report 6	4	4	1		
					Sum over port areas	4	4	2		
					Produce report 13	4	4	3		
		Oresund Kiel and Jylland (5)			Merge final network with other statistics	4	5	1		
					Calculate total values	4	5	2		
					Produce report 14	4	5	3		
		Rail Capacity (6)			Create bidirectional flow from LOAD_NET_RAIL_RCM	4	6	1		
					Create report 15	4	6	2		
					Create report 15b	4	6	3		
		Reports per geographical aggregation (7)			Merge Prop_link_Base2006 with scenario table	4	7	1		
					Dump link table for assigned network	4	7	2		
					Add CATEGORY ID_COUNTRY REGION ID_COUNTY attributes based	4	7	3		

The application structure

Application					Description	Programs	Subprograms			
					on values in Prop_link					
					Produce report 9	4	7	4		
					Produce first part of report 17	4	7	5		
					Produce first part of report 18	4	7	6		
					Produce report 16 adding county names	4	7	7		
					Produce report 17 adding county names	4	7	8		
					Produce report 18 adding county names	4	7	9		
					Updated the outputs table with new output	5 to 6				
					Save RCM results and other report	7				
					Update REPORT_GS_{SCENARIO_SHORTNAME}.prn file with the list of outputs created during the run	8				
Parameters Calculation (11)					Skip the process if user selected to run step by step some parts	1				
					Calculation of scaling factors per port area and STAN group. See 0 for details	2				
					Prepare data for next calculation	3				
					Calculation of scaling factor for Kiel canal. See 0 for details	4				
					Label to skip the process	5				
Pilot #12					End of cycling process					

Table 115 – Samgods application - second part Rail Capacity Management Module.

"Compare scenarios" application

Input data: LOS matrices , OD matrices STD, loaded networks STD and RCM for base scenario and selected scenario.

Output data: matrices with differences and networks with differences under scenario folder.

Purpose:

Compare the matrices and the loaded network between scenarios.

Application		Description	Programs	Subprograms
LOS matrices (1)		Execution order based on "Select_compare" catalog key	1	1
		Compare Road LOS matrix between base scenario and selected scenario	1	2
		Compare Rail LOS matrix between base scenario and selected scenario	1	3
		Compare Sea LOS matrix between base scenario and selected scenario	1	4
		Compare Air LOS matrix between base scenario and selected scenario	1	5
OD Matrices (2)		Execution order based on "Select_compare" catalog key	2	1
		Compare OD Loaded vehicle matrices by road	2	2
		Compare OD Loaded vehicle matrices by rail	2	3
		Compare OD Loaded vehicle matrices by sea	2	4
		Compare OD Loaded vehicle matrices by air	2	5
		Compare OD tonne matrices by road	2	6
		Compare OD tonne matrices by rail	2	7
		Compare OD tonne matrices by sea	2	8
		Compare OD tonne matrices by air	2	9
		Compare OD empty vehicle matrices by road		10
		Compare OD empty vehicle matrices by rail		11
		Compare OD empty vehicle matrices by sea		12
		Compare OD empty vehicle matrices by air		13
Assignment (3)		Execution order based on "Select_compare" catalog key	3	1
		Compare the loaded networks from Standard Logistics Module		2
		Compare the loaded networks from Rail Capacity Management Module		3
		End of process		4

"Handling scenario"

Input data: For the export function of the entire model.

Output data: Depends on the choice made by user. With the export function a new model will be set up in the folder indicated by "New_folder" catalog key. It will also result in an exported scenario in another model in case the "Exp_selection" catalog key is set to 2.

Purpose:

To handle the model in the following ways:

1. of deleting a scenario,
2. compress the geodatabases,
3. export a model or a scenario,
4. check the catalog values for an imported scenario.

Application			Description	Programs	Subprograms			
			Set the execution order based on "Delete" catalog key		1	1		
			Check if the scenario is in read mode. In this case go to the end of process		1	2		
Delete (1)			Skip all the other applications		1	3		
Compact (2)			Call the python script compact_GDB.pyt to compress the input_data.mdb and output0.mdb		2	1	1	
	Model Export (1)		If "Exp_selection" is equal to 1, create the entire model structure in a new folder and copy all the applications and script files.		2	1	1	
	General Tables (2)		Create DaTE_TIME.TXT to check the data of the system and change the execution order based on "Exp_selection" catalog key		2	2	1	
			Update the Model_description table in the new Input_Data.mdb file in the new model folder		2	2	2 to 3	
			Copy the tables and feature classes from the existing input_data.mdb to the new one. Also if the "Exp_selection" is set to 2, verify that the Input_Data.mdb exists in the new folder		2	2	4	
			If not exist jump all the processes to allow a verification by user.		2	2	5 to 6	
			Update the General_NewBase table in the new geodatabase		2	2	7	
			Update the LogMod_NewBase table in the new geodatabase		2	2	8	
	Create the new base (3)		Skip all these steps if "Exp_selection" catalog key is set to 2. The previous check (2.3.4) has pointed out if is possible run this part.		2	3	1	
			Merge SC_{SCENARIO_SHORTNAME}_Link and SC_{SCENARIO_SHORTNAME}_Node with base data to create the new base network		2	3	2	
			Merge the SC_{SCENARIO_SHORTNAME}_Frequency_Data_Link and SC_{SCENARIO_SHORTNAME}_Frequency_Data_Node with the base frequency_data_Base2006 to obtain the new frequency network		2	3	3	
			Merge Sc_{SCENARIO_SHORTNAME}_Cargo and Cargo_Base2006 to create the new cargo table		2	3	4	

Application			Description	Programs	Subprograms			
			Merge Sc_ {SCENARIO_SHORTNAME}_Node_Terminals with Node_Terminals_Base2006 to obtain the new Node_Terminals Base table		2	3	5	
			Merge Sc_{SCENARIO_SHORTNAME}_Nodes_Commodities with Nodes_Commodities_Base2006 to create the new Nodes_Commodities table		2	3	6	
			Merge SC_Nodes_{SCENARIO_SHORTNAME}to Nodes_Base2006 to obtain the new base nodes table		2	3	7	
			Create the new Tax_Country_{SCENARIO_SHORTNAME}table joining Sc_{SCENARIO_SHORTNAME}_Tax_Country with Tax_Country_Base2006		2	3	8	
			Create the new Tax_Category_{SCENARIO_SHORTNAME}table joining Sc_{SCENARIO_SHORTNAME}_Tax_Category with Tax_Category_Base2006		2	3	9	
			Create the new Tax_Link_{SCENARIO_SHORTNAME}table joining Sc_{SCENARIO_SHORTNAME}_Tax_Link with Tax_Link_Base2006		2	3	10	
			Create the new Toll_Link_{SCENARIO_SHORTNAME}table joining Sc_{SCENARIO_SHORTNAME}_Toll_Link with Toll_Link_Base2006		2	3	11	
			Merge Vehciles_parameters_Base2006 with SC_{SCENARIO_SHORTNAME}_Vehicle_parameters to obtain a new Vehicle_parameters base table		2	3	12	
			Merge ProprLink_Base2006 with SC_{SCENARIO_SHORTNAME}_ProprLink to obtain a new PropLink base table		2	3	13	
			Merge Rail_Capacity_Base2006 with SC_{SCENARIO_SHORTNAME}_Rail_Capacity to obtain a new Rail_Capacity base table		2	3	14	
			Merge Vehicles_exception_Base2006 with SC_{SCENARIO_SHORTNAME}_Vehicles_exc to obtain a new Vehicles_exception base table		2	3	15	
	Create the new scenario specific tables (4)	Create the scenario tables	Do the next steps if "Exp_selection" is equal to 2. If equal 1 the base scenario already exists		2	4	1	1
			Rebuild the base scenario for the current scenario		2	4	1	2 to 15
		Create the scenario	Compare all the tables between the new base and the current scenario		2	4	2	1 to 17

Application			Description	Programs	Subprograms			
		specific tables						
Scenario Import (4)			Select the execution order based on "Imp_Selection" catalog key and create Ckeck.txt file to verify if the scenario exists in the Input_Data.mdb		1			
			Check if the selected scenario exist, otherwise give a prompt and go to the end of process		2 to 3			
			Compare the values for the catalog keys existing in the catalog file and in general_{SCENARIO_SHORTNAME}table, creating a report REPORT_HL_BS05.TXT		4			

"PWC_matrices" application

Input data: PWC matrices under SamGods\ 01_Programs\LogMod\Input\{Year}\PWC folder in txt format. {Year} is the catalog key controlling the set of matrices per each provided year (2006, 2030 or 2030BnM).

