TRAFIKVERKET

SelectDirect Samgods 1.2.3

Trafikverket
Trafikverket, Box 388, 831 25 Östersund. Besöksadress: Kyrkgatan 43 B.
Telefon: 0771-921 921, Texttelefon: 010-123 99 97

Författare: Samgods arbetsgrupp inom avdelning Transportsystemutveckling vid Trafikverket
Dokumentdatum: 2026-05-04

Kontaktperson: Petter Hill, Trafikprognoser

Content

SelectDirect 4
What does it do? 4
Main use, network flows for GIS presentation 4
Main (initial) purpose, selected transport flows 4
Auxiliary functions 5
Program control 5
Command line 5
Control file 6
Program structure 9
Main classes 9
Utility classes 9
Changes in SelectDirect 1.2.1 and 1.2.2 10
Changes in SelectDirect 1.2 10
New features 10
Changes in functionality 10
Enhancements 10
References 10
Appendix – Output data description 11
LinksSumSL.dat 11
ConSumSL.dat 11
ChainsSL.dat 12
ConnectionsSL.dat 13
TotalFlows.dat 14

SelectDirect

SelectDirect.jar is a Java program that performs select link analysis.

What does it do?

SelectDirect uses spanning tree data to resolve transport flows to link and node level for various purposes. It also carries out some auxiliary tasks of aggregating traffic flows, and preparing files for further use in the Samgods model.

Main use, network flows for GIS presentation

The capabilities of SelectDirect are used in each run of a scenario in Samgods for the following purposes:

1. To compress spanning tree data to a more compact format, resulting in smaller input files and faster input of spanning tree data.
2. To derive network flows per bidirectional link split by commodity into ktonnes and number of vehicles per day (default 250 days per year) for GIS presentation after STD logmod
3. Same as 2 after RCM (Railway Capacity Management, XTD logmod)
4. To derive results for CBA analysis.

Main (initial) purpose, selected transport flows

The main purpose of SelectDirect is to generate input data for visualization of selected transport flows. For this analysis the program generates five output files. Each contains flows that fulfil the selection criteria given in the program control file.

- LINKSFILE (LinksSumSL.dat) - Total flow per link and vehicle type.
- CONSUMFILE (ConSumSL.dat) - Total flow per connection and vehicle type.
- CHAINSFILE (ChainsSL.dat) - Flow per each transport chain.
- CONNECTIONSFILE (ConnectionsSL.dat) - Flow and distance for each connection in each transport chain.
- TotalFlows.dat Total flow per vehicle type.

Two different types of criteria can be used:

- Select link analysis: Transports that pass certain nodes or links.
- Select zone analysis: Transports that begins in a selected interval of zones and end in another selected interval of zones.

For the first case links and / or nodes are given by keywords LINKSSL and NODESSL. If neither of these keywords are used, all transport chains for the selected commodities will be included. Output include transports that pass/starts at/ends at appointed combinations of links/nodes – a ***Selection*** set. The ***Selection*** can include any or all of the given nodes and links and can be required to be passed either in the order given in the keywords or in random order. These visit conditions are defined by the keywords VISITNODE and VISITLINK.

For the second case start zones are given by keyword OZONE, and end zones by DZONE.

Auxiliary functions

Traversal matrices
Using this option the program can produces so called traversal matrices, files specifying flows to and from the Swedish border zones.

GIS data
This option, STAN, produces a file that contains flows for each link and commodity as well as number of trains and used capacity for capacitated rail links. The name STAN is a reminiscence of a previous model version where it constituted a grouping of 35 commodities into 12 STAN-groups. The output file is named after the keyword STANFILE. An associated output file holds aggregated rail results, named after the keyword STANSUM. Examples for XTD logmod:

STANFILE=RCM\Output\Commodity_XTD.dat

STANSUM=RCM\output\RAIL_TONKM_VKM_SUM_XTD.DAT

Domestic ratio
For this option SelectDirect loops over all time and cost LOS matrices, and uses the resolved path together with time and cost data per link to produce new LOS matrices containing the domestic part of time and cost, respectively.

Compact Spanning Tree files
This option produces Spanning Tree files on a compact format, reducing the required storage space with approximately 90%.

Cost for empty vehicles
This function uses LOS data and vehicle cost data to produce a cost file for empty vehicles.

