EMU UTILITY DESCRIPTIONS

Table of Contents

Descriptions and Usage

1. Description: A general summary of the application and its use within the system. This description will also include reference to any additional information for a given utility, outside this document, as applicable.
2. Input: Where included, this section includes onscreen prompt input requirements (users enter as they go), and/or any files which are required for the application to run on any specific data file(s). Any file included under this heading is usually a fixed-format file that is read by the utility (hard coded) in question. While usually required to be located in a specific directory, files listed here are not required as part of the command line arguments (see Arguments section) unless otherwise specifically indicated.
3. Output: Where included, this section includes which files the utility generates and where those files are output – usually to a logical path, as specified in ELTransservice.cfg for utilities run standalone, and in Emusrv.inf for utilities run through EMU (Ex: TMP or UPD). This section can also include some information on contents of these output files as well..
4. Special Notes: Any special information or warnings will be included after this heading.

AddUpd.exe

Description: Updates any combination of the following student address location fields: "Prefix," "Street Name," "Type," "Suffix," and/or "MapZone." This change will be displayed in the "Location" field in the student dialog.

Output: TMP: <date/time>ADDUPD.AUD, <date/time>ADDUPD.CHK
EXPORT: ADDUPD.AUD

Special Notes: The input file Addupd.dat is generated and updated from the BuildTxt.exe application (located in the \elt\exe\report\ folder). Users should not normally edit the file manually. The utility updates the student file, but does not update stutrn or cmndst, and if a home stop is needed, it does not update the home stop either. User must run associated maintenance to complete these aspects.

AdScan.exe

Description: This is a student address matching utility. When a student record has been modified by Adscan, the utility makes sure that dependent trips and common destinations are appropriately updated.

Output: TMP: <date/time>ADSCAN.AUD, <date/time>ADSCAN.CHK, <date/time>ADSCAN.ERR, ADSCAN.ER2
EXPORT: ADSCAN.AUD, ADSCAN.ERR

Autostreeter.exe

Description: This utility extracts the geocode into the autostreets database. The layers created by this extraction are defined by using Autostreeter in standalone mode.

Output: GRF:autostreets.up,autostreets.nod,autostreets.stg, ,autostreets.pos,
autostreets.xys ,autostreets.lls, ,autostreets.mdb

Special Notes: Using EMU to update the Autostreets database is meant to be done in conjunction with the autostreet data transfer. EMU users will not be able to add/edit/change layers. That must be done using the stand alone flag on the server. ElBndPlan/ElPlot/ETC can update it's autostreets data from their servers. New .ini values ElBndPlan/ElPlot/ETC give the implementer the choice to update with either Ask|Always|Never. The 'Ask' option only asks the user if the data has changed. Standalone versions should choose Never and point DatabaseLocation to GRF. I favor the Always option for sites whose autostreets database is less than 2 megs and have reasonable networking. Always only updates when the data changes. The update occurs during program startup. UpdateData defaults to Never. ini values:
[AutoStreets]
UpdateData=Ask
DatabaseLocation=c:\elt\map\dat

BadStp.exe

Description: This utility lists stops which have something wrong with Node1/Node2/Distance1/ Distance2 or XY. It fixes stops which do not have correct values stored for Distance1 and Distance2.

Output: TMP: <date/time>BADSTP.AUD, <date/time>BADSTP.CHK, <date/time>BADSTP.ERR
EXPORT: BADSTP.AUD, BADSTP.ERR

Errors Reported in BadStp.err:

- No Nodes
- Invalid Nodes
- Invalid Segment
- Bad Distances
- Deleted home stop
- Changed home stop to non-home stop.
- Can't delete home stop - runs assigned

Can't delete home stop - students assigned
Can't change home stop - other students assigned
Non-travelable street segment.
Stop segment does not match x,y's
Right side pickup on one way street
Segment stop on Hazard Segment
Node stop with Node on Hazard Segment
Invalid Nodes - stop fixed
Invalid Segment - stop fixed

Badstu.exe

Description: Report of students which have something wrong with Node1/Node2/Distance1/Distance2/XY or Sch/Grd/Prg. This utility also will fix students which do not have correct values stored for Distance1 and Distance2.

