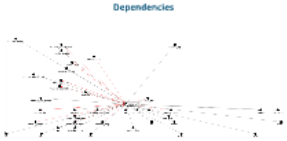
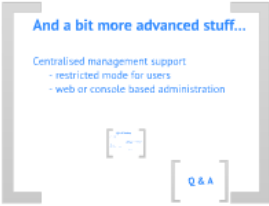
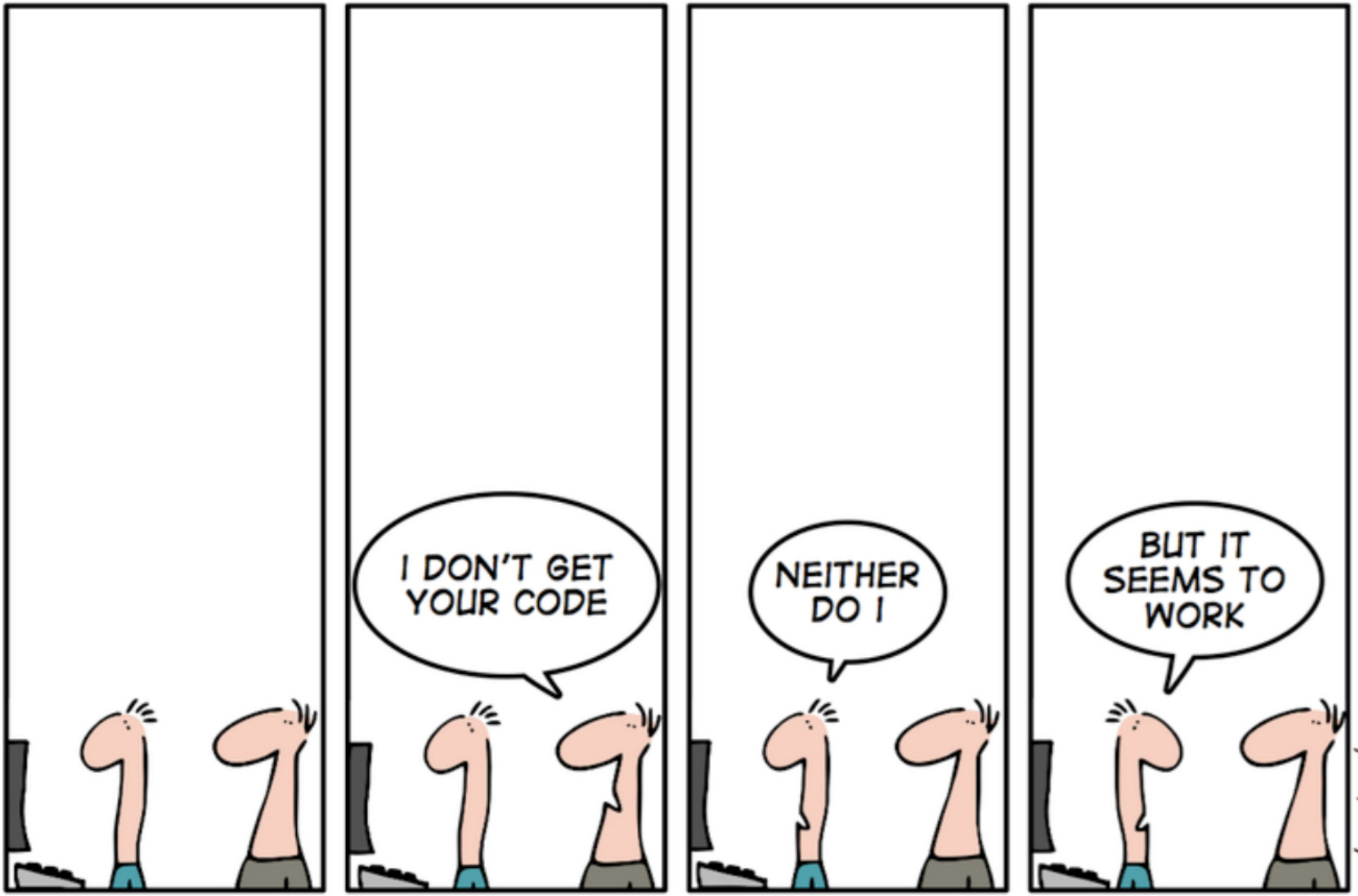


refactorerl.com



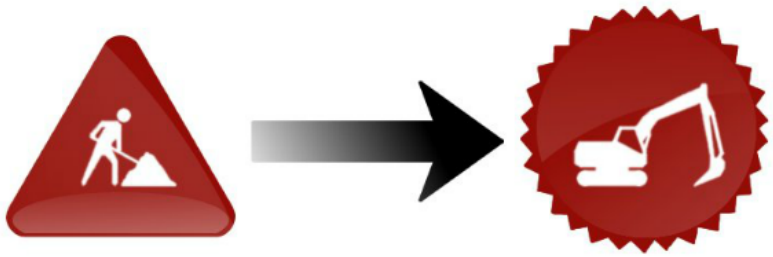


THE ART OF PROGRAMING

We do!

RefactorErl

Effective software maintenance



grokking

dependencies

investigations

code duplicates

metrics

refactoring

clustering

Knowledge sharing

Who are we?

