Skedc - Module exporting subroutines which return WMATA GTFS data from a database. Suitable for use by command line executed perl scripts. All exported subroutines are prefaced with **db** to facilitate their distinction in a calling script or program.

Exported Subroutines

dbGetRoutes - Gets a list of all bus routes from the database along with their corresponding route ids.

Arguments - none

Returns - A reference to an array of arrays in the form [[RtId1,RtName1],[RtId2,RtName2],..., [RtIdN,RtNameN]] where, for each route, RtId is the route_id (a number) and RtName is the route_short_name (a string). The array is ordered by route_id.

Example Array :

```
1    “10A”
2    “10B”
...  ...
307  “Z8”
308  “Z9”
```

dbGetStopRoutes - Gets a list of all routes providing service to a specified stop.

Arguments - stop_id

Returns - A reference to an array of route_ids for routes that service the stop indicated by stop_id.

Example Array (for stop_id 9820) :

```
11
12
13
14
15
16
17
18
20
21
22
```

dbGetTripHeadsigns - Gets a list of the headsigns for trips in both directions for a given route.

ASSUMPTION - A given route always has two directions.

Arguments - route_id

Returns - A reference to an array in the form [Direction0_Trip_Headsign, Direction1_Trip_Headsign], where the array elements are both strings. The array is ordered by direction_id, with the array index being equal to the direction_id.

Example Array :

```
“BALLSTON STATION”
“HUNTING POINT”
```

dbGetStopList - Gets a list of bus stops and related information from the database.

Arguments - route_id, direction_id, and a space-separated list of one or more specifiers from among the following:

- *wk* or *weekday* to specify weekday stops
- *sat* or *saturday* to specify saturday stops
- *sun* or *sunday* to specify sunday stops

Returns - A reference to an array of stops with each row in the form [stop_name, stop_id, stop_sequence, duplicate_flag]. All are numbers excepting stop_name, which is a string. The flag duplicate_flag is 1 if there is another stop in the array with an identical name but different stop_id. It is left to the caller to handle any such duplicate named different stops. The array is ordered by stop_sequence.

Example Array :

```
S WASHINGTON ST & SOUTH ST (T)     29997     1    0
S WASHINGTON ST & CHURCH ST        12688     2    0
...                                ...       ...  ...
N RANDOLPH ST & BALLSTON COMMOMS   26950     78   0
BALLSTON STATION & BUS BAY C       29216     79   0
```

dbGetStopTimetable - Gets a list of trip ids and arrival times at a particular stop.

Arguments :

- route_id or colon-separated list of stop_ids
- direction_id
- a single day-specifier (see encode_DaysToSQL)
- stop_id

Returns - A reference to an array with each row in the form [trip_id, stop_id, arrival_time, service_id]. All stop_ids in the array are identical to facilitate later combination with other such arrays. The array is ordered by arrival_time.

Example Array :

```
326  12218     06:07:12  2
327  12218     06:47:12  2
...  ...       ...
324  12218     24:09:12  2
325  12218     25:09:12  2
```

dbJoinStopTimetables - Combines a plurality of stop timetables into a multistop timetable.

Arguments :

- route_id
- An array of stop timetable references previously returned by dbGetStopTimetable. The references should be in order from earliest to latest within a trip.

Returns - A reference to an array with each row in the form [arrival_time, ..., arrival_time] where each arrival_time column corresponds to one of the stop timetable names, and where the columns are in the order of the stop timetable names in the argument array. The times are ordered in accordance with the leftmost column of arrival_times and with any NULL values in that column listed first.

Example Array :

```
2    4       --     05:22:12  05:28:12  05:32:42
2    4    05:18:48  05:52:12  05:58:12  06:02:42
2    4    05:48:48  06:23:12  06:29:42  06:34:42
...  ...  ...       ...       ...       ...
2    4    23:33:48  24:07:06  24:12:12  24:15:42
2    4    24:33:48  25:07:06  25:12:12  25:15:42
```

dbGetOverlappingRoutes - Gets a list or route_ids for routes that include, on a specified day, a specified number of stops corresponding to a list of stop_ids passed to the subroutine.