Output: TMP: <date/time>BADSTU.AUD, <date/time>BADSTU.CHK, <date/time>BADSTU.ERR
EXPORT: BADSTU.AUD, BADSTU.ERR

Errors Reported in BadStu.err:
No Nodes
Invalid Nodes
Invalid Segment
Bad Distances
Sch/Grd/Prg Err
Stop segment does not match x,y's
Invalid Nodes - student fixed
Invalid Segment - student fixed
ZXYADR Error

BatchEchoStops.exe

Description: Echo stops from a worklist to a selected school or cluster.

Output: TMP: BatchEchoStops.aud, BatchEchoStops.err

Special Notes: Schools stops (xxx.000) will not be echoed, and such entries will be listed in BatchEchoStops.err.

BatchRtedir.exe

Description: Batch process run directions for all the runs on all routes or all the runs on all the routes in a list.

Output: TMP: <date/time>BthRtedir.aud

BatchRundir.exe

Description: Batch generate run directions for all runs or for just a list of runs.

Output: TMP: <date/time>BthRundir.aud

Bthasgn.exe

Description: Batch assignment of student's trips to stop services.

Output: TMP: <date/time>BTHASG.CHK, <date/time>BTHASG.AUD, <date/time>BTHASG.ERR
EXPORT: BTHASG.AUD, BTHASG.ERR

Special Notes: For Codes1 (Valid Elg Codes) – Bthasgn will use this list in place of the standard valid codes: 0 – 6, not in addition to them, so '0' and '1' will need to be included in the list for those students to be assigned. If no elg codes are supplied then it defaults to the standard values. For Codes2 (Selected Trip Types) – Bthasgn will use the trip types entered (usually a value from 1 to 4) here in lieu of the two default trip types (1 and 2), unless no values are included, in which case it will use the default trips. For both sets of code arguments, the comma delimiter is required for each code added to the argument, and NO SPACES are added between each (Ex: 1,2,3,4 instead of 1, 2, 3, 4) – if spaces are added between the comma's, the utility will not run correctly.

BthElg.exe

Description: Batch calculation of student eligibility based on posted boundaries for schools and student locations in the geocode. Note that students must be matched to the geocode in order for this utility to calculate their eligibility. User-defined eligibility codes can be generated automatically as well, if users select the option to update user-defined elg codes to match system defined elg codes.

Output: TMP: <date/time>BTHELG.AUD, <date/time>BTHELG.CHK, <date/time>BTHELG.ERR
EXPORT: BTHELG.AUD, BTHELG.ERR

CalculateTimeatStop.exe

Description: This utility processes all non-transportation mode runs and calculates time at stop for each stop on the run based on current time next, and loading time values. It will also update the run direction times at stop as needed. This utility must be run after LodTme.exe or SpedLodTme.exe.

CreateTrips3and4.exe

Description: This utility will create trips of type 3 and 4 for selected students, if the students don't have any existing trips of types 3 or 4.

Special Notes: For the system to build trips of types 3 or 4, users must supply enough information for the system to build the destination part of the trip, and make the appropriate service assignment. To do this, the user must supply the school and the bell time. The school is Flag3. Flag5 and Flag7 allow the user to choose whether the bell time will be entered directly by the user, or will be computed based on the entered school, and the student's grade and program. Users also have two choices about what they want in the origin part of the trip. Flag2 allows them to choose either the student's home location or the student's transport school location as the origin location. This part of the trip will not be automatically assigned.

DateRollOver.exe

Description: This utility changes frequency dates stored in student trips, stops, stop services, runs, runtrn, route, rtetrn, and garages. If the data contains 100% consistent date/freq combo's throughout, the utility will update the Start Date and End date values for all dependent data in the dataset, and update STA:ELOptions.dat as well. Editing sysdef.inf is no longer required – the system no longer uses the Start and End date values in sysdef.inf.