Output data: PWC matrices under SamGods\ 01_Programs\LogMod\Input\PWC\VY_F folder in binary voyager format.

Purpose:

Visualize the PWC_matrices with Cube Interface.

"Change matrix format" application

Input data: COST, OD matrices for vehicles and tonnes under SamGods\Scenario_tree\{Scenario_name} folder.

Output data: dbf or txt file of a selected matrix.

Purpose:

Export a specific matrix into dbf or csv format.

Applied methods in the model

Manage different loading times, costs and capacities for different commodities

The following attributes have been specified for defined commodities or STAN groups:

- KM COST and HOUR COST for commodity 17
- NC_LCO_DRY, NC_LTI_DRY, CONT_LCO and CONT_LTI for STAN groups 10 and 11
- CAPACITY for commodity 15

To setup this differentiation the following technique has been applied:

1) Each commodity belongs to a different commodity group (Dry Bulk, General Cargo or Liquid Bulk);
2) For those commodities that should have a different setup compared to default values a new COST file has produced with updated values:
    a. VHCLS_DRY_BULK_IRON.TXT - commodity 15 with CAPACITY revised for Vehicle class205;
    b. vhcls_gen_cargoSTEEL.txt - commodity 17 - revised KM COST and HOUR COST for wagon load 207, 208 and 209;
    c. VHCLS_DRY_BULK_S10_11.TXT - STAN group 10 and 11 - commodities 18, 19, 20, 21, 23 - revised NC_LCO_DRY, NC_LTI_DRY, CONT_LCO and CONT_LTI values for wagon load 207, 208 and 209.

    This implies that under Input\COST are present 6 files instead of the standard 3 (vhcls_dry_bulk.txt, vhcls_gen_cargo.txt, vhcls_liq_bulk.txt)
3) In each control file for BuildChainXX.ctl the definition of VHCL control parameter that gives the name and location of vehicle parameters has maintained the default values.
4) In each control file for ChainCHainXX.ctl the definition of VHCL control parameter that gives the name and location of vehicle parameters has referred to the different cases. For instance in ChainChoi15.ctl VHCL is VHCL=..\Input\Cost\VHCLS_DRY_BULK_IRON.TXT.

In the following table the list of VHCL definitions in BuildChain and ChainChoi.

ID COM	STAN NR	AGGR_COM	COM TYPE	VHCL in BuildChain	VHCL in ChainChoi
1	1	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
2	1	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
3	1	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
4	1	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
5	2	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
6	3	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
7	3	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
8	3	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
9	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
10	4	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
11	1	Liquid bulk	2	vhcls_liq_bulk.txt	vhcls_liq_bulk.txt
12	5	Liquid bulk	2	vhcls_liq_bulk.txt	vhcls_liq_bulk.txt
13	5	Liquid bulk	2	vhcls_liq_bulk.txt	vhcls_liq_bulk.txt
14	6	Liquid bulk	2	vhcls_liq_bulk.txt	vhcls_liq_bulk.txt
15	7	Dry bulk	1	vhcls_dry_bulk.txt	VHCLS_DRY_BULK_IRON.TXT
16	7	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt

ID COM	STAN NR	AGGR_COM	COM TYPE	VHCL in BuildChain	VHCL in ChainChoi
17	8	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargoSTEEL.txt
18	10	Dry bulk	1	vhcls_dry_bulk.txt	VHCLS_DRY_BULK_S10_11.TXT
19	10	Dry bulk	1	vhcls_dry_bulk.txt	VHCLS_DRY_BULK_S10_11.TXT
20	10	Dry bulk	1	vhcls_dry_bulk.txt	VHCLS_DRY_BULK_S10_11.TXT
21	11	Dry bulk	1	vhcls_dry_bulk.txt	VHCLS_DRY_BULK_S10_11.TXT
22	6	Liquid bulk	2	vhcls_liq_bulk.txt	vhcls_liq_bulk.txt
23	11	Dry bulk	1	vhcls_dry_bulk.txt	VHCLS_DRY_BULK_S10_11.TXT
24	9	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
25	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
26	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
27	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
28	9	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
29	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
30	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
31	2	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
32	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
33	9	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt
34	12	Dry bulk	1	vhcls_dry_bulk.txt	vhcls_dry_bulk.txt
35	12	General cargo	3	vhcls_gen_cargo.txt	vhcls_gen_cargo.txt

Table 116 – List of values for VHCL control parameter in BuildChain and ChainChoi control files.

Manage different consolidation factors for different commodity groups

The consolidation range, expressed as a lower and upper limit, can be specified in two different ways in the general control file for BuilChain and ChainChoi.

A first control parameter, CONSOL, represents a general range used when no mode specific range is specified. These range can be specified in the GUI via the following two catalog keys:

- Lower bound for consolidation factor [Tonnes] (new) (applied to all commodities):0.05
- Upper bound for consolidation factor [Tonnes] (new) (applied to all commodities):0.95

A second set of CONSOL<mode> parameters can be specified using the table:

- "Low and upper bounds for consolidation factors" under Scenario Inputs\General tables of Data Panel.

The access to this table is only for developer or advance user.

The Logistics Module will give the priority to the second values , specified by submode and commodity, and ignore the first general values. An example of general control file with these parameters specified is:

```
CONSOL=0.05,0.95
CONSOLA=0.05,0.5
CONSOLB=0.05,0.5
CONSOLC=0.05,0.5
CONSOLS=0.5,0.9
```

Table 117 General control parameter - section related to CONSOL parameters. (yellow=CONSOL green=CONSOL<mode>).

In the calibration process a unique set of consolidation parameters per mode has shown limitations and different sets for specific commodities have been used for the calibration.

The CONSOL<mode> values are therefore specified in four different general control files:

- BuildChain_Common.ctl, ChainChoi_Common.ctl
- BuildChain_STAN2.ctl, ChainChoi_STAN2.ctl
- BuildChain_STAN8.ctl, ChainChoi_STAN8.ctl
- BuildChain_STAN9.ctl, ChainChoi_STAN9.ctl

Where CONSOL<mode> values are based on the table "Low and upper bounds for consolidation factors":

ID	MODE_C	CONSOL_L	CONSOL_U	Mode	CONSOL_L2	CONSOL_U2	CONSOL_L8	CONSOL_U8	CONSOL_L9	CONSOL_U9
1	A	0.03	0.24	Road	0.03	0.24	0.03	0.24	0.03	0.24
2	B	0.02	0.3	Road	0.02	0.3	0.02	0.3	0.02	0.3
3	C	0.05	0.3	Road	0.05	0.3	0.05	0.3	0.05	0.3
4	D	0.56	0.88	Rail	0.61	0.98	0.61	0.98	0.61	0.98
5	E	0.56	0.88	Rail	0.61	0.98	0.61	0.98	0.61	0.98
6	F	0.56	0.88	Rail	0.61	0.98	0.61	0.98	0.61	0.98
7	G	0.56	0.88	Rail	0.61	0.98	0.61	0.98	0.61	0.98
8	H	0.56	0.88	Rail	0.61	0.98	0.61	0.98	0.61	0.98
9	I	0.66	0.98	Rail	0.71	0.98	0.71	0.98	0.71	0.98
10	T	0.66	1	Rail	0.71	1	0.71	1	0.71	1
11	U	0.66	0.98	Rail	0.71	0.98	0.71	0.98	0.71	0.98
12	J	0.01	0.25	Sea	0.01	0.25	0.01	0.25	0.01	0.25
13	K	0.01	0.25	Sea	0.01	0.25	0.01	0.25	0.01	0.25
14	L	0.01	0.25	Sea	0.01	0.25	0.01	0.25	0.01	0.25
15	M	0.01	0.25	Sea	0.01	0.25	0.01	0.25	0.01	0.25
16	N	0.01	0.25	Sea	0.01	0.25	0.01	0.25	0.01	0.25
17	O	0.01	0.25	Sea	0.01	0.25	0.01	0.25	0.01	0.25
18	P	0.15	0.9	Sea	0.15	0.9	0.15	0.9	0.15	0.9
19	Q	0.15	0.9	Sea	0.15	0.9	0.15	0.9	0.15	0.9
20	R	0.15	0.8	Air	0.15	0.8	0.15	0.8	0.15	0.8
21	S	0.1	0.35	Road	0.1	0.35	0.1	0.35	0.1	0.35

Table 118 CONSOL<mode> parameters General (CONSOL_L, CONSOL_U) and STAN groups 2, 8, 9.