CBA sum file
Generates a file containing the total cost for all transports divided into the categories Domestic, Import, Export, and Transit transports. Also sums the total vehicle kms for these categories.

Program control

Command line

SelectDirect is a Java command line program. It is run by commands as shown below:

For the main usages 1-4 we have the following four alternatives:

```
java -d64 -Xmx4g -Xms2024m -jar SelectDirect.JAR Compact.ctl PackSpt
java -d64 -Xmx4g -Xms2024m -jar SelectDirect.JAR COMMODITY_STD.ctl STAN
java -d64 -Xmx4g -Xms2024m -jar SelectDirect.JAR COMMODITY_XTD.ctl STAN
java -d64 -Xmx4g -Xms2024m -jar SelectDirect.JAR COMMODITY_CBA.ctl STAN
```

For the actual select link function the main command is**:**

```
java -d64 -Xmx4g -Xms2024m -jar SelectDirect.JAR SelectDirect.ctl slnk
```

Command line parameters:

The task that the program is expected to carry out is defined by a command line argument. The arguments used in all scenario runs are `PackSpt` and `STAN` as shown above.

To run a select link analysis use argument `slnk` or leave out as this is the default action.

Control file

Each command requires a control file such as *Compact.ctl, COMMODITY_STD.CTL*, etc with parameters described below for the program run.

The last example of command lines with the control file *SelectDirect.ctl* represents the standard command for a select link run. The control file holds the necessary number of parameters for the program run. These are described below. The control file can contain comments. Those are rows starting with a hyphen (-).

Control file parameters *(examples in italic)*:

Required parameters are marked with black text.

Optional parameters are marked with blue text.

Parameters (with example values)	**Description**
ASEKLOS=Input_CBA\LOS	Filepath for ASEK LOS-matrices.
CBASUMFILE=RCM\Output\CBA_aggr.txt	Filepath for CBA summary output file.
CAPLINKS=RCM\RailLinkCapacitiesBidirectional_STD.DAT	List of links with Emme/Voyager numbering system and bidirectional capacity per day.
CCFILES=ChainChoi\OUTPUT\ChainChoi[v]XTD.out	Filepath for ChainChoiXTD files.
CHAINSFILE=RCM\OUTPUT\ChainsSL.dat	Filepath for ChainsSL output files.
COMMODITIESSL=1,2,3	All commodities from default value of "MAX_CMD" is checked. In Samgods 1.2 this value is 16.
CONNECTIONSFILE=RCM\OUTPUT\ConnectionsSL.dat	Filepath for ConnectionsSL files.
CONSUMFILE=RCM\OUTPUT\ConCumSL.dat	Filepath for output file for CONSUM.
COSTFILES=ChainChoi\Output\ChainChoi01data07XTD.out	Filepath for cost files to use.
DFR=200	Day factor rail.
DZONE	Destination zone.
EMPTYCOST=RCM\Output\EmptyCost.dat	Filepath to EmptyCost output file.
EMPTYFRAC=Extract\emptyfrac.dat	Filepath to emptyfraction file.
ERRORLOG=RCM\Error.log	Errors are logged to this file. If the ERRORLOG parameter is omitted the file is called Error.log and placed in the folder specified by the PATH