Output: TMP: <date/time>DATEROLLOVER.ERR

Special Notes: The dummy value forces the existing batch configurations on client machines to be updated – by not allowing any calls with only 4 parameter/arguments to take place. If the dataset does not have consistent date/freq values throughout, the utility will not run – user will need to run Fixdates.exe first, to clean-up these inconsistencies.

Deasgn.exe

Description: Batch de-assignment of stop services from student trips. This can be done using a student list and/or a stop/service list, or all students on all stops, or using a trip list.

Output: TMP: <date/time>DEASGN.AUD, <date/time>DEASGN.CHK
EXPORT: DEASGN.AUD

Special Notes: Works with trip and service lists as well as student and stop lists. If trip list is used then no other list can be used. A student list can be used with or without a stop or service list.

DeassignCheckPoints.exe

Description: This allows the user to deassign checkpoints onto runs en masse from all or a worklist of routes or runs.

DeassignRunsOnRoutes.exe

Description: Batch de-assigns runs from routes.

Output: TMP: <date/time>DEASSIGNRUNSONROUTES.AUD

DeassignStopsOnRuns.exe

Description: De-assigns non-school and non-transfer stops from runs. Users can also specify whether they want checkpoint stops de-assigned or not.

Output: TMP: <date/time>DEASSIGNSTOPSONRUNS.AUD

Special Notes: If the Optional Flag3 is missing, then the program defaults to not de-assigning the checkpoints.

DeleteRoutes.exe

Description: Goes through all the routes in the system checking for routes with no runs assigned. The utility purges the route record of any routes with no runs.

DeleteRunsNoStops.exe

Description: This utility will always purge non-deadhead, non-garage, and non-transportation mode runs that do not have any services assigned to them. Users have the option to also purge runs with only school services. If a run to be purged is on a route then it will de-assign them from the route before purging the run. If there are only garage runs left for that time of day, it will purge the garage run pair before de-assigning the run from the route.

Output: TMP: <date/time>DELETERUNSNOSTOPS.AUD

DeleteUnusedStreets.exe

Description: This utility deletes street name records from the geographic database which no longer have any associated segment or landmark data.

Output: TMP:DELETEUNUSEDSTREETS.RPT

DelInvalidSchSrv.exe

Description: Purges school services that do not match a bell time and are not assigned to runs or trips.

Diagnose.exe

Description: Allows users to diagnose their transportation data, and reports on any invalid data found.

Output: Refer to full documentation in Diagnose-Use.doc

Dumpall.exe

Description: Dumps all transportation data into DBF files used by FoxPro based applications.

Output: elt\tables*.dbf, to many to list
EXPORT: studntup.dbf, settrn.err. If errors nttofoxwrite.err and/or error.txt

DumpSchdst.exe

Description: Dumps school to school distance into DBF files used by FoxPro based applications.

Output: elt\tables\schdst.dbf

Special Notes: This utility will dump the schools and distances between into a fox file called schdst.dbf. This then allows the foxpro program SCDSRP to report on school to school distances.

FileFull.exe

Description: Reports on the total number of records used, the last record used, the number of empty records and number of deleted records for any transportation file the user chooses and the characters the user file is built for.

Output: TMP:FILEFULL.AUD
TMP:DSTUDNT.AUD

Flagset.exe

Description: Set/Reset student deletion, address match, eligibility, Upstu exclusion, special ed, home stop, availability exclusive, right side, add, change, and assign flags (and/or set deletion flags from a specified date).

InsertCheckPoints.exe

Description: This allows the user to insert checkpoints onto runs en masse from all or a worklist of routes or runs. If a checkpoint already exists at that location it will deassign the current checkpoint and insert the new checkpoint. The location of the checkpoint must be at a current stop or matchable location (see notes) . If inserting checkpoints using routes and a garage runs exists in that location then a checkpoint it not added .