In each BuilChainxx.ctl or ChainChoixx.ctl the control parameter INCL=points to the right values depending if xx is in any STAN2, 8, 9 groups or not. In all the other cases, the general values will be used.

In the following table the INCL definitions for BuildChain and ChainChoi are listed

ID COM	STAN NR	AGGR_COM	COM_TYPE	INCL in BuildChain	INCL in ChainChoi
1	1	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl

ID COM	STAN NR	AGGR_COM	COM_TYPE	INCL in BuildChain	INCL in ChainChoi
2	1	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
3	1	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
4	1	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
5	2	Dry bulk	1	BuildChain_STAN2.ctl	ChainChoi_STAN2.ctl
6	3	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
7	3	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
8	3	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
9	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
10	4	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
11	1	Liquid bulk	2	BuildChain_Common.ctl	ChainChoi_Common.ctl
12	5	Liquid bulk	2	BuildChain_Common.ctl	ChainChoi_Common.ctl
13	5	Liquid bulk	2	BuildChain_Common.ctl	ChainChoi_Common.ctl
14	6	Liquid bulk	2	BuildChain_Common.ctl	ChainChoi_Common.ctl
15	7	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
16	7	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
17	8	General cargo	3	BuildChain_STAN8.ctl	ChainChoi_STAN8.ctl
18	10	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
19	10	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
20	10	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
21	11	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
22	6	Liquid bulk	2	BuildChain_Common.ctl	ChainChoi_Common.ctl
23	11	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
24	9	Dry bulk	1	BuildChain_STAN9.ctl	ChainChoi_STAN9.ctl
25	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
26	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
27	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
28	9	Dry bulk	1	BuildChain_STAN9.ctl	ChainChoi_STAN9.ctl
29	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
30	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
31	2	Dry bulk	1	BuildChain_STAN2.ctl	ChainChoi_STAN2.ctl
32	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl
33	9	General cargo	3	BuildChain_STAN9.ctl	ChainChoi_STAN9.ctl
34	12	Dry bulk	1	BuildChain_Common.ctl	ChainChoi_Common.ctl
35	12	General cargo	3	BuildChain_Common.ctl	ChainChoi_Common.ctl

Table 119 – List of values for INCL control parameter in BuildChain and ChainChoi control files.

Capacity constraints on sea mode

There are two different mechanisms in place for the sea mode to limit the usage of some services and ports:

- constraints applied at network level (e.g. by link level, as for the Kiel Canal example). This are managed at the BuildChain step and applies only to the type vehicle. Should a too large vehicle be used as type vehicle, any path using the link cannot be used. As described below this is handled by removing links with too low capacity before running the LOS-calculations for the type vehicle.
- constraints applied in the evaluation of alternatives during ChainChoi process (as per the Vänern Lake). When evaluated at the ChainChoi-level the generated alternatives may be generated with a too large vehicle, but at this level only the vehicles small enough will be allowed. As described below this is handled by using a port capacity limit in lake Vänern for the allowed vessel capacity in the Vänern ports (otherwise the paths would have to be scrutinized after the LOS-calculations which is possible but not efficient).

In the following two paragraphs will be discuss their implementation in the model.

Capacity constrains in Kiel Canal

To apply capacity constraints on network level different actions have been performed:

- provided the information of which Dwell Capacity is allowed in Kiel Canal
- compared the capacity of the vessel type with the allowed capacity in the Kiel Canal
- removed the Kiel Canal as available link when skimming the LOS matrices for sea

The first point required a modification on the input network :

- Modified for the links 551679-550658 and 550658-551679 the mode attribute allowing all the vessel types ("xyzpq")

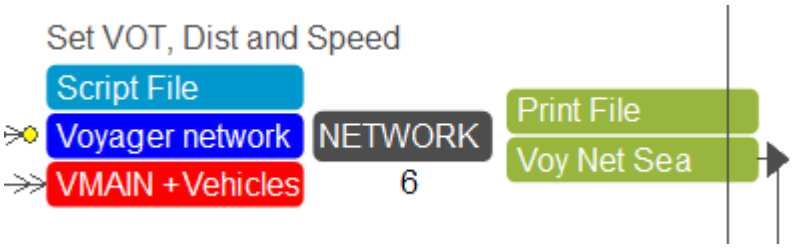
Figure 2 Links for Kiel Canal.

Applied methods in the model

- Coded in UL3 field the maximum dwell capacity of 20.000 t

The second and third point has required a change in the scripting related the network handling and LOS calculation:

1) when setting up the network attributes (Network step 6 under LOS Calculation\Sea application)



The vessel capacity from vehicle parameter table (red file in the picture) is read via a Lookup command:

```
LOOKUP LOOKUPI=1 NAME=SEA_COST,

 LOOKUP[1]=VEH_NR, RESULT=KM_COST, ; KM_COST (distance cost in SEK)

 LOOKUP[2]=VEH_NR, RESULT=HOURS_COST, ; HOURS_COST (Vot in SEK)

 LOOKUP[3]=VEH_NR, RESULT=SPEED,  ; Speed for vehicles classes 1-17

 LOOKUP[4]=VEH_NR, RESULT=CAPACITY ;capacity for vessel - to compare with UL3
```

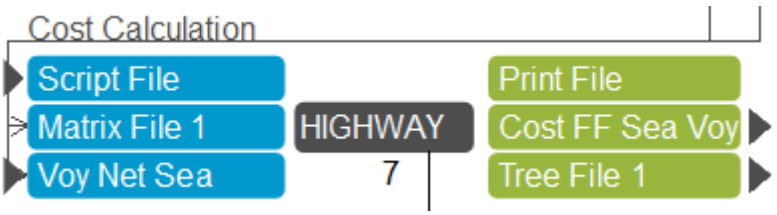
The capacity per each vehicle class (denoted as @MATRIX.N_VEHICLES@) is compared in the script with the capacity coded in the network (LI.1.UL3). If the vessel capacity is greater than the link capacity, the link is marked with a new attribute CLOSED=1:

```
IF(LI.1.UL3<SEA_COST(4,@MATRIX.N_VEHICLES@)&&LI.1.UL3<>0);comparison between vessel capacity and canal capacity coded in UL3

CLOSED=1

ENDIF
```

2) During the LOS calculation (Highway step 7 under LOS Calculation\Sea application)



The information of which links must be closed has recorded in the network attribute CLOSED=1 previously generated in the Network program. Using an IF statement, all the links marked as closed are added to GROUP=1. GROUP is an array available inside the HIGHWAY program to mark group of links and used for different purposes, in this case to exclude those links from the path calculation.

```
IF(LI.CLOSED=1) ADDTOGROUP=1
```

In the phase ILOOP, when generating the paths, the keyword EXCLUDEGROUP filter the available links excluding those on group 1.

```
PATHLOAD PATH=LW.SEA@MATRIX.N_VEHICLES@,
   MW[1]=PATHTRACE(LI.UL2),
   MW[2]=PATHTRACE(LI.T0_SEA@MATRIX.N_VEHICLES@),
   MW[3]=PATHTRACE(LW.SEA@MATRIX.N_VEHICLES@),
   MW[4]=PATHTRACE(LW.EXTRACOST@MATRIX.N_VEHICLES@),
   MW[5]=PATHTRACE(LW.DDIST@MATRIX.N_VEHICLES@),EXCLUDEGROUP=1
```

Capacity constrains in Vänern Lake

Some issues have been detected when applying the method described in the previous paragraph to Vänern Lake:

- the method limits the available choices with connectivity described in the LOS-matrices
- the LOS-matrices are fully applied in ChainChoi procedure for all vehicles. In the BuildChain procedure only the "typical" vessel is applied
- if there are no paths built for the "typical" vessel, no solution would be available for the evaluation
- for the Kiel canal that has not represented a problem since there is an alternative path not using the canal, but for Vänern Lake there is only one path in and out from Vänern. Since the appointed "typical" vessel does not comply with the capacity limits there is not any available connection, and therefore the chain is not constructed. Consequently, ChainChoi cannot at the succeeding stage select an appropriate vessel for the OD-leg through Vänern canal.