	parameter. If the keyword ERRORLOG is used the log is rewritten at the beginning of the J step.
IO_LOG=RCM\UTI_CBA.log	The line IO_LOG=RCM\IOLog_LP[i].log in the ctl file will cause all names of input and output files to be written to file, together with ctl settings.
LINKLIST=RCM\Links_List.txt	Input- list of links in the network with ID link followed by Voyager from and to nodes and Emme from and to nodes respectively. Blank or comma separated.
LINKSFILE=RCM\OUTPUT\LinksSumSL.dat	Filepath for LinksSumSL output file.
LINKSSL=550700-550652	Comma separated list with links to check. May also include a span (for example 333111-333222).
LINKTIMECOST=RCM\Exported_network_BS2012.txt	Filepath for exported network.
LOCKS=STD *LOCKS=RCM\LockedSTDLogMod_Soln.txt*	LOCKS parameter can take two types of values, either a filename, or the value STD. If the value is STD, locked chains are read from the STDLogMod output files LockedXX.log in folder ChainChoi\OUTPUT. The parameter is mandatory if locked flows are present.
LOSPATH=Input\LOS	Filepath for LOS-matrices.
MAX_ARRPATHSIZE=2000	Sets the maximum size of the best-path array. Default value is 2000.
MAX_CAPLINKS=1000	Number of rows from CAPLINKS file. Default 1000. Used by the class SpanningTree.
MAX_CMD=16	Sets the highest commodity number. Default 16.
MAX_EMMESPAN=300000	Maximum size of the span of Emme zones. Default 300 000 (700000 – 999999). Used by the class SpanningTree.
MAX_INTLINKNO=150000	Maximum internal link number in LINKLIST file. Default 150 000. Used by the class SpanningTree.
MAX_JKEYS=500000	Maximum number of Superindexes in the JLIST file per commodity. Default 500 000. Used in the methods Write_CG_LP_Itr0 and Write_CG_LP_ItrX. If either of these methods fail with an ArrayIndexOutOfBounds error increasing this parameter may fix the problem.
MAX_POSSOLS=1500000	Maximum number of positive solutions in the Linear Programming output. Used when writing LPX files in iterations 1 or higher. Default 1 500 000. If the program fails with a message of ArrayIndexOutOfBounds in method WriteLPX increasing this parameter may fix the problem.
MAX_SPANNODE=2000	Span of nodes in Spanning Tree. Default 2000. Used by the class SpanningTree.

MAX_VOYNODE=50000	Maximum Voyager node number in the NODESLIST file. Default 50 000. Used by the class SpanningTree.
NODESLIST=RCM\Nodes_List.txt	Input- List of nodes in the network with Emme and Voyager numbers. Blank or comma separated.
NODESSL=321724,321725	Comma separated list containing border nodes.
OZONE	Origin zone.
PATH=C:\workfolder	Filepath for default working directory.
SELECTCOMBINE=3	1-4. 1 = keepnode. 2 = keeplink. 3 = keepnode OR keeplink. 4 = keepnode AND keeplink.
SL_EMPTY=EXTRACT\OUTPUT\OD_Emp[f]_FIN.314	Filepath for OD empty files.
SLCTLINKNETFILE=RCM\OUTPUT\SelectLink_Netw_Flow.dat	Filepath for output file of link network flows.
SLCTLINKSUMMARY=RCM\\OUTPUT\\Link_TonLoadEmp.dat	Filepath for output file of link summary by ton, loaded, empties.
SPT_AIR=RCM\PathTreeNNNN.txt	Path to spanningtree input for mode NNN for compacting the file. Output will be *RCM\PathTreeNNN.cmp*. Modes are *Air, Rail, Road and Sea* respectively
SPT_AIR=RCM\PathTreeNNNN.cmp	Path to spanningtree input (compact format) for mode NNN. Modes are *Air, Rail, Road and Sea* respectively
STANFILE=RCM\Output\Commodity_STD.dat	Filepath for output report containing network flows by commodity after STD logmod
STANFILE=RCM\Output\Commodity_XTD.dat	Filepath for output report containing network flows by commodity after RCM (XTD logmod)
STANSUM=RCM\output\RAIL_TONKM_VKM_SUM_STD.DAT	Filepath for output summary report containing tonnes, load and empties for rail by commodity. This one is for STD logmod. XTD is used for results after RCM.
TM_EMPTY=EXTRACT\OUTPUT\OD_Emp[f]_FIN.314	Filepath for input data for empties used to generate traversal matrices from.
TM_TONNES=EXTRACT\OUTPUT\OD_Tonnes[f]_FIN.314	Filepath for input data for tonnes used to generate traversal matrices from.
TM_VHCL=EXTRACT\OUTPUT\OD_Vhcl[f]_FIN.314	Filepath for input data for vehicle load used to generate traversal matrices from.
TRAVERSALPATH=TRAVERSALFOLDER	Filepath for traversal matrices.
TrpCostLE_DXMT_Input_NNN.dat	Summary of results per vehicle type, split into domestic, export, import and transport flows. For results after step NNN, where NNN is STD, XTD or CBA.