Special Notes: If Location is not at a specific stop then the string needs to be preceded with '>'. In addition if the checkpoint is to be located at a specific node, the node number should be preceded with '#'.

KilStp.exe

Description: Delete services which have no load and are not assigned to any runs, and the option to also delete stops which have no services.

Output: TMP: <date/time>KILSTP.AUD

LodTme.exe

Description: Updates the loading time field in a runtrn record (service on a run) based on input from the user.

Output: TMP: <date/time>LODTME.AUD, <date/time>LODTME.CHK
EXPORT: LODTME.AUD

Special Notes: Either the Regular flag or the Special Ed flag must be used. If both are 0, then no processing takes place. Also, the utility doesn't recalculate time at stop or runtrn windows for the runtrn records modified. Run CalculateTimeAtStop.exe for now. For Flag values 4 through 33, all 30 positions MUST be filled in, even if the last several (or dozens) are only zero (0). Low count must be <= High count, with an increasing progression from left to right in the arguments (Ex: Low=1 High=5 Time=5 Low=6 High=10 Time=9, etc.). Refer to the documentation indicated above for more detailed information on setting up the command line arguments.

MakSht.exe

Description: Makes short files based on shortest path for walking distance.

Input: DYN: DBAJNG.DAT (built by NTADJANG.EXE)

Output: SHT: SHORTx.DAT, SHTHDx.DAT (Where x is 1, 2, 3, or 4, coinciding with the hazard level of the file)

MapSegUp.exe

Description: Update the mapseg database after changes have taken place in the geographic data.

Output: Mapsegup.err – this file lists errors generated while running mapsegup
TMP: DELSTP.AUD – this file lists stops that are no longer accessible
TMP: DELRUN.AUD – this file lists runs that have run direction segments that have been deleted
TMP: DELSEG.AUD - this file lists mapseg records that have been deleted

Special Notes: Updtadtm must be run after this utility. If running MapsegUp from EMU, updtadtm will be run automatically.

NTAdjang.exe

Description: Generates the file DYN:DBAJNG.DAT based on the GEO files.

Output: DYN: DBAJNG.DAT

PostRelocateSchool.exe

Description: This utility executes all the utilities that needs to be ran after relocating a school in ETC. This utility will run flagset.exe, bthelg.exe, schdst.exe, studst.exe, and stustopdist.exe.

Prestu.exe

Description: Build a NEWSTU file for UPSTU based on PRESTU.DAT and PRESTU.PRM.

Input: DYN: PRESTU.DAT, PRESTU.PRM

Output: DYN: NEWSTU.DAT
TMP: <date/time>PRESTU.ERR

Special Notes: See Prestu.doc for more details.

PstStu.exe

Description: Adds city codes to student addresses with duplicate street/address ranges. The input files, Ctycod.dat and Stname.dat, are generated and editable from within Buildtxt.exe, and are generally exported to the DYN directory for use by this utility.

Input: DYN: STNAME.DAT, CTYCOD.DAT

Output: TMP: <date/time>PSTSTU.AUD, <date/time>PSTSTU.CHK, <date/time>PSTSTU.ERR
EXPORT: PSTSTU.AUD, PSTSTU.ERR

Special Notes: The utility updates the student file, but does not update stutrn or cmndst, and if a home stop is needed, it does not update the home stop either. The StName.dat file contains max of 25,000 lines, and Ctycod.dat contains max of 250 lines.

Purgefox.exe

Description: Wrapper to call fox program purgefx that will pack all the transportation DBF files.

RebldKey.exe

Description: Rebuilds transportation data file keys. The following files are supported: Cluster, clustrn, cmndst, requst, route, rtetrn, rundir, runs, runtrn, schols, schtrn, stops, stptrn, studnt, stutrn

Special Notes: This utility, when run from the command line, is only capable of running on one file at a time. The option for psnger is still available, however, there is no psnger file, so this option will execute StopLoad only.

RebuildWalkPathFiles.exe

Description: Dumps geo data for walk based shortest path (wsp).