Started as a refactoring project

University Staff &

Supports code comprehension

PhD, MSc, BSc students

ELTE-Soft R&D staff

Open Source! Try it!

refactorerl.com

Ericsson-ELTE Software Technology Lab (2011)

Contact:

tothmelinda@elte.hu

Open Source! Try it!

refactorerl.com

Ericsson-EI

Contact:

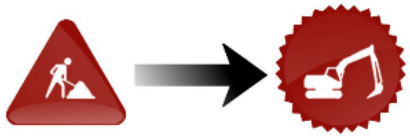
tothmelinda@elte.hu

What is RefactorErl?

Understand legacy code

Knowledge sharing

Effective software maintenance



Refactoring

Check code complexity/quality

in Erlang

for Erlang

Features

Compile-time analysis of
- Functions, variables, records, etc.
- Lifetime, scope, visibility
- Static and dynamic references

- Side-effects
- Data-flow, control-flow
- Dynamic function calls
- Hidden dependencies

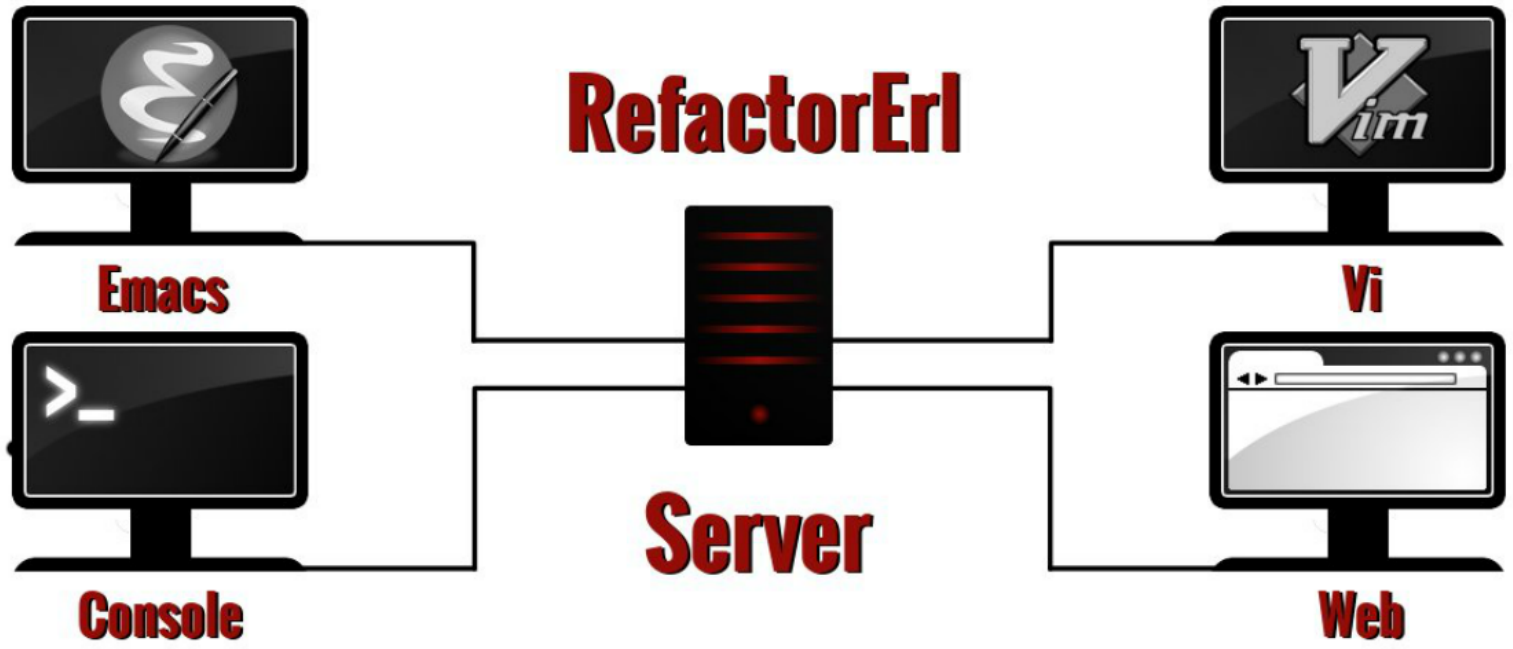
Program comprehension
- Semantic queries
- Software complexity metrics
- Bad smell detection
- Duplicated code detection
- Clustering
- Dependency visualisation

. . . and more than 20 refactoring transformations.

- Program comprehension
  - Semantic queries
  - Software complexity metrics
  - Bad smell detection
  - Duplicated code detection
  - Clustering
  - Dependency visualisation