The solution applied is therefore the following:

- in the BuilChain stage the values for MaxDwtContainerVessel(Tonnes), MaxDwtRoroVessel(Tonnes) and MaxDwtOtherVessel(Tonnes) are unconstrained (e.g. 9999999)
- in the ChainChain stage the values for MaxDwtContainerVessel(Tonnes), MaxDwtRoroVessel(Tonnes) and MaxDwtOtherVessel(Tonnes) are specified as 4001 tons (limit for ports in the lake).

The solution has required:

- changes in input data (Nodes_Base2006) table adding the VanernCan attribute (with ton limits for each port in port area 14);

ZONEID	NAME	ZONET	DOM	C_TECH	T_TEC	MAXDWTCO	MAXDWTRO	MAXDWTOT	SEAOUTP	SEACO	AIROU1	TranO	PortArea	VanernCan
844721	Sea: Otterbäck	844700	1	1	1	27200	9999999	40000	320	0	0	10	14	4001
848721	Sea: Vänersbor	848700	1	1	1	27200	9999999	40000	220	0	0	10	14	4001
848821	Sea: Trollhättar	0	1	1	1	27200	9999999	40000	10	0	0	10	14	4001
849421	Sea: Lidköping	849400	1	1	1	27200	9999999	40000	530	0	0	10	14	4001
876121	Sea: Skoghall	876100	1	1	1	5000	5000	5000	10	0	0	10	14	4001
878121	Sea: Kristineha	878100	1	1	1	27200	9999999	40000	290	0	0	10	14	4001
876421	Sea: Gruvön	876400	1	1	1	5000	5000	5000	500	0	0	10	14	4001
878021	Sea: Gruvön	878000	1	1	1	5000	5000	5000	170	0	0	10	14	4001

- setting up two different Nodes folder in LogMod structure
    - Logmod\Input\Nodes

    - Logmod\Nodes_CC
- point to different folders in BuildChain_Common.ctl and ChainChoi_Common.ctl
    - NODES=..\Input\Nodes in BuildChain
    - NODES=..\Input\Nodes_CC in ChainChoi

Rail Capacity Management procedure

For a detailed description of Rail Capacity Management procedure we refer to document listed under 0.6. In this paragraph we will describe the implementation in the Cube Interface, and the flow chart of information in each step of the procedure.

As a general concept the RCM procedure is an iterative process that:

- starts from the results obtained from the Standard Logistics Module
- uses some extra input files placed under LogMod\RCM folder
- sets up a linear programming problem that will generate viable alternative solutions in each iteration (compared to the best solution from standard logistics module). Should not any new alternative solutions be generated we are at the optimum!
- produces a new set of output files in terms ChainChoiNN_01LPX.out solutions to set up new OD matrices for empties and loaded vehicles
- the number of iterations is controlled by the user (setting up the related catalog key " Maximum number of loops for Linear Programming process (LP)")

A subsequent step, called Adjust Capacity Procedure, could be run discretionally by the user after the main LP loop. In the next paragraph will be provided further details on this. In this part the main focus is on the Linear Programming problem.

All the steps could be summarized in the following picture:

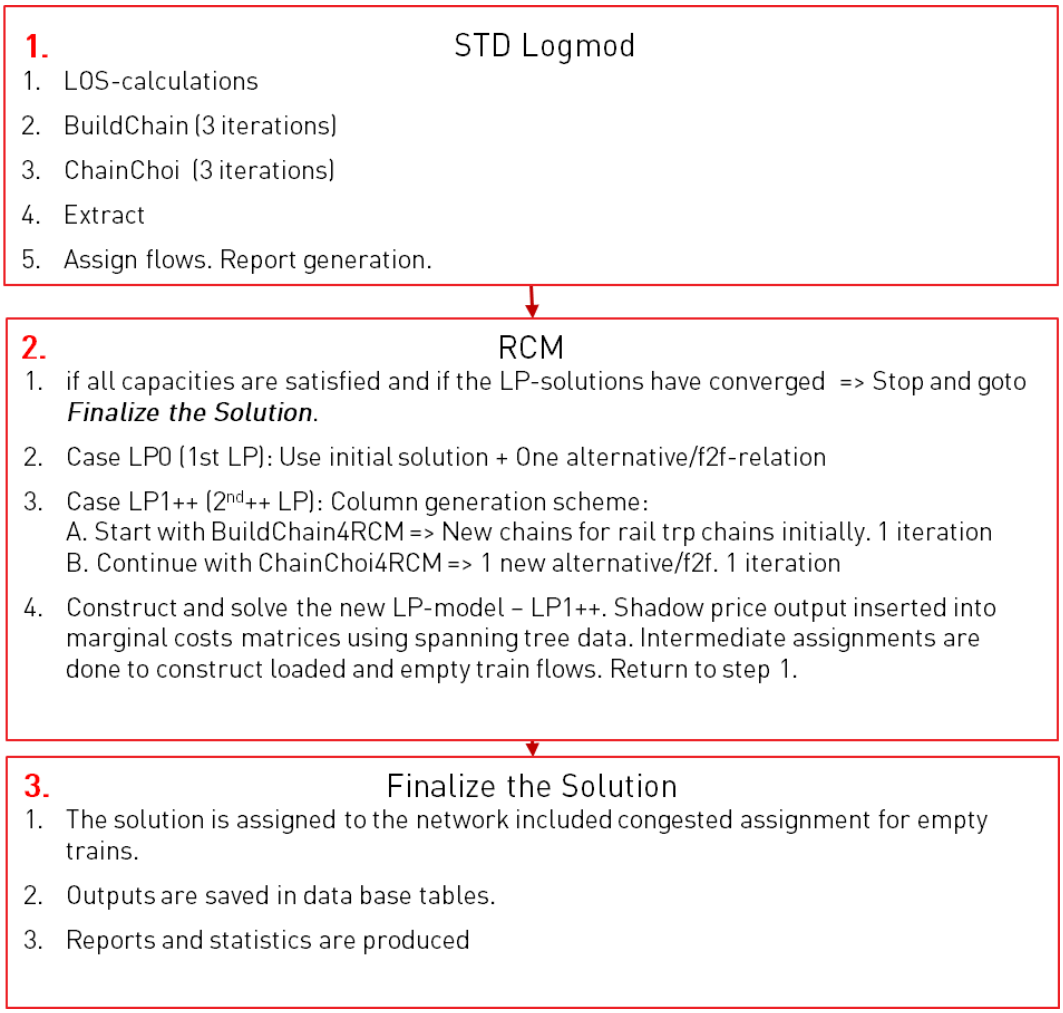


Figure 3 Flow chart of the logistics model and RCM.

LP0 step

LP0 steps corresponds to point 2.2. in Figure 3. This process invokes four different steps with the following purposes:

- step 1: all the results from the Standard Logistics Module, including spanning tree data for the railway paths to allow for link load calculations, are merged together in some extra input files for construction of the standardized MPS-formatted file LP_Rail_LP0.MPS. This is the file with all the variables and equations in the linear programming problem with the proper format for LP_Solve.exe program. The MPS.jar program is called from Cube with its control file mps_LP.ctl that contains all the input and output files (locations and names), and control parameters in the related batch file (JCMW). We refer to the user manual of the Java-program mps.jar for further details, see document listed under Section 0.1.
- step 2: the LP_Solve.exe is called to solve the LP problem and produces the LP_Rail_LP0.out file
- step 3: reads the LP_Rail_Lp0.out files and extract information about marginal costs for active capacity constraints. These are summarized into marginal costs per railway path using the spanning tree data and placed into marginal cost matrices (Vxxx_MC.314). ChainChoi solutions are assembled from the LP-solution and placed into approximate firm-to-firm solution files (ChainChoiNN_02LPX.out). f2f-solutions split into two or more alternatives are split exactly according to the LP-solution, without a detailed recalculation of the costs.
- step 4: From the ChainChoiNN_02LPX.out files the extract procedure is used to estimate a new set of OD matrices for empties and loaded trains (also for tonnes but that is not needed) for all the train types.