Output: DYN: walkmap.dmp, walkmapsize.dmp, shapenodes.dmp, convert.dmp

Rolovr.exe

Description: Advances students to new grade and/or new school. This utility is used to remove graduating seniors from the system and advance the remaining students to the next grade and/or new school at the end of the school year.

Input: STA: ROLOVR.DAT

Output: TMP: <date\time>ROLOVR.AUD, <date\time>ROLOVR.CHK, <date\time>ROLOVR.ERR
EXPORT: ROLOVR.AUD, ROLOVR.ERR

Special Notes: The grade = "-9 " is a valid grade selection, in effect graduating students (I.e. causes deletion flag to be set for these student records). When run from EMU, this utility has an extra option that allows users to move a Rolovr file(usually built from a fox executable called EditRoll.exe) from \EXPORT to \STA.

SchDst.exe

Description: Fills in DYN:SCHDST.DAT with distances from schools to individual node locations within the indicated distance.

Output: DYN: SCHDST.DAT

SchoolBnd.exe

Description: Utility to verify that all posted school boundaries still exist. If not, the school's posted boundary list is updated to remove invalid boundaries. Also makes sure that all posted boundaries are protected.

Output: TMP: <date\time>SCHOOLBND.AUD

SchoolLoadTime.exe

Description: This utility will fill in load times for all schools services on a run or the last/first school services on a run. It will only recalculate the time at stops if users ask for it to be updated.

Special Notes: This utility will need to be updated if we change LODTME to redo the calculation of time at stop for the runs.

SchoolScan.exe

Description: Address match schools.

Output: TMP: <date/time>SCHOOLSCAN.AUD, <date/time>SCHOOLSCAN.ERR

Errors reported in Schoolscan.err:
- No street matches found
- Too many street matches
- Street ok, Address not found

Official landmark not found
#node was an invalid node
Unable to relocate school
(x,y) not found

SpedLodTme.exe

Description: Loops through all the runtrn records and recalculates the loading time for special ed stops based on values the user provides to the utility. The user enters how much loading time per need one student requires, and the utility then multiplies those values by the actual number of students on the stop that have that need and sums up all of the values.

Input: None

Output: TMP: <date\time>SPEDLODTME.AUD, <date\time>SPEDLODTME.CHK
EXPORT: SPEDLODTME.AUD

StpChk.exe

Description: Reports on stops located on hazard segments, non-existent segments or segments that cannot be traveled.

Output: TMP: <date\time>STPCHK.CHK, <date\time>STPCHK.ERR
EXPORT: STPCHK.ERR

Errors reported in Stpchk.err:
Segment not an arterial
No travel segment
No speed for this segment
Wrong side of hazard segment
Segment not found
Non-right side stop on a hazard segment
Stop located on a hazard segment
Right side stop on wrong side of one-way segment
Right side stop on wrong side of one-way hazard

StpScn.exe

Description: Address match stops.

Output: TMP: <date\time>STPSCN.AUD, <date\time>STPSCN.CHK, <date\time>STPSCN.ERR
EXPORT: STPSCN.AUD, STPSCN.ERR

Errors reported in Stpscn.err:
No street matches found
Too many street matches
Street ok, Address not found
Official landmark not found
#node was an invalid node
All corner stops skipped
Ambiguous corners skipped
Street intersection error
Landmark nearest node error
Invalid segment w/o number
(x,y) not found

StpUpd.exe

Description: Changes street names in DYN:STOPS.DAT based on a fixed-format ASCII input file containing old and new location information. Changes any combination of "Prefix," "Street," "Type," "Suffix" and/or "Mapzone". These changes are also reflected in the description if the default description was chosen when the stop was built.

Output: TMP: <date\time>STPUPD.AUD, <date\time>STPUPD.CHK
EXPORT: STPUPD.AUD

Special Notes: The fixed-format file, containing old addresses and new addresses to apply to stop data, can be built manually or from within BuildTxt.exe.