VEHICLES=201,202,203,204,205,206,207,208,209	List of all train types in the ChainChoi output files. The above list is the default, the keyword is needed only if new train types are added. A current list is VEHICLES=201,202,203,204,205,206,207,208,209,210,211,212 The number of train types in this parameter is also used to determine the number of train types the program handles, and to allocate variables for this number of train types.
VHCLCOSTPATH=Input_CBA\COST\VHCLS_COM01.txt	Path to input file for vechicle costs.
VHCLREP=RCM\Output\VhclRep.dat	Filepath for vhcl report output file.
VISITLINK=ANY	Value that sets whether a link should be checked or not.
VISITNODE=ORDER	Available options: ANY, ALL, ORDER, SINGLE. ANY = any of the given nodes or links must be used. ALL = all of the given nodes and links must be used. ORDER = checks order of nodes. SINGLE = indicates that nodes must be passed in the order given by the NODESSL parameter.

Program structure

The program is originally developed in Eclipse Java IDE. The program was converted to Netbeans IDE for Samgods 1.2.

Main classes

SelectDirect.java
This class holds the java main procedure. It reads the control file and calls a procedure for the requested analysis.

SPT_Select.java
Reads and contains data for the spanning tree. Returns the path for a given connection.

Utility classes

CtlReader.java
A utility class that reads and holds the values from the control file.

Enums.java *
A utility class that describes array position values.

Funcs.java

A utility class that has various functions for string handling etc.

Funcs_CCSlctLink.java

A utility class that holds functions for empty calculation based on chainchoi input.

Funcs_CommoditySUM.java

A utility class that holds functions for creating the data by commodity.

Funcs_EmptySlctLink.java

A utility class that holds functions for empty calculation based on selected link input.

LgFileReader.java
A standard FileReader with the additional function that it logs the name of the file read.

LgFileWriter.java
A standard FileWriter with the additional function that it logs the name of the file written.

Lists.java
Lists keeps track of various input data.

Changes in SelectDirect 1.2.1 and 1.2.2

- Added new output tables (*VehicleTrpEmpInv_Input_STD/ XTD/CBA.dat*) containing data needed for CBA.
- Improved verification of input to select link-functionality.
- Bug fix related to empty cost calculations (late 2023).

Changes in SelectDirect 1.2

New features

Added new report (*VehicleTrpEmpInv_Input_CBA.dat*) for summarizing tonne and vehicle kms, empty vehicle kms and vehicle costs. Results are presented per vehicle type, and split into domestic, export, import and transit relations on domestic links (links on Swedish territory. This report shows vehicle kms also for the ferries themselves, whereas the standard reports show the vehicles on the ferries. The path can be changed with the VHCLREP parameter.

Changes in functionality

Emptycost is now split between 4 categories: domestic, import, export, transit.

SelectDirect now handles 16 commodities instead of 35. The default value for parameter MAX_CMD has been changed from 35 to 16.

Enhancements

SelectDirect now handles null references and closing of streams in an improved faction, preventing memory leakage.

Improved logging to error-files.

References

Henrik Edwards (2015): Railway Capacity Management for Samgods Using Linear Programming

Appendix – Output data description

This is a description of the Java program SelectDirect output data. The purpose with this text is to describe the output data, thus making the program and data more user-friendly and allow the user to gain a greater understanding. The objective you as a user should have when reading this text is to find out what the tables contain, and which variables are in each output file, an explanatory description and their data type.

The following five output files could be produced[1]:

- LinkSumSL.dat
- ConSumSL.dat
- ChainsSL.dat
- ConnectionsSL.dat
- TotalFlows.dat

LinksSumSL.dat

Each row relates to a link and vehicle combination. See Table 1.

Table 1 The LinksSumSL table with information about the variable name, the description and the data type.

Variable	**Description**	**Type**
Link	Index	Integer
FrVy	From-node (or start node) in Voyager numbering	Integer
ToVy	To-node (or end node) in Voyager numbering	Integer
FrEMME2	From-node in Emme numbering	Integer
ToEMME2	To-node in Emme numbering	Integer
Vhcl	Vehicle number, 3 digit format	Integer
kton/year	Link volume in kton per year	Double
NLoadVhcl/day	Link flow in number of loaded vehicles per day (250 days per year)	Double
NEmptyVhcl/day	Number of empty vehicles per day(derived from LOS-matrices[2]) (250 days per year)	Double

ConSumSL.dat

Each row relates to a connection and vehicle combination. Sums are derived from ConnectionsSL.dat. Please see the description of these variables in ConnectionsSL.dat. Variables Orig and Dest are centroid or terminal nodes. See Table 2.