Arguments :

- A single day-specifier (see encode_DaysToSQL)
- Number of differing stops, where 0 specifies all of the stop_id-corresponding stops. A minimum of two common stops is enforced regardless of this number.
- A list of stop_ids

Returns - A reference to an array of with each row in the form [route_id, direction_id].

Example Array :

```
2      0
49     1
```

dbGetRouteNames - Gets short and long route names for a given route id.

Arguments - route_id

Returns - A reference to an array in the form [route_short_name, route_long_name], each name being a string.

Example Hash :

```
route_id (key)  info array (referenced by hash value)
--------------  ----------------------------------------
11              ('16A','')
12              ('16D','')
13              ('16E','')
```

Example Arrays :

* At time of writing, most WMATA bus routes include a blank route_long_name value:

  “16A”
  “”

* Exemplary Exception:

  “DCDGR”
  “DC Circulator: Dupont - Georgetown - Rosslyn”

* Exemplary Exception:

  “”
  “Richmond Highway Express Bus”

* Exemplary Exception:

  “TAGS”
  “Springfield Circulator-Metro Park Shuttle”

dbGetUserInfoForStops - Gets a stop name and NextBus id for each stop specified in a list of stop_ids.

Arguments - A list of stop_ids.

Returns - A reference to a hash containing to a list of references to arrays, each array containing a stop name and NextBus id for a given stop specified by a stop_id in the argument list. The stop_ids serve as hash keys.

Example Hash :

```
stop_id (key)   info array (referenced by hash value)
-------------   ----------------------------------------
10257           (5000689,'PATRIOT DR & AMERICANA DR')
10025           (6000137,'COLUMBIA PI & JEFFERSON ST')
9820            (6000242,'COLUMBIA PI & S WALTER REED DR')
```

dbGetSynonymousStops - For a given list of stop_ids, get stop_ids that include the given stop_ids and other stop_ids for stops, if any, at the same location (e.g. a Metro station) as each stop of the given stop_ids.

Arguments - array of stop_ids.

Returns - a hash record including:

- a “thesaurus” hash with stop_id keys and each value being a string including the key stop_id and its synonymous stop_ids separated by colons, or simply the stop_id if it has no synonymous stops.
- an “all” array containing stop_ids of all of the given stops and all of the synonyms.
- a “count” number indicating the total number of synonymous stops for the given stops.

Example Record Values :

```
“thesaurus”

     stop_id (key)       synonymous stops (value)
     -------------       --------------------------------
     30008               '30007:30008:30011:30012'
     12218               '12218'
     26815               '26815'
     29217               '29216:29217:29222:29293:29302'

“all”

     ('30007','30008','30011','30012','12218','26815','29216','29217','29222','29293','29302')

“count”

     7
```

dbGetProximateStop - For a given stop_id, find the closest stop on a specified overlapping route and direction_id within a specified distance from the given stop. The shortest distance is determined using comparison between summations of Great Circle distances along street paths obtained from the Gosmore routing application.

Arguments :

- stop_id of stop of interest.
- route_id of overlapping route obtained from dbGetOverlappingRoutes and, where applicable, dbGetSynonymousStops.
- direction_id of the overlapping route.
- Maximum distance, in miles, from the stop of interest to be considered.

Returns :

- A reference to an array including stop_id of the most proximate stop to the stop of interest and the distance between the stop of interest and the proximate stop; the form of the array is [stop_id, distance_in_miles], or
- 0 if the stop_id includes synonymous stops or no proximate stop is found.

Example Array :

```
10513     0.217609534136351
```

SortNulls - Export of SortNulls subroutine imported from the TimetableProcessor module.

MergeTimetables - Export of MergeTimetables subroutine imported from the TimetableProcessor

module.

MakeStopNameHeader - Export of MakeStopNameHeader subroutine imported from the TimetableProcessor module.