StuBn2.exe

Description: Update each student record (user defined fields) with the appropriate boundary descriptions of the boundaries in which the student lives. The input file, Stubn2.dat, is generated and editable from within Buildtxt.exe, and is generally exported to the DYN directory for use by this utility.

Input: DYN: STUBN2.DAT

Output: TMP: STUBN2.ERR
TMP: STUBN2.AUD for each students put each of the boundaries that it is assigned to for Jordan, Utah only

Special Notes: The version number for the custom StuBn2.exe for Jordan now has an additional version that will be letters. Please remember to keep the major version numbers for these utilities the same.

StudentPurge.exe

Description: Purge all deleted student records

StuDst.exe

Description: Fill in student distance to school field in student record based on data stored in DYN:SCHDST.DAT

StuStopDist.exe

Description: Updates the distance to stop for both pickup and drop-off services in the stutrn records.

Output: TMP:INVALIDSTOPASSIGNMENTS.AUD

Special Notes: Need the short files to be up to date (see maksht.exe). Was formally named FxDstStp.exe (7/15/99).

UpdateGradeProgram.exe

Description: This utility updates STA:PRGRAM.DAT and STA:GRADES.DAT with the current programs and grades that are used in the system. It has a limit of 40 programs and grades.

Output: STA:GRADES.DAT
STA:PRGRAM.DAT

UpdateHeadCountLoad.exe

Description: For selected stptrn records will zeros out the headcount values, update them based on the assigned values, or update them with value in the input file.

Input: STA:UPDATEHEADCOUNTLOAD.INP

Output: TMP: <date\time>UPDATEHEADCOUNTLOAD.AUD

Special Notes: Input file: 1 – 14 = Service Id, 15 = Space, 16 – 17 = HeadCount Value

UpdateMaxRunTime.exe

Description: This utility allows users to update the runs max load and/or max ride time fields using the values in sysdef or entering a value. This does not update garage or deadhead runs. It will put the runs in an audit file if they exceed the new max run time or max load.

Output: TMP: <date\time>MAXRUNTIME.AUD

UpdateTime1and2.exe

Description: This utility processes each stop record, and updates the time1 and time 2 fields if they have changed. This utility was designed to fix up time 1 and time 2 fields that contained the value 32767.

UpdtAdTm.exe

Description: Builds adjacency and times files from the mapseg database.

Output: DYN: ARCADJ.BIN, ARCDST.BIN, ARCLEN.BIN, ARCTIM.BIN, SEGTIM.BIN, ARCCOUNT.DAT

UpLandmark.exe

Description: Adds new landmarks to the geocode from an input file (any name) eg. STA:UpLandmark.dat which is generated and editable from within Buildtxt.exe.

Input: Fixed-format input file (Ex: STA:UpLandmark.dat)

Output: TMP: <date\time>UpLandmark.aud, <date\time>UpLandmark.err

Special Notes: The format for the input file is the same as the georeports\landmark report with the addition of the contact.

UpperCase.exe

Description: Converts the contents of the input file to uppercase and then writes it to the output file. The file name defaults for the utility are Input (DYN:NEWSTU.DAT), and Output (DYN:NEWSTU.NEW). Users can enter different file names at the command line if needed.

Input: Optional input file name (Default is: Newstu.dat)

Output: Output file name (Default is: Newstu.new)

UpStu.exe

Description: Uploads student info from auxiliary sources into Edulog.nt. An ASCII file (DYN:NEWSTU.DAT) containing the student information to upload is processed, and the DYN:STUDNT.* files are updated.

Input: DYN: NEWSTU.DAT
STA: STUPRM.DAT, Translation files (if any) as specified by user

Output: TMP: UPSTU.AUD, UPSTU.CHK, UPSTU.ERR
EXPORT: UPSTU.AUD, UPSTU.ERR

Special Notes: Upmode 1 = all records are to be added. Upmode 4 = process records as Add/Change/Delete based on Newstu. Upmode 5 = process records as Add/Change based on existing data – incoming newstu records with blank last name fields will cause Upstu to process the record as a delete.