In the picture below the file naming and flow chart representing the process is shown:

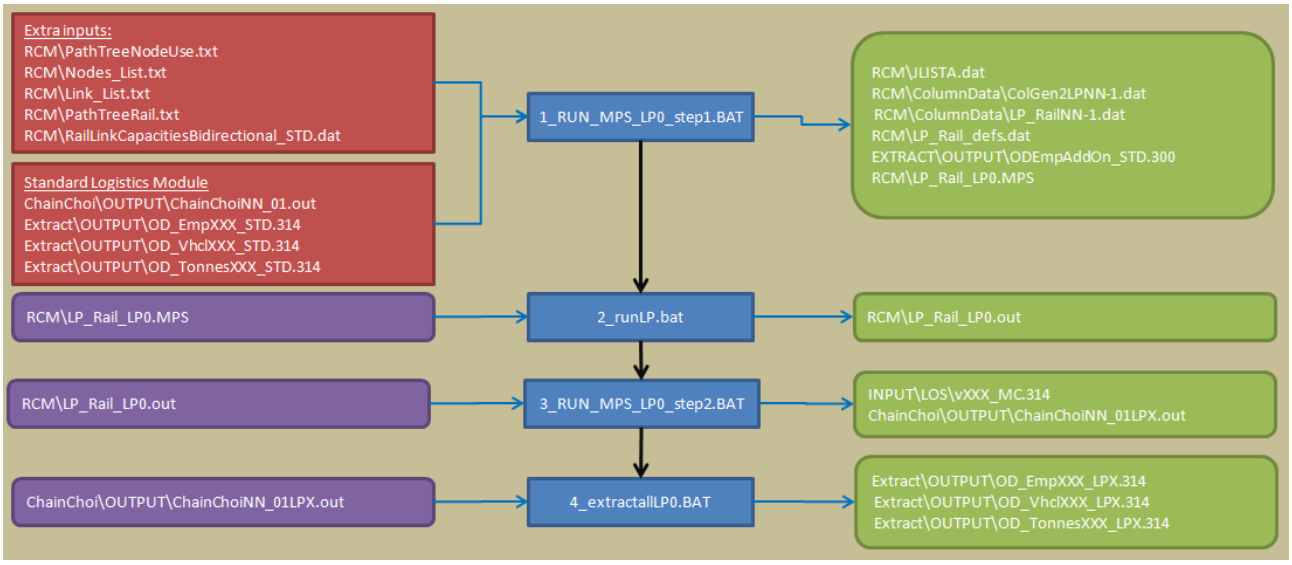


Figure 4 LP0 process: input and output data and processes involved.

The requested input data, apart from the ones produced by the Standard Logistics Module, is produced in different steps of the Samgods application:

1) PathTreeRail.txt: this file contains spanning tree data (see document listed under Section 0.6 Appendix C for details on the structure) that represents all the possible paths from each origin to each destination present in the rail network. This is produced during the LOS calculation for Rail under Samgods\LOS Calculation\Rail application in Highway steps #6 and #9. This is not a standard output for Voyager software since it is a customization required by Trafikverket. The instructions for the proper installation are give in the User Manual.

2) PathTreeNodeUse.txt: full list of nodes used in the paths (it might be the case that some links are not used at all in any path and consequently not the related nodes). This list is used to setup the dimension of some arrays in the computation. The file is produced under Samgods\Rail Capacity Management\Data Preparation LP application at steps #4 and #5

3) Nodes_List.txt and Links_List.txt: List of nodes and links in the network both with emme and voyager number systems and primary key (Internal_node_number) per each element. They are produced under Samgods\Rail Capacity Management\Data Preparation LP application at steps #6 and #7. These are absolutely necessary because the backtracking of the paths utilize the mapping between these two sets of nodes and links respectively.

4) RailLinkCapacitiesBidirectional_STD.dat: this is the translation in text format (ascii) of the Rail_Capacity_{Scenario_ShorName} table for the specific scenario. The file is produced under Samgods\Rail Capacity Management\Data Preparation LP application at steps #8 and #9

The output files from the process

Output file	Description
RCM\JLISTA.dat	List of transport chain solutions from all the commodity groups and definition of superindex, e.g. a primary key for each solution. This primary key will be referred to in the MPS file
RCM\ColumnData\ColGen2LPNN-1.dat	Intermediate file
RCM\ColumnData\LP_RailNN-1.dat	Intermediate file
RCM\LP_Rail_defs.dat	Computed data for the constraints in the Linear programming problem
EXTRACT\OUTPUT\ODEmpAddOn_STD.300	AddOn factors for empty vehicles.
RCM\LP_Rail_LP0.MPS	Linear Programming problem in MPS format that will be solved by LP_Solve.exe program
RCM\LP_Rail_LP0.out	Results of linear programming problem solution in LP_Solve.exe format
INPUT\LOS\vXXX_MC.314	Marginal cost matrices for V201, V202, ..V209 (only rail mode).
ChainChoi\OUTPUT\ChainChoiNN_01LPX.out	Transport chain solutions produced
Extract\OUTPUT\OD_EmpXXX_LPX.314	OD empty vehicles for the rail mode
Extract\OUTPUT\OD_VhclXXX_LPX.314	OD total vehicles for the rail mode
Extract\OUTPUT\OD_TonnesXXX_LPX.314	OD tonnes for the rail mode

Table 120 –Output files from LP0

LP1 step

LP1 steps corresponds to point 2.3. in Figure 3. This process invokes the same four steps as LP0 but with two extra steps:

- step 1: run BuilChain4RCM.exe and ChainChoi4RCM.exe programs (Column generation model). These are special versions of the logistics modules BuildChain and ChainChoi that only operates on the transport relations defined in JLISTA.dat (the ones initially using railway). They are used for computing a new first best solution given the updated marginal costs. The modules are run with the same conditions as in iteration 3 of Standard Logistics Module in which a large number of endogenously generated input data, such as consolidation factors and volumes, are left unchanged.
- step 2, 3,4 and 5: same steps 1, 2,3 and 4 in LP0
- step 5: convergence check to establish if the process should be iterated further or can be stopped. The number of iterations is controlled by " Maximum number of loops for Linear Programming process (LP)".

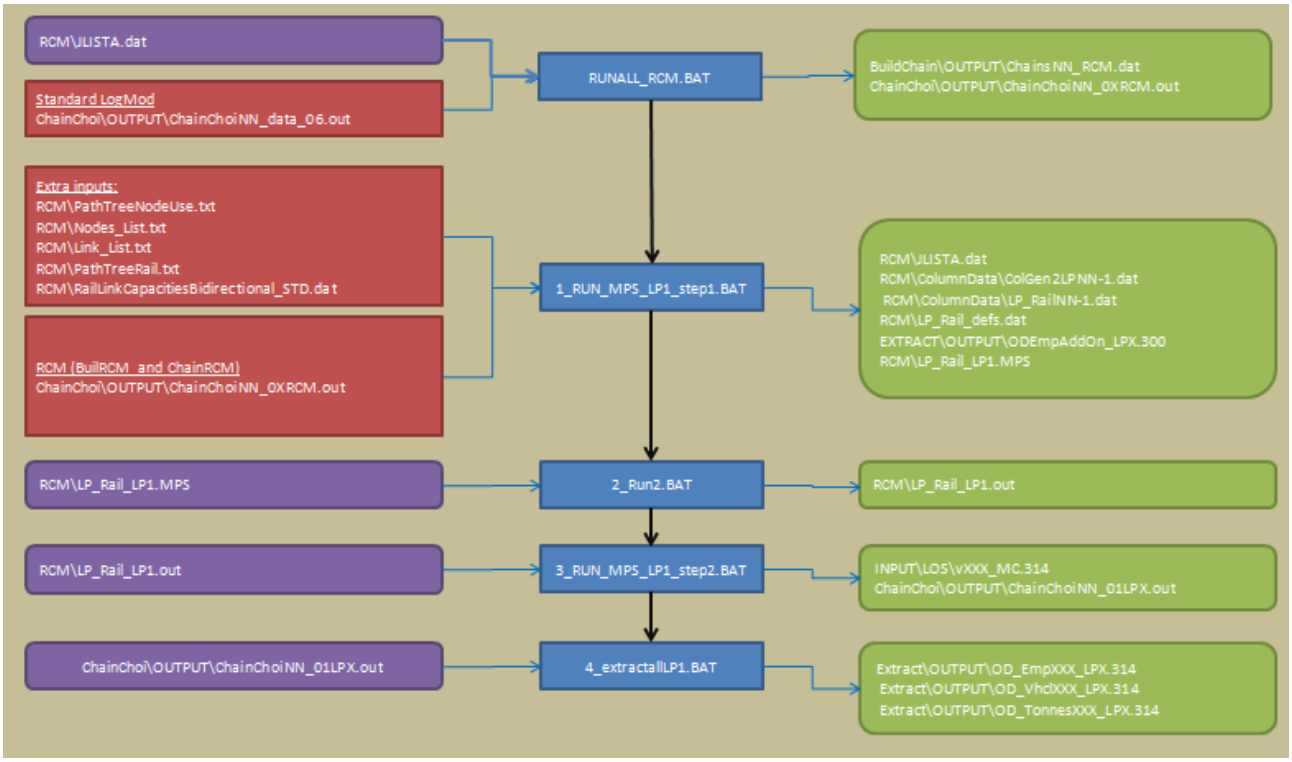


Figure 5 LP1 process: input and output data and processes involved.