Non-Exported Subroutines

_connectdb - Connects program to a local MySQL or MariaDB database containing WMATA GTFS data.
Arguments - none
Caller - Module initialization code
Returns - handle to database
Failure Condition - kills program if unable to connect to database

_encodeDaysToSQL - Encodes a list of day-specifiers into an SQL WHERE clause search condition.
Arguments - A space-separated list of one or more day specifiers from among the following:
- *wk* or *weekday* to specify weekday stops
- *sat* or *saturday* to specify saturday stops
- *sun* or *sunday* to specify sunday stops

Caller - dbGetStopList, dbGetOverlappingRoutes, get_StopTimetable
Returns - A either a null string (if no valid specifier is listed) or a string usable as an SQL WHERE clause search condition to specify particular days in a SELECT statement retreiving data from the database Trips table.

_getStopList - Gets a list of bus stops and related information from the database.
Arguments - route_id, direction_id, days of interest encoded by encode_DaysToSQL as a WHERE clause search condition.
Caller - dbGetStopList
Returns - A reference to an array of stops with each row in the form [stop_name, stop_id, stop_sequence, duplicate_flag]. All are numbers excepting stop_name, which is a string. The flag duplicate_flag is 1 if there is another stop in the array with an identical name but different stop_id. It is left to the caller to handle any such duplicate named different stops. The array is ordered by stop_sequence.
Example Array :

```
S WASHINGTON ST & SOUTH ST (T)     29997     1    0
S WASHINGTON ST & CHURCH ST        12688     2    0
...                                ...       ...  ...
N RANDOLPH ST & BALLSTON COMMOMS   26950     78   0
BALLSTON STATION & BUS BAY C       29216     79   0
```

_getStopTimetable - Creates within the database a temporary table of trip_ids and arrival_times for a given stop. The temporary table has a name consisting of concatenated variable values $routeid$directioniddaystopid, where the value of $day is the day-specifier provided by the caller. The temporary table persists until the program closes. The same stop_id is included with each record to facilitate later combination with other such tables. The table is ordered by arrival_time.

Arguments :

- route_id or colon-separated list of stop_ids
- direction_id
- a single day-specifier (see encode_DaysToSQL)
- stop_id
- a name for the table to be created.

Caller - dbGetStopTimes

Returns - none

Table Description :

```
+--------------+------------+
| Field        | Type       |
+--------------+------------+
| trip_id      | int(11)    |
| stop_id      | int(11)    |
| arrival_time | varchar(8) |
| service_id   | int(11)    |
+--------------+------------+
```

_getTable - Performs a `SELECT * from` *`tablename`* operation to get a complete table from the database.

Arguments - The name *`tablename`* of a table in the database.

Caller - Any module-internal subroutine needing a table from the database.

Returns - A reference to an array of rows of *`tablename`*. If *tablename* has only one value per row, then the returned array is an array of values. Otherwise, an array of arrays is returned.

_getTripIds - Creates within the database a temporary table of trip_ids with associated service_ids from a union of all trip_ids from each of a plurality of stop timetables. The table presists until the next call to the subroutine or until the program closes.

Arguments - An array of stop timetable names (see _getStopTimetable).

Caller - dbJoinStopTimetables

Returns - none

Table Description :

```
+--------------+------------+
| Field        | Type       |
+--------------+------------+
| trip_id      | int(11)    |
| service_id   | int(11)    |
+--------------+------------+
```

_makeCSVlist - Copies a list of values into a string containing a comma separated list of the values with each value enclosed in single quotes.
Arguments - A list of values.
Callers - Any module-internal subroutine needing the aforementioned string; particularly useful in an SQL WHERE clause IN range check.
Returns - The aforementioned string.

_change_in_latitude - Given a distance of northward displacement, returns the change in latitude over that distance.
Arguments - Displacement distance in miles.
Caller - dbGetProximateStop
Returns - Change in latitude.

_change_in_longitude - Given a distance of latitude and a westward displacement, returns the change in longitude over that distance.
Arguments :

- Latitude of a position.
- Displacement distance in miles.

Caller - dbGetProximateStop
Returns - Change in longitude.

TimetableProcessor - Module exporting a subroutines for merging timetables and sorting trips in the tables into proper order. Used by Skedc.pm.