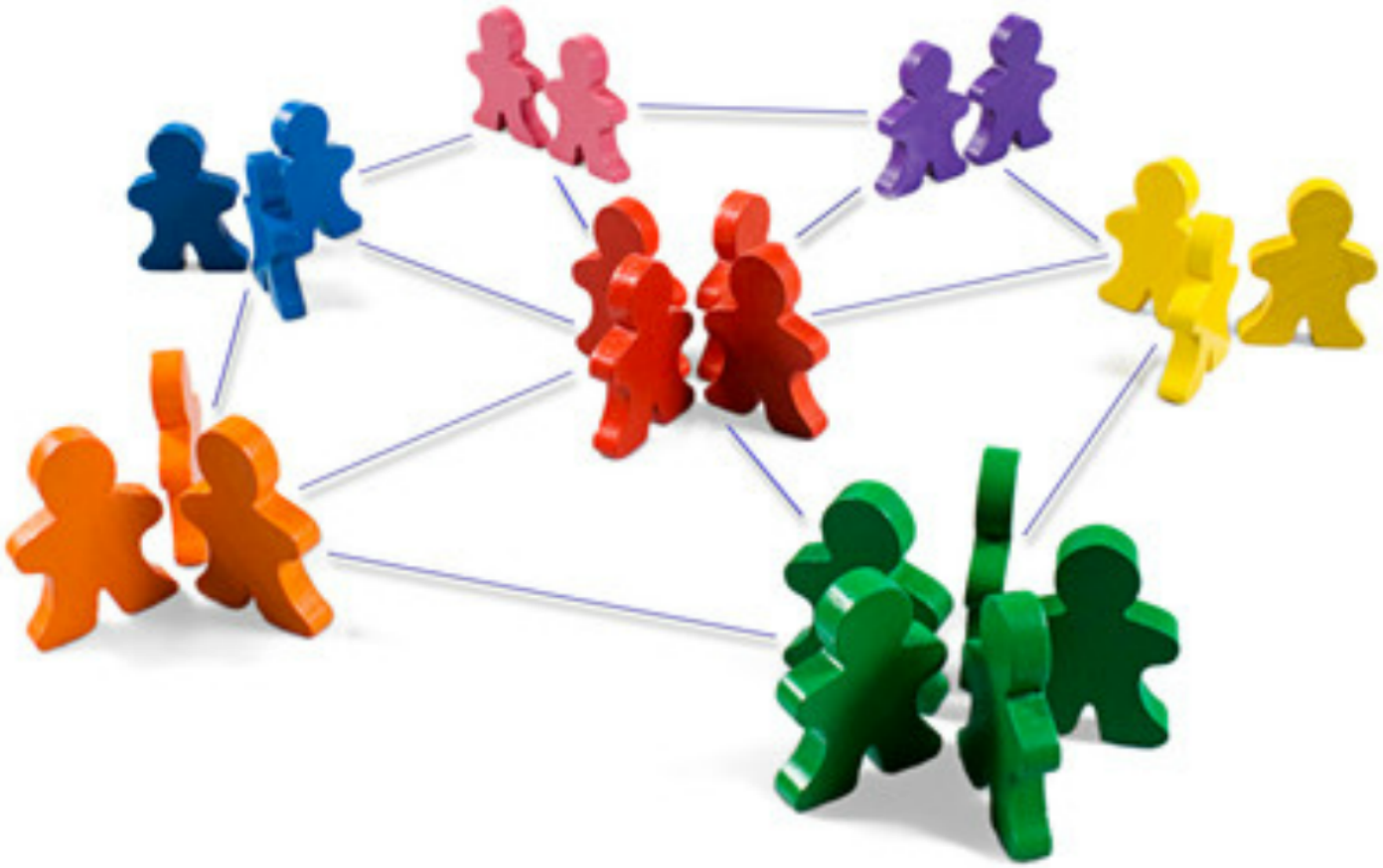
Easy to use, no restrictions

User Interfaces

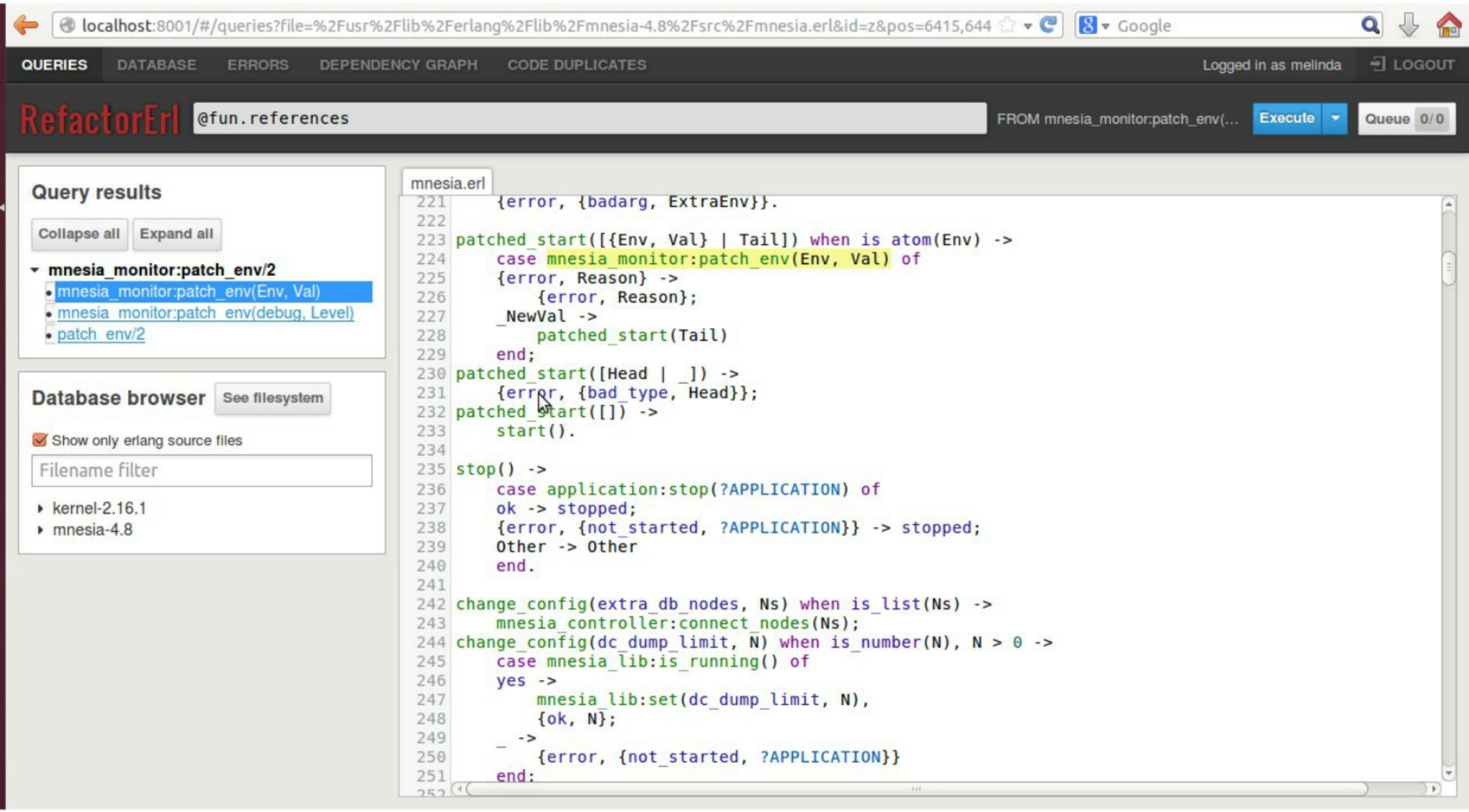


WX_GUI

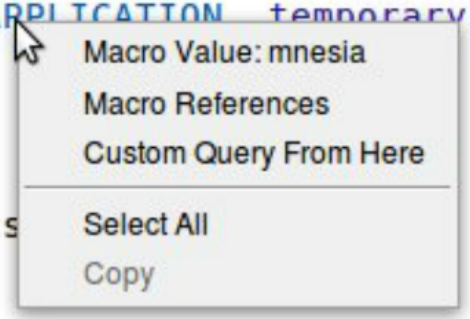
Sharing information between the team members