The output files from the process

Output file	**Description**
ChainChoi\OUTPUT\ChainChoiNN_01RCM.out	Transport chains in column generator process
RCM\ColumnData\ColGen2LPNN-2.dat	Intermediate file
RCM\ColumnData\LP_RailNN-2.dat	Intermediate file
EXTRACT\OUTPUT\ODEmpAddOn_LPX.300	Add-On factors for empty vehicles.
RCM\LP_Rail_LP1.MPS	Linear Programming problem in MPS format that will be solved by LP_Solve.exe program
RCM\LP_Rail_LP1.out	Results of linear programming problem solution in LP_Solve.exe format
INPUT\LOS\vXXX_MC.314 (over-writing previous files)	Marginal cost matrices for V201, V202, ..V209 (only rail mode).
ChainChoi\OUTPUT\ChainChoiNN_01LPX.out (over-writing previous files)	Transport chain solutions produced
Extract\OUTPUT\OD_EmpXXX_LPX.314	OD Empty vehicles for the rail mode
Extract\OUTPUT\OD_VhclXXX_LPX.314	OD Total vehicles for the rail mode
Extract\OUTPUT\OD_TonnesXXX_LPX.314	OD Tonnes for the rail mode

Table 121 –Output files from LP1

Adjust Capacity Procedure

The Adjust Capacity Procedure will handle the cases where there will be remaining over-loaded links, e.g. with an over-load at more than 1 train per day or 1% of the capacity level. In this process all the input data from LP1 process is kept unchanged except the capacity on the links. The capacity is reduced by a fraction specified via " Percentage of Overcapacity to be removed (over_capacity_factor):" catalog key where the number represent the percentage of the over-capacity flow that is be used for a reduction of the capacity value. The basic idea is to force the LP model to use less of the capacity to alleviate the over-load. For details about the function applied we refer to document listed under Section 0.6 Appendix D.

All the steps performed in LP1 are performed in Adjust Capacity Procedure with the following exceptions:

1) the column generator process is not performed at all

2) the input capacity file RCM\RailLinkCapacitiesBidirectional_LP1AdjC(1).dat is a capacity file with revised capacities (will change in each Adjust capacity loop accordingly the function described in document listed under Section 0.6 Appendix D)

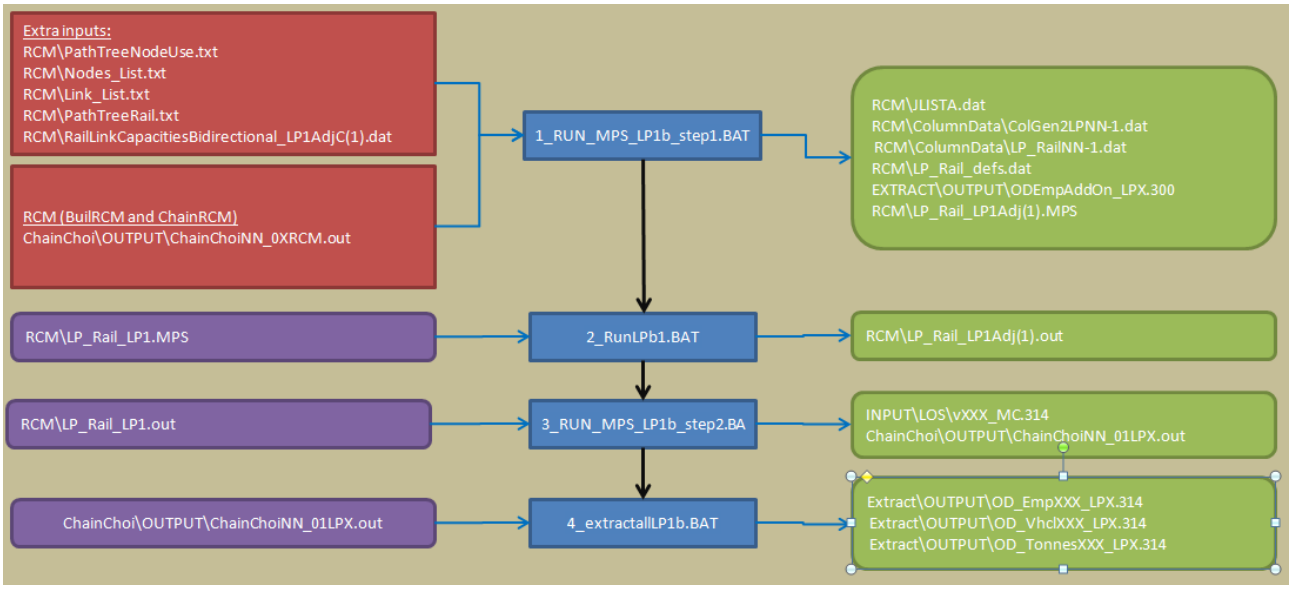


Figure 6 LP1b process (Adjust Capacity process): input and output data and processes involved.

Final process (FIN)

The final results of Rail Capacity Management saved in ChainCHoiNN_01LPX.out files are read and rewritten to the final solution files ChainCHoiNN_01FIN.out. In theory only the split solutions from the LP model should be modified in this process, but marginal changes in transport costs and total costs occur in many places. .

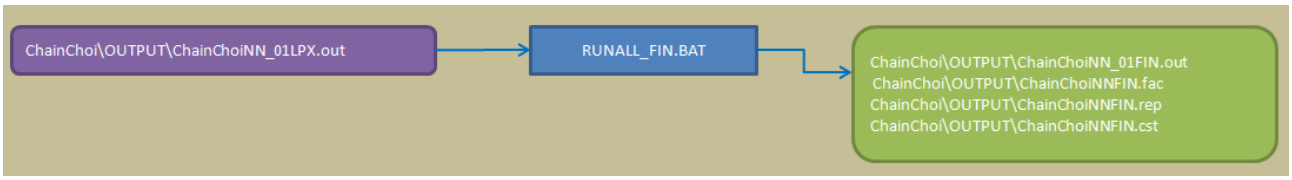


Figure 7 Final process: input and output data and processes involved.

The output files from the process

Output file	Description
ChainChoi\OUTPUT\ChainChoiNN_01FIN.out	Final transport chain solutions with STD Logmod format
ChainChoi\OUTPUT\ChainChoiNNFIN.fac	Aggregate report on load factors for commodity NN
ChainChoi\OUTPUT\ChainChoiNNFIN.rep	Aggregate report on costs, tonne volumes, transport statistics etc for commodity NN
ChainChoi\OUTPUT\ChainChoiNNFIN.cst	detailed cost log for commodity NN for selected cases

Table 122 –Output files from FIN process

Rail Assignment

Capacity constraints

To distribute empty trains according to an assignment including options to distribute the empty train routes while considering generalized time, marginal cost and on top of that capacity constraints the following assignment model has been coded in Highway assignment:

$$\boldsymbol{min} \sum_{\boldsymbol{train\ types}} \sum_{\boldsymbol{links}} (\boldsymbol{regular\ costs}) * (\boldsymbol{empty\ trains\ flows}) + \sum_{\boldsymbol{capacities}} \boldsymbol{Penalty}(\boldsymbol{overload}) \quad (5.1)$$

where:

- Penalty(overload): an optional function for penalizing loading above capacity
- Costs: costs specified in vehicle parameters and network (operational costs, taxes, tolls and extra costs)
- Overload: total bidirectional trains minus bidirectional capacity

The penalty function used in the equation (5.1) is:

$$\boldsymbol{Penalty}(\boldsymbol{overload}) = \mathbf{10000} \times \left(\boldsymbol{max}\left[\mathbf{0}, \left(\sum_{\boldsymbol{traintypes}} \sum_{\boldsymbol{directions}} (\boldsymbol{empty\ and\ loaded\ train\ flows}) - \boldsymbol{capacity}\right]\right)^{\mathbf{2}} \quad (5.2)$$

The Assignment method described above is formulated as system optimization instead of user equilibrium. The first formulation is not applicable in Voyager scripting, therefore the optimal solution has been reformulated as:

INTEGRAL(0,Q) t(q)dq = $[T(q)]_0^Q$ = T(Q) – T(0) = T(Q).

where:

- t(q) = vd-function
- T(q) = the primitive function of the vd-function

Thus, by using as vd-function the derivative of the penalty function:

$$\boldsymbol{Vdf(overload)} = \boldsymbol{2 \times 10000 \times \left(max\left[0, \left(\sum_{traintypes} \sum_{directions} (empty\ and\ loaded\ train\ flows) - Capacity\right]\right)\right.}$$

(5.3)

and integrating over flows from 0 to Q (=all flows on the link), we obtain an objective value equal to the system optimal value for the penalty function term.