Exported Subroutines

SortNulls - Repositions timetable array rows having NULLs in the leftmost time column into proper locations within the timetable. The subroutine will sort based on successive time columns if necessary where a trip includes NULLs in succeeding time columns as well as the leftmost time column.

Arguments - Reference to the timetable array

Returns - Reference to the sorted timetable array

Example Array :

```
### UNSORTED ###
12   2       --     06:53:24  07:03:00  07:10:36
12   2       --     09:10:24  09:20:00  09:27:36
12   2       --     05:35:18  05:44:00  05:49:42
12   2       --     06:10:18  06:19:00  06:24:42
12   2    06:12:00  06:37:24  06:47:00  06:54:36
12   2    06:43:00  07:08:24  07:18:00  07:25:36
12   2    06:58:00  07:23:24  07:33:00  07:40:36
12   2    07:30:00  07:55:24  08:05:00  08:12:36
12   2    08:00:00  08:25:24  08:35:00  08:42:36
12   2    08:30:00  08:55:24  09:05:00  09:12:36
12   2    09:07:00  09:32:24  09:42:00  09:49:36
12   2    09:30:06  10:01:24  10:12:00  10:19:36
12   2    10:00:06  10:31:24  10:42:00  10:49:36

### SORTED ###
12   2       --     05:35:18  05:44:00  05:49:42
12   2       --     06:10:18  06:19:00  06:24:42
12   2    06:12:00  06:37:24  06:47:00  06:54:36
12   2       --     06:53:24  07:03:00  07:10:36
12   2    06:43:00  07:08:24  07:18:00  07:25:36
12   2    06:58:00  07:23:24  07:33:00  07:40:36
12   2    07:30:00  07:55:24  08:05:00  08:12:36
12   2    08:00:00  08:25:24  08:35:00  08:42:36
12   2    08:30:00  08:55:24  09:05:00  09:12:36
12   2       --     09:10:24  09:20:00  09:27:36
12   2    09:07:00  09:32:24  09:42:00  09:49:36
12   2    09:30:06  10:01:24  10:12:00  10:19:36
12   2    10:00:06  10:31:24  10:42:00  10:49:36
```

MergeTimetables - Merges a plurality of timetables and sorts the resulting timetable into proper order.

Arguments - A list of references to a plurality of timetable arrays to be merged.

Returns - A reference to the resulting merged and sorted timetable.