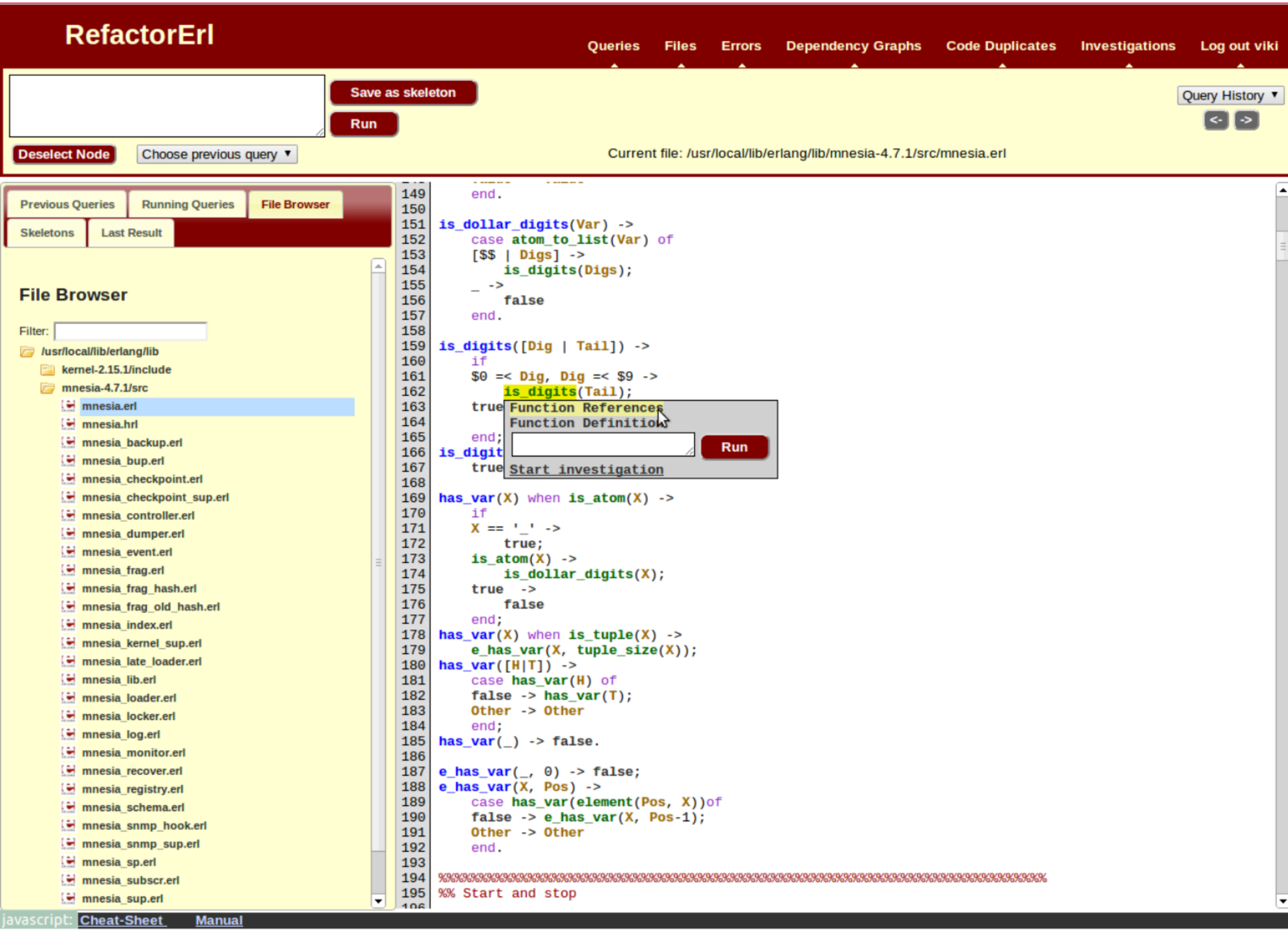


Gathering information about the source code

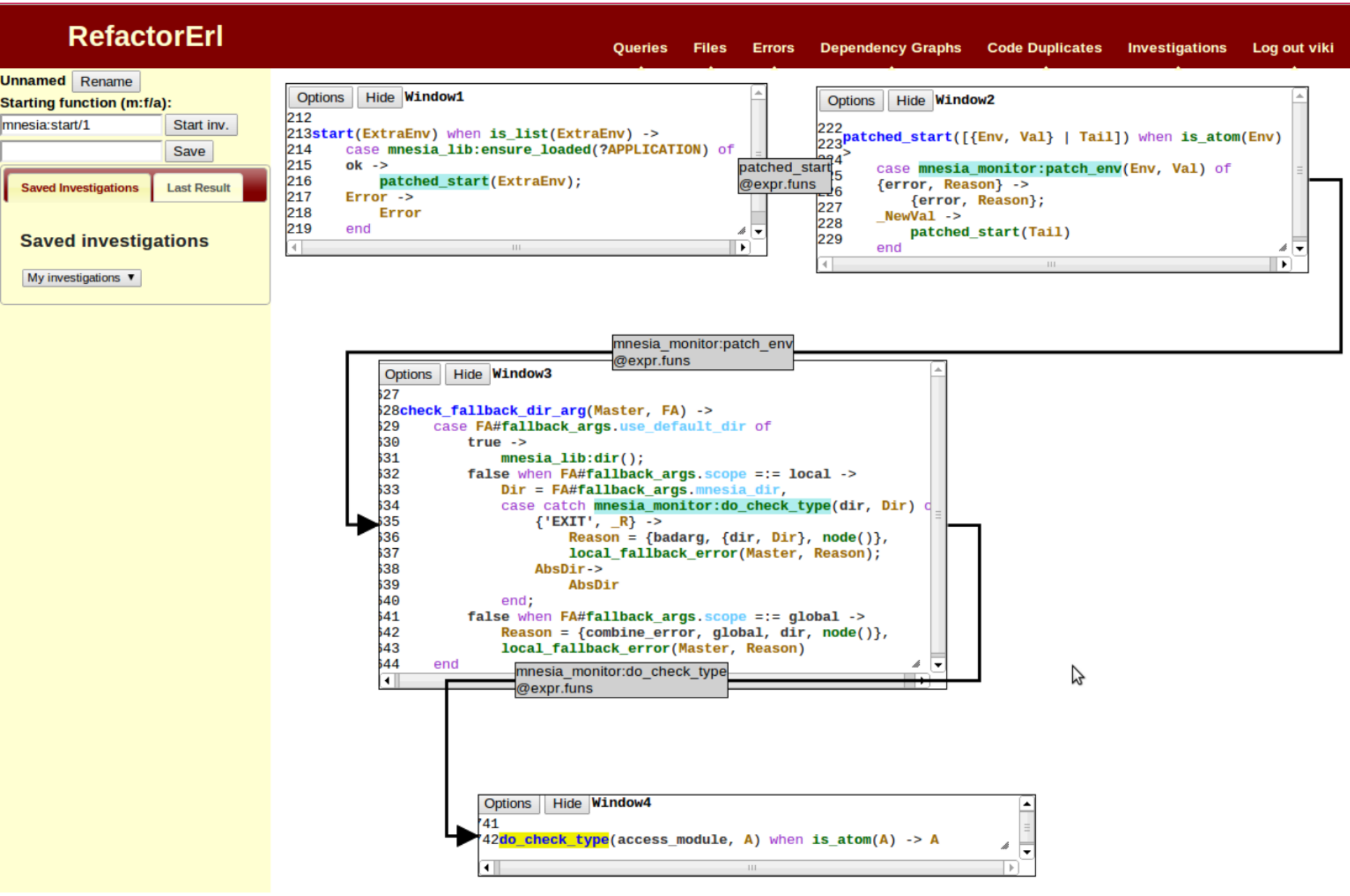


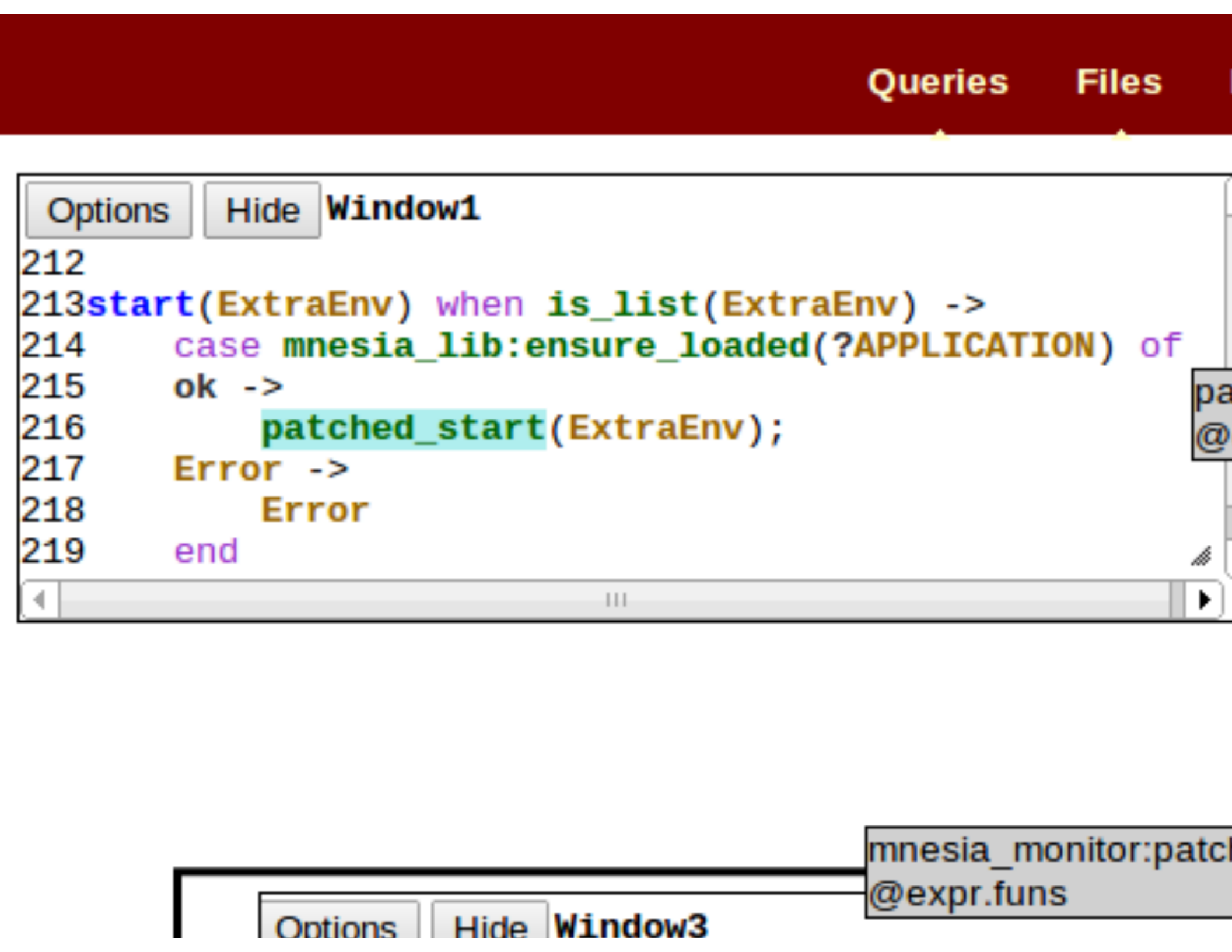
```
start() ->
    {Time , Res} =  timer:tc(application, start, [?APPLICATION, temporary]),

    Secs = Time div 1000000,
    case Res of
    ok ->
        verbose("Mnesia started, ~p seconds~n",[ Secs
        ok;
    {error, {already_started, mnesia}} ->
        verbose("Mnesia already started, ~p seconds~n",[ Secs]),
        ok;
    {error, R} ->
        verbose("Mnesia failed to start, ~p seconds: ~p~n",[ Secs, R]),
        {error, R}
    end.
```




RefactorErl
Queries
Files
Errors
Dependency Graphs
Code Duplicates
Investigations
Log out viki
Save as skeleton
Run