[1] The exact names of the files are actually handled in a control file, thus dependent on the names given by the keywords in the control file. The names refer to parameters LINKSFILE, CONSUMFILE, CHAINSFILE and CONNECTIONSFILE (TotalFlows.dat hard-coded).

[2] LOS-matrices given by parameter SL_EMPTY, value found by the key FrEMME2, ToEMME2 and vhcl.

Table 2 The ConSumSL table with information about the variable name, the description and the data type.

Variable	Description	Type
Orig	Origin node in Emme numbering	Integer
Dest	Destination node in Emme numbering	Integer
Vhcl	Vehicle number, 3 digit format	Integer
N_Lvhcls	OD-connection sum number of loaded vehicles per day	Double
Tonnes	OD-connection sum tonnes per year	Double
N_EmptyV(O_D_L)	Number of empty vehicles (derived from LOS-matrices)	Double
TransportTimeCost_(SEK)	Not implemented	NA
LoadingTimeCost_(SEK)	Not implemented	NA
InterestCost_(SEK)	Not implemented	NA
DistCost_(SEK)	Not implemented	NA
InfraCost_(SEK)	Not implemented	NA
LoadingCost_(SEK)	Not implemented	NA

ChainsSL.dat

The table contains the best chain for a given shipment. Variables orig and dest are centroids or terminals. See Table 3.

Table 3 The ChainsSL table with information about the variable name, the description and the data type.

Variable	Description	Type
Cmd	Commodity	Integer
Key	Key for a specific shipment[3]	Integer
chaintype	Chain type	String
Orig	Origin node in Emme numbering	Integer

[3] An attempt to describe a shipment: a serial index to the combination P-C-F2Fcategory. The exact definition is found in the documentation concerning the logistic module.

Dest	Destination node in Emme numbering	Integer
Prob	The probability that the chain is the best chain	Double
Tonnes	Total volume (tonnes) per year (scaled by probability) [4]	Double
TrpCost_(SEK)	Not implemented	NA
AllCost_(SEK)	Not implemented	NA

ConnectionsSL.dat

Data is split by commodity. Each value for key consists of one (or multiple) legs, each leg with a submode. Variables orig and dest are centroids or terminals. The variables orig and dest are centroids or terminal nodes. See Table 4.

Table 4 The ConnectionsSL table with information about the variable name, the description and the data type.

Variable	**Description**	**Type**
Cmd	Commodity (numbers 1-16)	Integer
Key	Key for a specific shipment	Integer
Leg	Order for each leg in the transport chain (1,2,…,5)	Integer
Mode	Submode	String
Orig	Origin node in Emme numbering	Integer
Dest	Destination node in Emme numbering	Integer
Vhcl	Vehicle number, 3 digit format	Integer
Prob	Probabilty (share) of the demand that use this solution	Double
N_LVhcls	Number of loaded vehicles per year (weighted by probability)	Double
Tonnes	Total volume (tonnes) per year (weighted by probability)	Double
Dist	Domestic distance (from ddist LOS matrix)	Double
TransportTimeCost_(SEK)	Not implemented	NA
LoadingTimeCost_(SEK)	Not implemented	NA
InterestCost_(SEK)	Not implemented	NA

[4] From the file given by CCFILES parameter in control file, this is calculated as tonnes = Prob * NRelations * AnnualVolume_(Tonnes).

DistCost_(SEK)	Not implemented	NA
InfraCost_(SEK)	Not implemented	NA
LoadingCost_(SEK)	Not implemented	NA
PositioningCost_(SEK)	Not implemented	NA
FairwayDues_(SEK)	Not implemented	NA
PilotFees_(SEK)	Not implemented	NA
sel_nodes	Not implemented	NA

TotalFlows.dat

Values are derived from ConnectionsSL.dat. See Table 5.

Table 5 The TotalFlows table with information about the variable name, the description and the data type.

Variable	**Description**	**Type**
Vhcl	Vehicle number, 3 digit format	Integer
N_LVhcls_sumOD	Not implemented	NA
Tonnes_sumOD	Not implemented	NA

TRAFIKVERKET

Trafikverket, Box 388, 831 25 Östersund. Besöksadress: Kyrkgatan 43 B.
Telefon: 0771-921 921, Texttelefon: 010-123 99 97

www.trafikverket.se