Example Array :

```
### Route 16A ###
11 4  04:43:00  04:56:30     --     05:16:30
11 4  05:21:00  05:34:30     --     05:54:30
11 4  06:02:00  06:15:30     --     06:35:30
11 4  06:56:00  07:09:30     --     07:31:18
11 4  07:54:00  08:07:30     --     08:30:30
11 4  08:46:00  09:00:30     --     09:20:36
11 4  09:47:00  10:01:30     --     10:21:36
11 4  10:47:00  11:01:30     --     11:21:36
11 4  11:47:00  12:01:30     --     12:21:36
11 4  12:47:00  13:01:30     --     13:21:36
11 4  13:47:00  14:01:30     --     14:21:36
11 4  14:48:00  15:02:30     --     15:22:36
11 4  15:46:00  16:00:30     --     16:21:30
11 4  16:53:00  17:07:30     --     17:28:30
11 4  17:51:00  18:03:30     --     18:21:36
11 4  18:54:00  19:06:30     --     19:24:36
11 4  19:54:00  20:06:30     --     20:24:36
11 4  20:24:00  20:36:30     --     20:54:36
11 4  20:59:00  21:10:18     --     21:24:36
11 4  21:29:00  21:40:18     --     21:54:36
11 4  22:29:00  22:40:18     --     22:54:36

### Route 16D ###
13 4  06:27:00  06:40:30     --     07:02:18
13 4  07:24:00  07:37:30     --     08:00:30
13 4  08:18:00  08:31:30     --     08:54:30
13 4  09:17:00  09:31:30     --     09:51:36
13 4  10:17:00  10:31:30     --     10:51:36
13 4  11:17:00  11:31:30     --     11:51:36
13 4  12:17:00  12:31:30     --     12:51:36
13 4  13:17:00  13:31:30     --     13:51:36
13 4  14:17:00  14:31:30     --     14:51:36
13 4  15:17:00  15:31:30     --     15:52:30
13 4  16:19:00  16:33:30     --     16:54:30
13 4  17:23:00  17:37:30     --     17:58:30
13 4  18:23:00  18:35:30     --     18:53:36
13 4  19:24:00  19:36:30     --     19:54:36

 ### Route 16E ###
14 4     --     22:56:12  23:10:36  23:14:24
14 4     --     23:17:12  23:31:36  23:35:24
14 4  23:20:00  23:36:12  23:50:36  23:54:24
14 4     --     24:06:12  24:20:36  24:24:24
14 4     --     24:40:12  24:54:36  24:58:24
14 5     --     25:06:12  25:20:36  25:24:24
14 5     --     25:48:12  26:02:36  26:06:24
14 5     --     26:08:12  26:22:36  26:26:24
```

```
### Routes 16A,16D,16E MERGED ###
11 4  04:43:00  04:56:30     --     05:16:30
11 4  05:21:00  05:34:30     --     05:54:30
11 4  06:02:00  06:15:30     --     06:35:30
13 4  06:27:00  06:40:30     --     07:02:18
11 4  06:56:00  07:09:30     --     07:31:18
13 4  07:24:00  07:37:30     --     08:00:30
11 4  07:54:00  08:07:30     --     08:30:30
13 4  08:18:00  08:31:30     --     08:54:30
11 4  08:46:00  09:00:30     --     09:20:36
13 4  09:17:00  09:31:30     --     09:51:36
11 4  09:47:00  10:01:30     --     10:21:36
13 4  10:17:00  10:31:30     --     10:51:36
11 4  10:47:00  11:01:30     --     11:21:36
13 4  11:17:00  11:31:30     --     11:51:36
11 4  11:47:00  12:01:30     --     12:21:36
13 4  12:17:00  12:31:30     --     12:51:36
11 4  12:47:00  13:01:30     --     13:21:36
13 4  13:17:00  13:31:30     --     13:51:36
11 4  13:47:00  14:01:30     --     14:21:36
13 4  14:17:00  14:31:30     --     14:51:36
11 4  14:48:00  15:02:30     --     15:22:36
13 4  15:17:00  15:31:30     --     15:52:30
11 4  15:46:00  16:00:30     --     16:21:30
13 4  16:19:00  16:33:30     --     16:54:30
11 4  16:53:00  17:07:30     --     17:28:30
13 4  17:23:00  17:37:30     --     17:58:30
11 4  17:51:00  18:03:30     --     18:21:36
13 4  18:23:00  18:35:30     --     18:53:36
11 4  18:54:00  19:06:30     --     19:24:36
13 4  19:24:00  19:36:30     --     19:54:36
11 4  19:54:00  20:06:30     --     20:24:36
11 4  20:24:00  20:36:30     --     20:54:36
11 4  20:59:00  21:10:18     --     21:24:36
11 4  21:29:00  21:40:18     --     21:54:36
11 4  22:29:00  22:40:18     --     22:54:36
14 4     --     22:56:12  23:10:36  23:14:24
14 4     --     23:17:12  23:31:36  23:35:24
14 4  23:20:00  23:36:12  23:50:36  23:54:24
14 4     --     24:06:12  24:20:36  24:24:24
14 4     --     24:40:12  24:54:36  24:58:24
14 5     --     25:06:12  25:20:36  25:24:24
14 5     --     25:48:12  26:02:36  26:06:24
14 5     --     26:08:12  26:22:36  26:26:24
```

Non-Exported Subroutines

_compareColumnTimes - Compares a selected value in a two-dimensional array to successive values in the same column as the value to determine the rows between which the row with the value should be positioned so that the array values are in ascending order from top to bottom.

Arguments :

- Reference to the array
- Index of row of selected value
- Index of column of selected value
- Reference to upper row of range of rows to compare
- Reference to lower row of range of rows to compare

Caller - SortNulls

Returns - none

Effective Returns - Modifies upper and lower references to values as close as possible to the indices of rows between which the row of the selected value should be positioned.