Query History
<-
->
Deselect Node
Choose previous query
Current file: /usr/local/lib/erlang/lib/mnesia-4.7.1/src/mnesia.erl
Previous Queries
Running Queries
File Browser
Skeletons
Last Result
File Browser
Filter:
/usr/local/lib/erlang/lib
kernel-2.15.1/include
mnesia-4.7.1/src
mnesia.erl
mnesia.hrl
mnesia_backup.erl
mnesia_bup.erl
mnesia_checkpoint.erl
mnesia_checkpoint_sup.erl
mnesia_controller.erl
mnesia_dumper.erl
mnesia_event.erl
mnesia_frag.erl
mnesia_frag_hash.erl
mnesia_frag_old_hash.erl
mnesia_index.erl
mnesia_kernel_sup.erl
mnesia_late_loader.erl
mnesia_lib.erl
mnesia_loader.erl
mnesia_locker.erl
mnesia_log.erl
mnesia_monitor.erl
mnesia_recover.erl
mnesia_registry.erl
mnesia_schema.erl
mnesia_snmp_hook.erl
mnesia_snmp_sup.erl
mnesia_sp.erl
mnesia_subscr.erl
mnesia_sup.erl
end.
is_dollar_digits(Var) ->
case atom_to_list(Var) of
[$$ | Digs] ->
is_digits(Digs);
_ ->
false
end.
is_digits([Dig | Tail]) ->
if
$0 =< Dig, Dig =< $9 ->
is_digits(Tail);
true
end;
is_digit
true
Function References
Function Definition
Run
Start investigation
has_var(X) when is_atom(X) ->
if
X == '_' ->
true;
is_atom(X) ->
is_dollar_digits(X);
true ->
false
end;
has_var(X) when is_tuple(X) ->
e_has_var(X, tuple_size(X));
has_var([H|T]) ->
case has_var(H) of
false -> has_var(T);
Other -> Other
end;
has_var(_) -> false.
e_has_var(_, 0) -> false;
e_has_var(X, Pos) ->
case has_var(element(Pos, X))of
false -> e_has_var(X, Pos-1);
Other -> Other
end.
%%
%% Start and stop
javascript:
Cheat-Sheet
Manual

