
FRED Modeling Language Release Notes

['Epistemix Inc.']

Feb 07, 2022

CONTENTS

1	Revision Notes for FRED 7.11	3
1.1	7.11.0	3
1.2	Notes	3
2	Revision Notes for FRED 7.10	5
2.1	7.10.0	5
2.2	Notes	5
3	Revision Notes for FRED 7.9	7
3.1	7.9.0	7
3.2	Notes	8
4	Revision Notes for FRED 7.8	9
4.1	7.8.1	9
4.2	7.8.0	9
4.3	Notes	10
5	Revision Notes for FRED 7.7	13
5.1	7.7.4	13
5.2	7.7.3	13
5.3	7.7.2	13
5.4	7.7.1	13
5.5	7.7.0	14
5.6	Notes	14
6	Revision Notes for FRED 7.6	17
6.1	7.6.0	17
6.2	Notes	18
7	Revision Notes for FRED 7.5	19
7.1	7.5.0	19
7.2	Notes	19
8	Revision Notes for FRED 7.4	21
8.1	7.4.0	21
8.2	Notes	21
9	Revision Notes for FRED 7.3	23
9.1	7.3.0	23
9.2	Notes	24

10 Revision Notes for FRED 7.2	25
10.1 7.2.0	25
10.2 Notes	25
11 Revision Notes for FRED 7.1	27
11.1 7.1.0	27
11.2 Notes	28

[Return to docs home](#)

This document provides release notes for the FRED Modeling Language™, based on changes since FRED 7.0.0.

If you have questions about the company or are interested in speaking with our sales, recruiting, or other teams, please visit the [Epistemix](#) website.

REVISION NOTES FOR FRED 7.11

1.1 7.11.0

The following features were added in FRED version 7.11.0:

- The function `sample_without_replacement()` now uses the Efraimidis P., Spirakis P. (2008) algorithm for weighted sampling without replacement.
- New Meta agent action: `read_agents(<filename>, <global_list_variable>)` reads a CSV file with one line per new agent using the format `ID,AGE,RACE,SEX`. If `ID = 0`, FRED will generate a unique agent id. New agents are assigned a unique new household that is not added to the simulation region. New agents are not assigned to any other mixing group. The second argument is optional; if a global list-variable is supplied, it is overwritten with the list of IDs of the new agents.
- New place property `clear_schedule = 1` clears the scheduled opening times for a place type, to enable models to redefine schedules for built-in place types.
- Updated documentation.
- Bug fixes and efficiency improvements.

1.2 Notes

No further notes are provided for this release.

REVISION NOTES FOR FRED 7.10

2.1 7.10.0

The following features were added in FRED version 7.10.0:

- The file `FRED/FRED_AUTH_KEY` contains an authorization key that expires after a set period of time. If the file-based authorization key is expired, FRED checks the environmental variable `FRED_AUTH_KEY` for an updated authorization key before executing.
- `set_difference(list-expression1, list-expression2)` - returns the list of items that are in the first list but not in the second list.
- `sus_list(condition-name)` - returns the list of agents that are susceptible to the named condition.
- The functions `sample_with_replacement()` and `sample_without_replacement()` now take an optional third argument, a list of weights to be used when sampling from the list specified in the first argument. The list of weights must have the same length as the list in the first argument.
- Updated documentation.
- Bug fixes and efficiency improvements.

2.2 Notes

No further notes are provided for this release.

REVISION NOTES FOR FRED 7.9

3.1 7.9.0

The following features were added in FRED version 7.9.0.

- A new data type called `list_table` that has key-value pairs in which the value is a list. These will be global variables. Supporting functions include:
- `clear(list-table-name)` - erase all key-value pairs
- `list-table-name[key-or-key-list] = <list-expression>` - assign a list to one or more keys
- `<list-variable> = list-table-name[key]` - get a list associated with a key
- `erase(list-table-name,key)` - remove a list from the table

Other new functions:

- `sample_without_replacement(list-expression, k)` - returns a random sample of the list of size `k`, without replacement
- `sample_with_replacement(list-expression, k)` - returns a random sample of the list of size `k`, with replacement
- `select(list-expression, list-of-indexes)` - returns the sub-list of the first argument corresponding to the list of indexes in the second argument.
- `select(list-expression, test)` - returns the sub-list of the first argument consisting of the items that pass the test. The special variable `_` in the test refers to the list item being evaluated. For example `select(list(20, 5, 2, 40), (10 < _))` would return the list `(20, 40)`.
- `select_index(list-expression, test)` - returns the list of the indexes for the list in the first argument corresponding to items that pass the test. The special variable `_` in the test refers to the list item being evaluated. For example `select_index(list(20, 5, 2, 40), (10 < _))` would return the list `(0, 3)`.
- `prev_state(condition-name)` - return the agent's previous state before transitioning to the current state. Note: previous transition may have been from the current state.
- `add()`, `sub()`, `mult()` and `div()` all take list-valued arguments. If either or both argument is a list, the results is a list of values.
- Updated documentation.
- Bug fixes and efficiency improvements.

3.2 Notes

The parser translates `X[string]` to `select(X, string)`, so the following are equivalent:

- `select(x_list, index_list)` and `x_list[index_list]`
- `select(x_list, _>0)` and `x_list[_>0]` : returns all positive numbers in `x_list`
- `select(x_list, ask(_, age)>18)` and `x_list[ask(_, age)>18]` : returns the list of agents in `x_list` that have age > 18.

REVISION NOTES FOR FRED 7.8

4.1 7.8.1

The following features were added in FRED version 7.8.1.

- Compiler warning messages for all deprecated features.
- Updated documentation.
- Bug fixes and efficiency improvements.

4.2 7.8.0

The following features were added in FRED version 7.8.0:

- The new variable-type, `table`, has been added. See Chapter 5 of the guide.
- `int(<expression>)` - returns the integer part of the value of the expression
- `percentile(<expression>, <sorted_list_expression>)` - returns a real number between 0.0 and 100.0 reflecting the percent of the values in the sorted list expression that are less than or equal to the value of the first argument. The second argument is assumed to be a list sorted in ascending order. If it is not, the result is arbitrary.
- Several factors have been deprecated in favor of function calls. See details below.
- Symbolic names have been added for all county and state locations.
- Output file buffering has been enabled to reduce the frequency of file I/O operations.
- Compiler warnings have been added for all deprecated features.
- The following commands now take an option `-w 0` to suppress warning messages:

```
$ fred_compile <options> -w 0  
$ fred_job <options> -w 0  
$ run_fred <options> -w 0
```

- Bug fixes and efficiency improvements.

4.3 Notes

The following factors are now deprecated. The list below shows a new function to the left of the ← that is equivalent to the deprecated factor on the right:

- `current_state(condition_name)` ← `current_state_in_<condition_name>`
- `transmissions(<condition_name>)` ← `transmissions_of_<condition_name>`
- `source(<condition_name>)` ← `source_of_<condition_name>`
- `current_count(<condition_name>.<state_name>)` ← `current_count_of_<condition_name>.<state_name>`
- `total_count(<condition_name>.<state_name>)` ← `total_count_of_<condition_name>.<state_name>`
- `daily_count(<condition_name>.<state_name>)` ← `incidence_count_of_<condition_name>.<state_name>`
- `admin_id(<group_name>)` ← `admin_of_<group_name>`
- `income(<group_name>)` ← `income_of_<group_name>`
- `size(<group_name>)` ← `size_of_<group_name>`
- `adi_national_rank(<place_name>)` ← `adi_national_rank_of_<place_name>`
- `adi_state_rank(<place_name>)` ← `adi_state_rank_of_<place_name>`
- `block_group(<place_name>)` ← `block_group_of_<place_name>`
- `census_tract(<place_name>)` ← `census_tract_of_<place_name>`
- `county(<place_name>)` ← `county_of_<place_name>`
- `state(<place_name>)` ← `state_of_<place_name>`
- `latitude(<place_name>)` ← `latitude_of_<place_name>`
- `longitude(<place_name>)` ← `longitude_of_<place_name>`
- `elevation(<place_name>)` ← `elevation_of_<place_name>`
- `size(inlinks(<network_name>))` ← `degree_of_<network_name>`
- `size(inlinks(<network_name>))` ← `in_degree_of_<network_name>`
- `size(outlinks(<network_name>))` ← `out_degree_of_<network_name>`

The following factors have deprecated because they can be derived from other expressions:

- `sum_of_<variable_name>_in_<group_name>` → `sum(apply(members(<group_name>), ask(_, <variable_name>)))`
- `age_in_weeks` → `age_in_days / 7`
- `age_in_months` → `age_in_days / 30`
- `time_since_entering_<condition_name>.<state_name>` - replace by using a personal variable
- `average_of_<variable_name>_in_<group_name>` - compute sum then divide by size
- `current_percent_of_<condition_name>.<state_name>_in_<group_name>` - compute count then divide by size
- `incidence_percent_of_<condition_name>.<state_name>_in_<group_name>` - compute count then divide by size

- `total_percent_of_<condition_name>.<state_name>_in_<group_name>` - compute count then divide by size
- `size_quartile_of_<group_name>` - use percentile function
- `income_quartile_of_<group_name>` - use percentile function
- `elevation_quartile_of_<group_name>` - use percentile function
- `size_quintile_of_<group_name>` - use percentile function
- `income_quintile_of_<group_name>` - use percentile function
- `elevation_quintile_of_<group_name>` - use percentile function
- `last_inward_edge_in_<network_name>` - use variable
- `last_outward_edge_in_<network_name>` - use variable
- `inward_edge_with_max_weight_in_<network_name>` - use variable
- `outward_edge_with_max_weight_in_<network_name>` - use variable
- `inward_edge_with_min_weight_in_<network_name>` - use variable
- `outward_edge_with_min_weight_in_<network_name>` - use variable

REVISION NOTES FOR FRED 7.7

5.1 7.7.4

The following features were added in FRED version 7.7.4.

- Bug fixes and efficiency improvements in the snapshot-restart feature.

5.2 7.7.3

The following features were added in FRED version 7.7.3.

- Bug fixes in the snapshot-restart feature.
- Bug fixes in the order of processing the meta agent and the admin agents in rare circumstances.
- The default values for `start_date` and `end_date` have been removed. FRED will now generate a compiler error if either of these is missing.

5.3 7.7.2

The following features were added in FRED version 7.7.2.

- Bug fixes in the snapshot-restart feature.

5.4 7.7.1

The following features were added in FRED version 7.7.1.

- Bug fixes in the snapshot-restart feature.
- Bug fix in the case of an agent's state being initialized by two conditions (using `set_state()`)

5.5 7.7.0

The following features were added in FRED version 7.7.0.

- Simulation snapshots are now supported, as described in this note.
- The allocation of memory for conditions has been optimized, in particular so that conditions with `start_state = Excluded` and `admin_start_state = Excluded` now only allocate memory for the meta agent.
- New functions added to replace existing factors, as described later in this note. The replaced factors are now deprecated.

5.6 Notes

5.6.1 Snapshots and Restarts

The major change in this release is support for snapshots and restarts. A *snapshot* captures the state of a model at a given simulation time, while a *restart* is the ability to continue (restart) a simulation from a given snapshot. Two new simulation control parameters have been added to enable this capability.

```
snapshots = <N>           # default snapshots = 0
snapshot_interval = <days> # default snapshot_interval = 0
```

The `snapshots` parameter controls how many snapshots to keep. The `snapshot_interval` gives the number of days between snapshots. If `snapshots > 0` and no interval is given, the interval defaults to 99999 and each run will produce a single snapshot at the end of the run.

If `snapshot_interval > 0` and `snapshots` is not specified, the value of `snapshots` is set to 1.

Each run produces its own snapshot each time the number of days specified by `snapshot_interval` have passed. Run-specific snapshots are rolled into a single snapshot at the end of the job. This combined snapshot file for the job is stored in `../OUT/snapshot.tgz`.

The new FRED command `fred_get_snapshot` fetches the snapshot file for the job with the given key (specified as `-k <key>`) and copies it to the file `snapshot.tgz` in the current directory. To store the snapshot to an alternate file name use the `-o` option.

```
$ fred_get_snapshot -k <key> -o <output_file_name>
```

To restart a job from a snapshot, use the new `-R` option to `fred_job`:

```
$ fred_job -k <key> ... -R <snapshot_file_name>
```

The resulting job will be initialized using the given snapshot, and will set the simulation day to the day after the snapshot day. The `start_date` of the new job should be the same as the `start_date` of the snapshot job, and the `end_date` of the new job must be after the snapshot day.

In addition to the combined snapshot of all run snapshots produced at the end of job, interim job-level snapshot files are created throughout the job based on the `snapshot_mode` setting. This simulation parameter defaults to 0.

The possible values for this setting are 0 and 1.