_moveTrip - Moves a row of a two-dimensional array from one position to another within the array.

Arguments :

- Reference to the array
- Index of row to be moved
- Index of row preceding the new location of row to be moved

Caller - SortNulls

Returns - none

_byFirstTimeColumn - Custom subroutine for use by the Perl sort function to sort trips into chronological order (earliest to latest) based on the leftmost timetable column containing arrival times and where trips with NULL times in that column are listed before the earliest time. The leftmost time column is hardcoded in global variable $firsttimecolumn.

Arguments - References to two rows of a timetable array temporarily assigned to global variables $a and $b.

Caller - sort function in MergeTimetables

Returns - 1, 0, or -1 depending on the result of a cmp comparison of the first values of arrays referenced by $a and $b.

TimetablePrinter_WMATA - Module exporting a subroutines to facilitate printing a timetable a style similar to that used by WMATA. Used by a program in conjunction with Skedc.pm.

Exported Subroutines

PrintStopnamesHeader - Prints a header including stop names intended to appear above time columns in a timetable.

Arguments - A reference to a list of stop names.

Returns - none

Example Stopnames Header :

```
                        Es ist
This is                 eine
only a      Hello,      Stop
test        World       Name
```

PrintNextbusIDsHeader - Prints a header including NextBus IDs intended to appear above time columns in a timetable.

Arguments :

- A reference to a list of stop_ids.
- A reference to a hash in the form returned by dbGetUserInfoForStops.

Returns - none

Example NextBus ID Header :

```
6000213   6000345   6000902
```

PrintServiceHeader - Prints AM, PM, or After Midnight Service headers at appropriate points in a timetable.

Arguments :

- 0 to print an initial AM Service header.
- A reference to a trip containing stop times to print subsequent Service headers.

Usage :

- Call initially with 0 prior to printing and timetable trips.
- Call subsequently with each trip reference prior to printing each timetable trip.

Returns - none

Example Service Header :

```
------------- AM Service -------------
```

PrintStoptime - Given a 24-plus-hour stoptime and a separator index indicating a chronsep x a predetermined list, prints an appropriate 12-hour stoptime in the form HHxMM. AM, PM, or After Midnight are to be indicated by position in the printed timetable.

Usage - Call for each stoptime in each trip prior to printing that stoptime.

Arguments :

- A stoptime in the form HH:MM:SS, where HH > 24 for hours past midnight.
- A separator index indicating a chronsep from the list `(':','!','¡','¦','|','\'','.')`.

Returns - none

PrintFootnotes - Prints a list of notes identifying the stops corresponding to proximate stop times including particular chronseps. A note is not printed for the standard chronsep, i.e. ':', which corresponds to the main route stops.

Arguments :

- A reference to a hash in the form returned by dbGetUserInfoForStops.
- A list of stop_ids in an order such that the position of a stop_id in the list corresponds to the index of the chronsep assigned thereto.

Returns - none

Example Footnote :

```
! = S GLEBE RD & S 13TH ST (6000198)
```

Non-Exported Subroutines

_makeStopNameHeader - Creates a two-dimensional array representing a header including stop names intended to appear above time columns in a timetable.

Arguments -

- Maximum column width (# of characters)
- Reference to an array of stop names

Caller - PrintStopnamesHeader

Returns - A reference to an array where each column includes a word-wrapped stop name. Row 0 is the bottom of the header, and wrapped names are bottom-aligned.

Example Array :

```
                        Es ist
This is                 eine
only a      Hello,      Stop
test        World       Name
```

_wordwrap - Converts a string to an array, breaking the string into chunks less than or equal to a specified length. Short words in the string are be combined where the combined length is within the specified length. Words that are longer than the specified length are truncated.

Arguments :

- Maximum column width (# of characters)
- Stop name

Caller - _makeStopNameHeader

Returns - Array containing chunks of the stop name string, each within the maximum column width.

_printServiceHeader - Prints a timetable section header including a given string followed by the word "Service", such as "---------- AM Service ----------".

Arguments - A string

Caller - PrintServiceHeader

Returns - none