Investigations

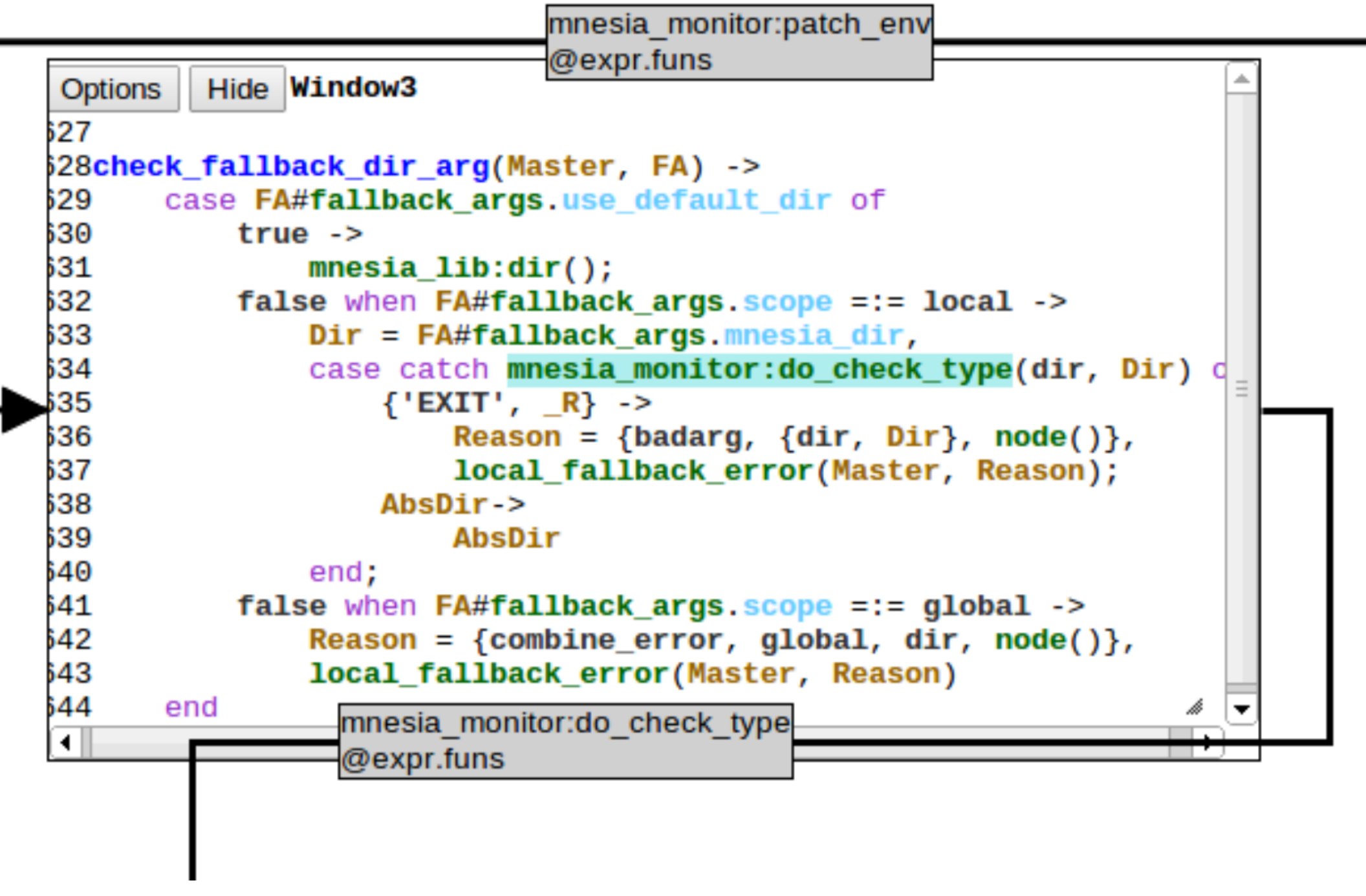



Queries
Files
Options
Hide
Window1
212
213start(ExtraEnv) when is_list(ExtraEnv) ->
214 case mnesia_lib:ensure_loaded(?APPLICATION) of
215 ok ->
216 patched_start(ExtraEnv);
217 Error ->
218 Error
219 end
mnesia_monitor:patch
@expr.funs
Options
Hide
Window3

Dependency Graphs Code Duplicates Investigations Log out

Options Hide **Window2**

```
222
223 patched_start([{Env, Val} | Tail]) when is_atom(Env)
224 ->
225     case mnesia_monitor:patch_env(Env, Val) of
226     {error, Reason} ->
227         {error, Reason};
228     _NewVal ->
229         patched_start(Tail)
        end
```