The proposed assignment method has been implemented in the highway program via the following steps:

- the bidirectional volumes for each rail link with capacity are stored in one array where the dimension is the ID_LINK from capacity table
- in the adjust procedure, when the times are corrected based on the vdf functions, two functions are applied:
    - for all rail links that have no capacity (e.g. 0) the fix time derived from vehicle parameters is applied
    - for all rail links with specified capacity, the bidirectional flows stored in the array are retrieved and compared to the capacity, if overcapacity the vdf function specified in (5.3) is applied
- the volumes are averaged along the iterations using the MSA algorithm (CONVERGENCE=AVE).

Domestic flows allowed only on domestic links

To avoid unrealistic routes for domestic flows (e.g. where the origin and destination are inside Sweden and foreign railway links are used for empty flows) a designated procedure has been put in place. The procedure works in the following manner:

- a dummy matrix containing information if an OD is domestic or not is used to filter the demand matrix
- two different classes (domestic flows and all the others) are provided to the assignment
- in the LINKREAD phase (when all the variables for the assignment are specified) an IF statement is used to add to group 10 all the non-domestic links (similar technique used also in Kiel Canal)
- in the ILOOP phase (when all the possible paths are constructed) two different sets of paths are produced:
    - the first set has the input matrix containing only the domestic flows and the links on group 10 are excluded (using EXCLUDEGROUP=10) statement in PATHLOAD commend
    - the second set uses the matrix with all the other flows and all the links are open in the network (no EXCLUDEGROUP is coded for 10).

Calibration procedures

Forbidden over-seas ship transports to domestic small ports

To prevent over-seas transport to enter selected small ports in Sweden in the LOS matrices the combination of OD from/to transoceanic ports is entered with a high impedance.

The following changes have been made in the model:

- The table "PortNodes" has been updated with a new field: TransOceanFac. Ports having the value equals 10 are small ports in Sweden (74 ports)
- In the Sea application, under LOS calculation, additional logic has been inserted in step 3 of the MATRIXprocedure . Its function is to read the PortNodes table to identify small ports and to read the Transoceanic factor. Furthermore, it sets a dummy matrix with scaling factors that will be applied in the next Highway step. The details of calculation are:

```
TransOceanL='1075-1099' ;range of zones that represent the transoceanic ports

JLOOP

OrigZone=Renumber(1,I) ; origin zone number in emme format

DestZone=Renumber(1,J) ; destination zone number in emme format

OrigFactor=TransOceanic(1,OrigZone) ; origin factor from port definition table (-1 means is not a port)

DestFactor=TransOceanic(1,DestZone) ; destination factor from port definition table (-1 means is not a port)



;not transoceanic

IF((INLIST(I,TransOceanL)=0)&&(INLIST(J,TransOceanL)=0)) ;if both the ports are domestic no factors are applied

MW[1]=1 ;no factor

ENDIF



IF((INLIST(I,TransOceanL)=1)||(INLIST(J,TransOceanL)=1)) ;if one of them is transoceanic the final factor is the multiplication of factor in origin and factor in destination

MW[1]=(1+OrigFactor)*(1+DestFactor)

ENDIF

ENDJLOOP
```

- In the Sea application, under LOS calculation, the HIGHWAY step 7 reads the dummy matrix produced with the previous step and applies the factor to the TIME skim:

```
MW[200]=MI.1.1 ;LOS factor

;apply los factor

MW[2]=MW[2]*MW[200]
```

Rescaling factors for LOS matrices in sea mode

The rescaling factors are produced by the cycling process and they are used to rescale the TIME skims applying a factor that varies per port area and STAN group.

Their implementation and calibration have required several steps in different parts of the model. They are:

- a revision of Nodes table including information of PortAreaNr
- a new set of output LOS folders, one for each STAN group, placed under LogMod\Input\LOS\S01 ..S12
- a new application under Logistics Module named " Port Area LOS inputs (STAN)". In this application external program MPS.jar is called to:
    - rescale the V301_timeh.314, ..., V321_timeh.314 LOS matrices and place the rescaled files under LOS\S01..S12 folders.

      The time value is rescaled for each OD applying the combined factor from port area in origin and destination and STAN group. An example is shown in the following table:

Orig	Dest	TIMEH	Origin weighting factor (STAN1)	Destination weighting factor (STAN1)	Final value	Final TIMEH
718021	718121	10.067336	1.1	1.1	1.21 (1.1*1.1)	12.18148
718021	718221	0.637215	1.1	1.1	1.21	0.77103
718021	718821	9.59218	1.1	1.1	1.21	11.60654
718021	719221	7.884646	1.1	1.1	1.21	9.540422
718021	730521	12.73721	1.1	1	1.1	14.01093
718021	731921	11.457068	1.1	1	1.1	12.60277
718021	738221	9.061406	1.1	1	1.1	9.967547
718021	738222	9.066202	1.1	1	1.1	9.972822

Table 123 – Example on rescaling calculation for LOS sea mode.

    - the program uses the input files CalibrationParameters.txt placed under Input folder and collects the port area numbers from PortAreas_def.txt. The first is a copy of PortAreasParameters_22092014_L1_Base2006R21_RUN20.txt (it is the first loop of the calibration process) or the current values estimated in the previous loop (saved under scenario folder and named CalParameterNextLoop.txt)
    - copy the other LOS files
- Under the Parameters Calculation application the scaling factors are re-estimated by comparing the port statistics per port area and STAN group with model values. The values are adjusted based on the results. The function applied is as follow:

```
If (Abs(Diff) < 100) Then ; if the difference between modeled and statistics is less than 100 tons (DIFMAX)
 Adjust = factor    ; the previous value is maintained
endif
 If (Diff < 0) Then   ; if the difference is a negative value it means we have to reduce the factor (that is an impedance)
  Adjust = factor - 0.05 ; new factor = previous factor -0.05 (STEP)
  If (Adjust < 0.01) Then  ; if the adjust factor is less than 0.01 (MINVAL)
   Adjust = 0.05   ; is set to a minimum value 0.05 (VAL0)
  End If
 Else      ; if the difference is a positive value it means we have to increase the factor (that is an impedance)
  Adjust = factor + 0.05 ; new factor = previous factor+0.05 (STEP)
```

```
End If
End If
```

- The parameters that control the process are:
  - DIFMAX: maximum difference between modelled and statistics (100 means 100 tons)
  - STEP: the value that will use to increase the factor
  - MINVAL: boundary value to evaluate small or negative factors. if Adjust <= MINVAL then Adjust = VAL0
  - VAL0: reset value used if the function goes negative
- All of them could be revised modifying the table Parameters_portcalibration.dbf

ID	STEP	MINVAL	DIFMAX	VAL0
1	0.02	0.01	100	0.02

Table 124 Parameter to control the revision of scaling factors.