- When `snapshot_mode = 0`, the first run to make a snapshot is designated the `snapshot_run`. When the `snapshot_run` completes, the next run to make a snapshot takes over as `snapshot_run`. Whenever the `snapshot_run` run produces a snapshot, a new job-level snapshot is created, combining the current run-level snapshots into a single snapshot file for the overall job. The use of mode 0 makes fewer job-level snapshots

than mode 1, although for a large job some run-level snapshots could be lost if the job is interrupted before it completes.

- When `snapshot_mode = 1`, each run creates a new job-level snapshot immediately after making a run-level snapshots. This mode makes more job-level snapshots, which could impact performance in a job with a large number of parallel runs. Mode 1 reduces the chance that a run-level snapshot is lost if the job is interrupted.

5.6.2 Functions that Replace Factors

The following list shows the new function to the left of the = sign that is equivalent to the factor on the right. The factors shown are now deprecated, with the function now preferred.

```
state(<condition_name>) = current_state_in_<condition_name>
transmissions(<condition_name>) = transmissions_of_<condition_name>
source(<condition_name>) = source_of_<condition_name>

current_count(<condition_name>.<state_name>, <group_name>) = current_count_of_<condition_
↳name>.<state_name>[_in_<group_name>]
total_count(<condition_name>.<state_name>, <group_name>) = total_count_of_<condition_
↳name>.<state_name>[_in_<group_name>]
daily_count(<condition_name>.<state_name>) = incidence_count_of_<condition_name>.<state_
↳name>
```

Note: If the second argument `<group_name>` is omitted in `current_count` or `total_count`, the function returns the count for the entire population.

REVISION NOTES FOR FRED 7.6

6.1 7.6.0

The following features were added in FRED version 7.6.0:

- Achieved major reduction in memory requirements through reorganization of fundamental data structures.
- Removed the built-in Self place in favor of adding latitude and longitude to each agent.
- To move the agent itself, use:

```
move(<dx>, <dy>)  
move_to(<place_name>)  
move_to_location(<latitude>, <longitude>)
```

- To access the agent's position, use:

```
my_lat = latitude()  
my_lon = longitude()  
my_x_coordinate = getx()  
my_y_coordinate = gety()
```

- Run-time error messages for attempts to move or get coordinates for non-existent places.
- Added `sqrt(x)` function.
- Additional compiler errors for unknown function names.
- Compiler error messages for rules that set an agent's susceptibility or transmissibility for non-transmissible conditions
- Support for synthetic population of Canada.
- Trigger a compiler error if any 32-bit integer property is initialized to a value larger than a 32-bit integer: 2147483647.

6.2 Notes

No further notes are provided for this release.

REVISION NOTES FOR FRED 7.5

7.1 7.5.0

The following features were added in FRED version 7.5.0.

- The default synthetic population updated to US_2010.v4 which includes zip codes for all places. Note that models will not look for this population by default in your FRED data directory under `country/usa/US_2010.v4`. See the installation instructions if you need to download this synthetic population.
- Added three new output files for each run:
 - `parameters.txt` - the final values of all parameters before the simulation begins.
 - `variables.txt` - the final value of all global variables before the simulation begins.
 - `simulation.txt` - the final values for simulation locations, `start_date`, `end_date` and `seed`.
- Changes to treatment of `parameters` blocks - a block is now treated as a sequence of assignment statements executed in order after all global variables are defined and initialized in `variables` block(s).

7.2 Notes

No further notes are provided for this release.

REVISION NOTES FOR FRED 7.4

8.1 7.4.0

The following features were added in FRED version 7.4.0.

- Added action `open_file(<filename>, <header>)` that specifies the header line for the given filename.
- Added action `print_file(<filename>, <expr>, ..., <expr>)` to print one line to a previously opened file.
- Added predicate `is_file_open(<filename>)` which returns true if the file has been opened, and false otherwise.
- The action `print_csv(<filename>, <expr>, ..., <expr>)` is now considered deprecated. Users should use `open_file(<filename>, <header>)` and `print_file(<filename>, <expr>, ..., <expr>)` instead, as these provide more robust error handling.
- `true` and `false` are now reserved words, and evaluate to 1 and 0, respectively.
- New library module `Read_Attributes` for assigning new attributes to agents based on an input file.
- New library module `Read_Schedule` for assigning values to variables based on the schedule in an input file.
- The compiler now substitutes the module name suffixed with a unique instance id for the string `$$$` occurring in the source code of a module.
- Any agent may change the state of any other agent or list of agents with the action `send(<expr>, <condition_name>, <state_name>)`
- Added function `zipcode(<place_type>)` that returns the zip code for the agent's indicated place type.
- Additional compiler warnings and error messages
- Improved efficiency and stability

8.2 Notes

No further notes are provided for this release.