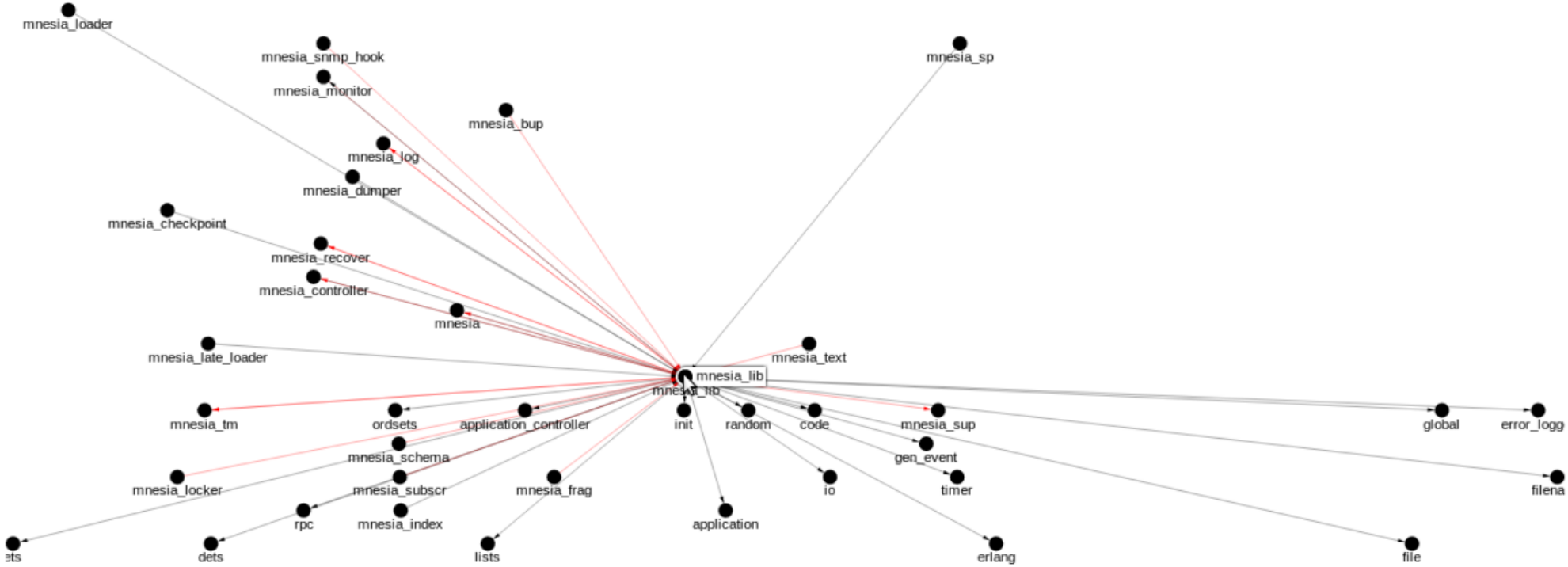

mnesia_monitor:patch_env
@expr.funs
Options
Hide
Window3
627
628check_fallback_dir_arg(Master, FA) ->
629   case FA#fallback_args.use_default_dir of
630   true ->
631   mnesia_lib:dir();
632   false when FA#fallback_args.scope =:= local ->
633   Dir = FA#fallback_args.mnesia_dir,
634   case catch mnesia_monitor:do_check_type(dir, Dir) c
635   {'EXIT', _R} ->
636   Reason = {badarg, {dir, Dir}, node()},
637   local_fallback_error(Master, Reason);
638   AbsDir->
639   AbsDir
640   end;
641   false when FA#fallback_args.scope =:= global ->
642   Reason = {combine_error, global, dir, node()},
643   local_fallback_error(Master, Reason)
644   end
mnesia_monitor:do_check_type
@expr.funs

@expr.funs

Options | Hide **Window4**

```
741
742do_check_type(access_module, A) when is_atom(A) -> A
```

Dependencies



Industrial Applications

Network Simulators

Telecom Gateway Controller

AXD ATM switch

More than 4 millions of LOC

Code Comprehension

Clustering

Industrial Applications

ork Simulators

Telecom Gateway Controller

ATM switch

More than 4 mill

More than 4 millions of Code Comprehension Clustering

Experience

"A problem solved in one hour using the query language"

"Without the tool: approx. 1 day"

Easy to setup and use

Initial setup - 2-3 hours

Build the database - few hours

- only once at the beginning

Use the tool!

Use the tool!

Some advantages

shorten time-consuming daily jobs

make the possibility of better teamwork in different ways

reduce human faults

ease deploying releases

minimise the training time of newbies

refactorerl.com

pnyf.inf.elte.hu/trac/refactorerl

News

Pattern Discovery

http://paraphrase-enlarged.elte.hu/

Demo time

http://plc.inf.elte.hu/erlang/cmd.txt

Demo time

http://plc.inf.elte.hu/erlang/cmd.txt

Quick Startup

bin/referl -build tool
bin/referl -db kcmini

ri:ls().

ri:envs().
ri:addenv(appbase, "/home/melinda/mylibpath").
ri:add(usr, mnesia).

ri:q(mods).
ri:q("mods.funs").
ri:q("mods[name~mnesia_snmp].funs").

ri:start_web2().

Simple easy to use features

Code Browser

Click and Browse

- function, record, record_field, variable, macro, etc
- built-in dynamic function calls
- references, definition, macro value on right click

Synchronised function list

Searh and find:

- plain text search
- semantic search

Dependence graphs

Database & File System Browser

Search and view Erlang & other files

Advanced Features

Semantic Queries & Parametrised Queries

```
mods[name=mnesia_log]
    .funs.exprs
        .sub[type=tuple[
            .sub[index=1]
                .origin[type=atom,
                        value =backup_args]]]
```

Collaborative work support

- query & graph sharing
- stateful links

```
mods[name=mnesia_log]
    .funs[name=open_log]
        .refs[
            .param[index=1]
                .origin[type=atom,
                        value=decision_log]]
```

Investigations

s & Parametrised Queries

```
mods[name=mnesia_log]
    .funs.exprs
        .sub[type=tuple[
            .sub[index=1]
                .origin[type=atom,
                        value =backup_args]]]
```

ort

ring

```
mods[name=mnesia_log]
    .funs[name=open_log]
        .refs[
            .param[index=1]
```

```
.sub[index=1]
    .origin[type=atom,
            value =backup_args]]]


mods[name=mnesia_log]
  .funs[name=open_log]
    .refs[
      .param[index=1]
        .origin[type=atom,
                value=decision_log]]
```

And a bit more advanced stuff...

Centralised management support

- restricted mode for users
- web or console based administration

Q & A

Q & A