- The statistic used for the comparison is {CATALOG_DIR}\05_Input_Data\Input_Data.mdb\Port_statistics

The set of matrices per STAN group are provided to the Logistics Module and Rail Capacity Management procedure specifying a different LOS folder in the BuildChainXX.ctl and ChainChoi.ctl control files:

```
INCL=buildchain_common.ctl

COMMODITY=01

VHCL=..\Input\Cost\vhcls_dry_bulk.txt

LOSDIR=..\Input\LOS\S01

PWC=C:\Samgods_V10_150307\01_Programs\LogMod\Input\PWC\2030\PWC_01.txt
```

Above an example for commodity 01 is given

Rescaling factor for Kiel Canal

A similar procedure to the one described in the previous paragraph has been applied to toll on Kiel Canal.

Its implementation has been developed as follow:

- an initial scaling factor is provided using catalog key " Starting value for scaling factor on Kiel canal";
- under "LOS calculation\ Data preparation" the application Network step 17 takes this value and rescale the TOLL value for each vessel type

```
TOLL301=LI.1.TOLL301*parKiel(1,@LOOP_CAL@)

TOLL302=LI.1.TOLL302*parKiel(1,@LOOP_CAL@)

TOLL303=LI.1.TOLL303*parKiel(1,@LOOP_CAL@)

TOLL304=LI.1.TOLL304*parKiel(1,@LOOP_CAL@)

TOLL305=LI.1.TOLL305*parKiel(1,@LOOP_CAL@)
```

@LOOP_CAL@ represent the variable representing the current loop.

- a new set of output LOS matrices has produced rerunning the LOS calculation for Sea Mode
- those matrices are subsequently rescaled for the scaling factors per port area and STAN group
- Under Parameters Calculation application the scaling factor is re-estimated comparing the distribution of flows between Kiel Canal and North of Jylland (here is compared the percentage of the tons through Kiel Canal against those going north of Jylland) from the model results and statistics (present in {CATALOG_DIR}\05_Input_Data\Input_Data.mdb\Other_statistics table) The function applied is as follow:

```
If (Abs(Diff) < 100) Then ; if the difference between modelled and statistics is less than 100 tons (DIFMAX)
 Adjust = factor    ; the previous value is maintained
endif
 If (Diff < 0) Then   ; if the difference is a negative value it means we have to reduce the factor (that is an impedance)
  Adjust = factor - 0.05 ; new factor = previous factor -0.05 (STEP)
  If (Adjust < 0.01) Then  ; if the adjust factor is less than 0.01 (MINVAL)
   Adjust = 0.05   ; is set to a minimum value 0.05 (VAL0)
  End If
 Else     ; if the difference is a positive value it means we have to increase the factor (that is an impedance)
  Adjust = factor + 0.05 ; new factor = previous factor+0.05 (STEP)
 End If
End If
```

  - The parameters that control the process are:
    - DIFMAX: maximum difference between modelled and statistics (2 means 2%)
    - STEP: the value that will use to increase the factor
    - MINVAL: cut off value to evaluate small or negative factors
    - VAL0: reset value used if the function goes negative
  - All of them could be revised modifying the table Parameters_kielcalibration.dbf

ID	STEP	MINVAL	DIFMAX	VAL0
1	0.02	0.01	2	0.02

Table 125 Parameter to control the revision of scaling factor.

  - The statistic used for the comparison is {CATALOG_DIR}\05_Input_Data\Input_Data.mdb\Other_statistics table

Samgods Parallelization Module

The purpose of the Samgods Parallelization Module is to speed up an execution process by using parallelization. This is done by splitting up the workload of two or several independent program calls from running on one processor in a sequence to run on several processors in parallel. The module calculates the most even amount of workload per processor then feeds the operating system (OS) the divided workload in batch files and starts the execution. In the end of the process the module returns after having checked that all steps in the original program call sequence has finished.

The default input to the program is three input files with fixed file names: “execution_times.txt", "runall.bat" and "nr_of_program_instances.txt”. The file name runall.bat may be altered and overrun with the first expression in the parameter list of the module.

The input files

- “execution_times.txt”: This file gives the information the module needs to calculate the workload per program and the workload per processor. The input in this file has two pieces of information per row.

One is a reference to the program call to be run and the other one is the estimated runtime for that program to finish. Semicolon is used as a delimiter.

- "runall.bat": This file contains the sequence of program calls to be executed. The parallelization module interprets two keywords namely *call* and *rem*. If the line starts with *call* the module reads the line and if the line starts with *rem* the module jumps to the next line. After the word *call* the name to the file to be called by the OS is found and lastly the parameter passed to the file to be called (this is the reference number to the execution times mentioned above.) It may look something like this: call commodity 10

- "nr_of_program_instances.txt": This file contains one number that tells the module how many parts the full workload shall be divided into. It is usually good practice to set this number to less or equal the number of processors available for execution. (One limiting factor may also be the available memory.)

Controlling that the execution has finished

After having read the input files and calculated the most even workload per processor, the module writes batch files that are associated with a corresponding stop file. Each batch file contains the calls that where originally found in "runall.bat". The last line in each batch file has a command to delete the corresponding stop file. For the module to know that all the program steps have finished, checks are made if any stop file exists every 20 seconds. If no stop file exists, then all steps have been run and the module returns.

Programs and licenses required

Programs

Cube software

For the Samgods GUI to function, certain Cube software is required. They are:

- Cube Base: 6.1.0 service pack 1
- Cube Voyager: 6.1.0 service pack 1
- Cube GIS: ArcGIS 10.1 service pack 1

This software is available at ftp://citilabsftp.com/release/cube610SP1setup.exe.

The installation also require some bespoke programs developed specifically for Samgods 1.0. Them are available by contacting Petter Hill at Trafikverket

JAVA software

Java runtime environment (jre). Platform: 1.6. Product: 1.6.0_17 (later program versions are also possible to use)

Location for the above programs: http://www.java.com/sv/download/

Java virtual machine Windows x64

Location for the above programs: http://www.oracle.com/technetwork/java/javase/downloads/java-se-jre-7-download-432155.html

http://download.oracle.com/otn-pub/java/jdk/7/jre-7-windows-x64.exe

LP Solve

Michel Berkelaar, Kjell Eikland and Peter Notebaert (2004): LPsolve citation data: Description : Open source (Mixed-Integer) Linear Programming system Language : Multi-platform, pure ANSI C / POSIX source code, Lex/Yacc based parsing Official name : lp_solve (alternatively lpsolve) Release data : Version 5.0.0.0 dated 1 May 2004 Co-developers : Michel Berkelaar, Kjell Eikland, Peter Notebaert License terms : GNU LGPL (Lesser General Public Licence)

Citation policy : General references as per LGPL

Roland Wunderling (1996), Paralleler und Objektorientierter Simplex-Algorithmus,ZIB technical report TR 96-09, Berlin 1996.

Bibtex-reference:

@PhdThesisWunderling1996,

author = Roland Wunderling

title = Paralleler und objektorientierter Simplex-Algorithmus

School = Technische Universit, Berlin

Year = 1996

Note = \url{http://www.zib.de/Publications/abstracts/TR-96-09/

Licence requirements

5.1.1. Cube software

For the above Cube software, Citilabs License 2013 is required (but newer software and license versions may be available later on).

5.1.2. JAVA software

No licences are required.

5.1.3. LP solve

No licences are required.

References

1. Bergström J: Documentation of mps.jar, Compose IT, 2015
2. de Jong and Baak, Documentation of LP2CC, BuildChain4RCM, ChainChoi4RCM Appendix, 2015
3. de Jong, G., Baak, Method Report - Logistics Model in the Swedish National Freight Model System (Version 2.1), Deliberable 6B for Trafikverket, March 2014
4. de Bok, M., Baak, J., de Jong, G., Program documentation for the logistics model for Sweden, March 2014
5. Edwards, H**.**, Swedish Base Matrices Report. Estimates for 2004, estimation methodology, data, and procedures. March 2008
6. Edwards H: Railway Capacity Management for Samgods Using Linear Programming, Sweco 2015-02-05.
7. Sala G. Romanò D. : Samgods User Manual Ver 1.0, V&B Software Services, 2015
8. Vierth, I., Lord, N., McDaniel, J., Representation of the Swedish transport and logistics system (Logistics Model 2.00), VTI 2009
9. Wunderling R., Paralleler und Objektorientierter Simplex-Algorithmus, ZIB technical report TR 96-09, Berlin 1996.

Programs and licenses required

TRAFIKVERKET

Trafikverket, Box 388, 831 25 Östersund. Besöksadress: Kyrkgatan 43 B
Telefon: 0771-921 921, Texttelefon: 010-123 99 97

www.trafikverket.se