REVISION NOTES FOR FRED 7.3

9.1 7.3.0

The following features were added in FRED version 7.3.0.

- Revised `agent_record` features
- Revised `agent_record` features
- Renamed `get_other()` to `ask()` – see the [FRED Language Guide](#) for more details
- Renamed `set_other()` to `tell()` – see the [FRED Language Guide](#) for more details
- Added actions on lists:
 - `push(<list_variable>, <expression>)` – Appends the value of the expression to the list.
 - `pop(<list_variable>)` – Removes the last item from the list.
- Added numeric operators on lists:
- `sum(<list_expression>)` - sum of values in the list
- `prod(<list_expression>)` - product of values in the list
- `max(<list_expression>)` - max of values in the list
- `min(<list_expression>)` - min of values in the list
- `last(<list_expression>)` - last value in the list
- `fred_make_sa` command to support sensitivity analysis workflows
- Added condition property `start_state = <state_name>`
- Improvements in efficiency and stability
- Improved compiler error and warning messages

9.2 Notes

No further notes are provided for this release.

REVISION NOTES FOR FRED 7.2

10.1 7.2.0

The following features were added in FRED version 7.2.0.

- `print_csv(<filename>, <expr>, ..., <expr>)`
- `fred_get_csv -k <key> -f <csv_file> -r <run>`

10.2 Notes

No further notes are provided for this release.

REVISION NOTES FOR FRED 7.1

11.1 7.1.0

The following features were added in FRED version 7.1.0.

- Updates to list processing functions:
- renamed `filter(<list_expr>,<test1>,,,<testN>)` to `filter_agents(<list_expr>,<test1>,,,<testN>)`
- renamed `filter_index(<list-expression>,<test1>,,,<testN>)` to `index_agents(<list-expression>,<test1>,,,<testN>)`
- For filtering numeric lists, added list-valued function `filter_values(<list_expr>,<op>,<expr>)` where `op` is `<`, `<=`, `=`, `==`, `>`, `>=`, `!=`
- For getting the indices of numeric lists, added list-valued function `index_values(<list_expr>,<op>,<expr>)`
- To select items from a list given the list of desired index values, added list-valued function `filter_by_index(<list_expr>,<list_of_desired_indexes>)`
- Added list-valued function `union(<list_expr>,<list_expr>)` to get a list of unique items that occur in either list, sorted in increasing order.
- Added list-valued function `intersection(<list_expr>,<list_expr>)` to get a list of unique items that occur in both lists, sorted in increasing order.
- Add list-valued functions `sort(<list>)`, `arg_sort(<list>)` and `shuffle(<list>)`
- Added the debugging function `abort()`. If triggered, a message is printed in errors.txt like

```
FRED terminated with errors:
=====
RUN1: FRED ERROR: Agent 164581176 aborts in condition F00 state B on sim day
↪10 sim date 2020-01-11:
if(age<18) then abort()
```

- Added an `Index out of range` error when reading a CSV file.
- Added a condition property `shuffle` which, when set, causes the order of agent-transitions to be shuffled for each time step, to ensure that agents are processed independently of their order in the input files. Note: this may cause an increase in run-time. Applies to all states within the condition.
- Added function `steps_between(<timestamp1>,<timestamp2>)` that returns the time steps between `<timestamp1>` and `<timestamp2>`. Positive results mean that `<timestamp2>` is later than `<timestamp1>`. Throws an error message for invalid timestamps.

- Any agent is permitted to execute the `add_site()` function. This was previously restricted to the meta agent, but in some models it may be useful for admin agents or ordinary agents to create new sites, e.g. a house party at an agent's household.
- Changes to FRED commands:
 - `fred_sweep` was upgraded to permit parallel sweeps to execute without interference.
 - `fred_format` command provides standardized formatting for FRED program files
 - `fred_debug` retrieves debugging statements associated with a job.
 - `fred_get_progress` shows the progress of any FRED job with a default value of 60 seconds. Previously was off by default.

11.2 Notes

The GitHub FRED-tutorial repository is now publicly accessible. See the [FRED Language Tutorials](#) document for details.